# Comparative Study of Inference Methods for Bayesian Nonnegative Matrix Factorisation

Thomas Brouwer[1], Jes Frellsen[2], Pietro Lió[1]

[1]Computer Laboratory, University of Cambridge, United Kingdom
[2]Department of Computer Science, IT University of Copenhagen, Denmark

**Abstract.** In this paper, we study the trade-offs of different inference approaches for Bayesian matrix factorisation methods, which are commonly used for predicting missing values, and for finding patterns in the data. In particular, we consider Bayesian nonnegative variants of matrix factorisation and tri-factorisation, and compare non-probabilistic inference, Gibbs sampling, variational Bayesian inference, and a maximum-a-posteriori approach. The variational approach is new for the Bayesian nonnegative models. We compare their convergence, and robustness to noise and sparsity of the data, on both synthetic and real-world datasets. Furthermore, we extend the models with the Bayesian automatic relevance determination prior, allowing the models to perform automatic model selection, and demonstrate its efficiency.

## 1 Introduction

Matrix factorisation methods have been used extensively in recent years to decompose matrices into latent factors, helping us reveal hidden structure and predict missing values. In particular we decompose a given matrix into two smaller matrices so that their product approximates the original one (see Figure 1). Nonnegative matrix factorisation models [9] have been particularly popular, as the nonnegativity constraint makes the resulting matrices easier to interpret, and is often inherent to the problem—such as in image processing or bioinformatics [9, 20]. A related problem is that of matrix tri-factorisation, first introduced by Ding et al. (2006) [6], where the observed dataset is decomposed into three smaller matrices, which again are constrained to be non-negative.

Both matrix factorisation and tri-factorisation methods have found many applications in recent years, such as for collaborative filtering [13, 5], sentiment classification [11], predicting drug-target interaction [8] and gene functions [12], and image analysis [23]. Methods can be categorised as either non-probabilistic or Bayesian. For the former, finding the factorisation (*inference*) is commonly done using multiplicative updates, whereas for the latter we use approximate Bayesian inference methods. Non-probabilistic or maximum a posteriori (MAP) solutions give a single point estimate, which can lead to overfitting more easily and neglects uncertainty. Bayesian approaches address this issue, by instead finding a full distribution over the matrices, where we define prior distributions over the matrices and then compute their posterior after observing the actual data.

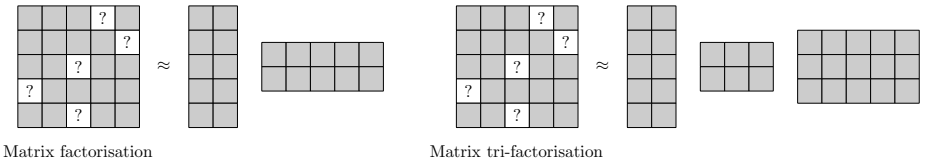



Fig. 1: Overview of matrix factorisation and matrix tri-factorisation methods, with missing values (?-entries).

This can greatly reduce overfitting. A key question that arises is: what exactly are the trade-offs between different matrix factorisation inference approaches? In particular, which perform better in terms of speed of convergence, predictive performance, and robustness to noise and sparsity?

In this paper we answer these questions by performing a thorough empirical study to explore these trade-offs between non-probabilistic and Bayesian inference approaches, which to our knowledge had not been done before. We consider the popular non-probabilistic matrix factorisation model from Lee and Seung (2000) [10], and a Bayesian nonnegative matrix factorisation and tri-factorisation model from Schmidt et al. (2009) [15] and Brouwer and Lió (2017) [4], respectively. These models use exponential priors to enforce nonnegativity, giving Gibbs sampling algorithms for inference. The former paper also introduced a MAP algorithm, called iterated conditional modes (ICM). Both of these approaches rely on a sampling procedure to eventually converge to draws of the desired distribution—in this case the posterior of the matrices. This means that we need to inspect the values of the draws to determine when our method has converged (burn-in), and then take additional draws to estimate the posteriors.

We introduce a fourth inference technique for the Bayesian nonnegative models, based on variational Bayesian inference (VB), where instead of relying on random draws we obtain deterministic convergence to a solution. We do this by introducing a new distribution that is easier to compute, and optimise it to be as similar to the true posterior as possible. Some papers (for instance [14]) assert that variational inference gives faster but less accurate inference than sampling methods like Gibbs. One study investigating this for latent dirichlet allocation can be found in [1], but ours is the first paper giving a thorough empirical study of the trade-offs for matrix factorisation. We furthermore extend the Bayesian models with automatic relevance determination (ARD), to eliminate the need for model selection.

We perform extensive experiments on both artificial and real-world data to explore the trade-offs between speed of inference, and robustness to sparsity and noise for predicting missing values. We show that Gibbs sampling is the most robust, while VB and ICM give significant run-time speedups but sacrifice some robustness, and that non-probabilistic inference tends to be fast but not robust. Finally, we show that ARD is an effective way of performing automatic model selection, and increases the robustness of matrix factorisation models if they are given the wrong dimensionality.

Although we study a specific Bayesian nonnegative matrix factorisation and tri-factorisation model, we believe that many of our findings and insights apply to the broad range of other matrix factorisation and tri-factorisation methods, as well as tensor and Tucker decomposition methods—their three-dimensional extensions.

## 2 Models

### 2.1 Nonnegative Matrix Factorisation

We follow the notation used by Schmidt et al. (2009) [15] for nonnegative matrix factorisation (NMF), which can be formulated as decomposing a matrix $\boldsymbol{R} \in \mathbb{R}^{I \times J}$ into two latent (unobserved) matrices $\boldsymbol{U} \in \mathbb{R}_+^{I \times K}$ and $\boldsymbol{V} \in \mathbb{R}_+^{J \times K}$, whose values are constrained to be positive. In other words, solving $\boldsymbol{R} = \boldsymbol{U}\boldsymbol{V}^T + \boldsymbol{E}$, where noise is captured by matrix $\boldsymbol{E} \in \mathbb{R}^{I \times J}$. The dataset $\boldsymbol{R}$ need not be complete—the indices of observed entries can be represented by the set $\Omega = \{(i,j) \mid R_{ij} \text{ is observed}\}$. These entries can then be predicted by $\boldsymbol{U}\boldsymbol{V}^T$.

We take a probabilistic approach to this problem. We express a likelihood function for the observed data, and treat the latent matrices as random variables. As the likelihood we assume each value of $\boldsymbol{R}$ comes from the product of $\boldsymbol{U}$ and $\boldsymbol{V}$, with some Gaussian noise added,

$$R_{ij} \sim \mathcal{N}(R_{ij} | \boldsymbol{U}_i \cdot \boldsymbol{V}_j, \tau^{-1})$$

where $\boldsymbol{U}_i, \boldsymbol{V}_j$ denote the $i$th and $j$th rows of $\boldsymbol{U}$ and $\boldsymbol{V}$, and $\mathcal{N}(x|\mu,\tau) = \tau^{\frac{1}{2}}(2\pi)^{-\frac{1}{2}} \exp\left\{-\frac{\tau}{2}(x-\mu)^2\right\}$ is the density of the Gaussian distribution with precision $\tau$. The set of parameters for our model is denoted $\boldsymbol{\theta} = \{\boldsymbol{U}, \boldsymbol{V}, \tau\}$. In the Bayesian approach to inference, we want to find the distributions over parameters $\boldsymbol{\theta}$ after observing the data $D = \{R_{ij}\}_{i,j \in \Omega}$. We can use Bayes' theorem,

$$p(\boldsymbol{\theta}|D) \propto p(D|\boldsymbol{\theta}) p(\boldsymbol{\theta}).$$

We need priors over the parameters, allowing us to express beliefs for their values—such as constraining $\boldsymbol{U}, \boldsymbol{V}$ to be nonnegative. We can normally not compute the posterior $p(\boldsymbol{\theta}|D)$ exactly, but some choices of priors allow us to obtain a good approximation. Schmidt et al. choose an exponential prior over $\boldsymbol{U}$ and $\boldsymbol{V}$, so that each element in $U$ and $V$ is assumed to be independently exponentially distributed with rate parameters $\lambda_{ik}^U, \lambda_{jk}^V > 0$.

$$U_{ik} \sim \mathcal{E}(U_{ik}|\lambda_{ik}^U) \qquad V_{jk} \sim \mathcal{E}(V_{jk}|\lambda_{jk}^V)$$

where $\mathcal{E}(x|\lambda) = \lambda \exp\{-\lambda x\}\, u(x)$ is the density of the exponential distribution, and $u(x)$ is the unit step function. For the precision $\tau$ we use a Gamma distribution with shape $\alpha_\tau > 0$ and rate $\beta_\tau > 0$,

$$p(\tau) \sim \mathcal{G}(\tau|\alpha_\tau, \beta_\tau) = \frac{{\beta_\tau}^{\alpha_\tau}}{\Gamma(\alpha_\tau)} x^{\alpha_\tau - 1} e^{-\beta_\tau x}$$

where $\Gamma(x) = \int_0^\infty x^{t-1} e^{-x} dt$ is the gamma function.

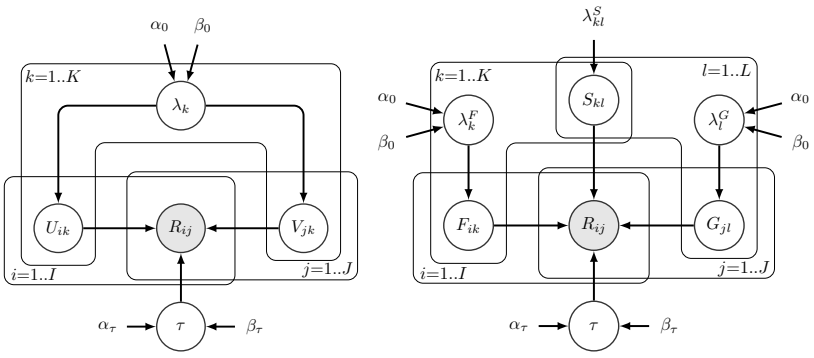



Fig. 2: Graphical model representation of Bayesian nonnegative matrix factorisation (left) and tri-factorisation (right), with ARD.

### 2.2 Nonnegative Matrix Tri-Factorisation

The problem of nonnegative matrix tri-factorisation (NMTF) can be formulated similarly to that of nonnegative matrix factorisation, and was introduced by Brouwer and Lió (2017) [4]. We now decompose $\boldsymbol{R}$ into three matrices $\boldsymbol{F} \in \mathbb{R}_+^{I \times K}$, $\boldsymbol{S} \in \mathbb{R}_+^{K \times L}$, $\boldsymbol{G} \in \mathbb{R}_+^{J \times L}$, so that $\boldsymbol{R} = \boldsymbol{F}\boldsymbol{S}\boldsymbol{G}^T + \boldsymbol{E}$. This decomposition has the advantage of extracting row and column factor values separately (through $\boldsymbol{F}$ and $\boldsymbol{G}$), allowing us to identify both row and column clusters. We again use a Gaussian likelihood and Exponential priors for the latent matrices.

$$R_{ij} \sim \mathcal{N}(R_{ij}|\boldsymbol{F}_i \cdot \boldsymbol{S} \cdot \boldsymbol{G}_j, \tau^{-1}) \qquad \tau \sim \mathcal{G}(\tau|\alpha_\tau, \beta_\tau)$$
$$F_{ik} \sim \mathcal{E}(F_{ik}|\lambda^F_{ik}) \qquad S_{kl} \sim \mathcal{E}(S_{kl}|\lambda^S_{kl}) \qquad G_{jl} \sim \mathcal{E}(G_{jl}|\lambda^G_{jl})$$

### 2.3 Automatic Relevance Determination

Automatic relevance determination (ARD) is a Bayesian prior which helps perform automatic model selection. It works by replacing the individual $\lambda$ parameters in the factor matrix priors by one that is shared by all entries in the same column (in other words, shared for each factor). We then place a further Gamma prior over all these $\lambda_k$ parameters. For the NMF model, the priors become

$$U_{ik} \sim \mathcal{E}(U_{ik}|\lambda_k) \qquad V_{jk} \sim \mathcal{E}(V_{jk}|\lambda_k) \qquad \lambda_k \sim \mathcal{G}(\lambda_k|\alpha_0, \beta_0).$$

Since this parameter is shared by all entries in the same column, the entire factor $k$ is either activated (if $\lambda_k^t$ has a low value) or "turned off" (if $\lambda_k^t$ has a high value), pushing factors that are active for only a few entities further to zero. This prior has been used for both real-valued [19, 18] and nonnegative matrix factorisation [17]. Instead of having to choose the correct $K$, we give an upper bound and the model will automatically determine the number of factors to use.

A similar approach can be found in [7], which incorporates the elimination of unused factors into their expectation-maximisation inference algorithm. ARD is implemented on a model level, and therefore works with all inference approaches.

For NMTF we use two ARD's, one for $\boldsymbol{F}$ ($\lambda_k^F$) and another for $\boldsymbol{G}$ ($\lambda_l^G$),

$$F_{ik} \sim \mathcal{E}(F_{ik}|\lambda_k^F) \quad \lambda_k^F \sim \mathcal{G}(\lambda_k^F|\alpha_0,\beta_0) \qquad G_{jl} \sim \mathcal{E}(G_{jl}|\lambda_l^G) \quad \lambda_l^G \sim \mathcal{G}(\lambda_l^G|\alpha_0,\beta_0).$$

The graphical models for Bayesian NMF and NMTF are given in Figure 2.

## 3 Inference

In this section we give details for four different types of inference for nonnegative matrix factorisation (NMF) and tri-factorisation (NMTF) models. Non-probabilistic inference gives a point estimate solution. Gibbs sampling and variational Bayesian inference both give a full posterior estimate, whereas iterated conditional modes gives a maximum a posteriori (MAP) point estimate.

### 3.1 Non-Probabilistic Inference

A non-probabilistic (NP) approach for NMF can be found in Lee and Seung (2000) [10]. Their algorithm relies on multiplicative updates, where at each iteration the values in the $\boldsymbol{U}$ and $\boldsymbol{V}$ matrices are updated using the following values:

$$U_{ik} = U_{ik} \frac{\sum_{j\in\Omega_i} R_{ij}V_{jk}/(\boldsymbol{U}_i\boldsymbol{V}_j)}{\sum_{j\in\Omega_i} V_{jk}} \qquad V_{jk} = V_{jk} \frac{\sum_{i\in\Omega_j} R_{ij}U_{ik}/(\boldsymbol{U}_i\boldsymbol{V}_j)}{\sum_{i\in\Omega_j} U_{ik}}$$

where $\Omega_i = \{j \mid (i,j) \in \Omega\}$ and $\Omega_j = \{i \mid (i,j) \in \Omega\}$. These updates can be shown to minimise the I-divergence (generalised KL-divergence),

$$D(\boldsymbol{R}||\boldsymbol{U}\boldsymbol{V}^T) = \sum_{(i,j)\in\Omega} \left( R_{ij} \log \frac{R_{ij}}{(\boldsymbol{U}\boldsymbol{V}^T)_{ij}} - R_{ij} + (\boldsymbol{U}\boldsymbol{V}^T)_{ij} \right).$$

Yoo and Choi (2009) [22] extended this approach to NMTF, giving the following multiplicative updates, with $\boldsymbol{S}_{\cdot l}$ denoting the $l$th column of $\boldsymbol{S}$:

$$F_{ik} = F_{ik} \frac{\sum_{j\in\Omega_i} R_{ij}(\boldsymbol{S}_k\boldsymbol{G}_j)/(\boldsymbol{F}_i\boldsymbol{S}\boldsymbol{G}_j)}{\sum_{j\in\Omega_i}(\boldsymbol{S}_k\boldsymbol{G}_j)} \quad G_{jl} = G_{jl} \frac{\sum_{i\in\Omega_j} R_{ij}(\boldsymbol{F}_i S_{\cdot l})/(\boldsymbol{F}_i\boldsymbol{S}\boldsymbol{G}_j)}{\sum_{i\in\Omega_j}(\boldsymbol{F}_i S_{\cdot l})}$$

$$S_{kl} = S_{kl} \frac{\sum_{(i,j)\in\Omega} R_{ij}F_{ik}G_{jl}/(\boldsymbol{F}_i\boldsymbol{S}\boldsymbol{G}_j)}{\sum_{(i,j)\in\Omega} F_{ik}G_{jl}}.$$

### 3.2 Gibbs Sampling

Schmidt et al. [15] introduced a Gibbs sampling algorithm for approximating the posterior distribution—a similar NMF model that uses Gibbs sampling can

be found in [24, 25]. Gibbs sampling works by sampling new values for each parameter $\theta_i$ from its marginal distribution given the current values of the other parameters $\boldsymbol{\theta}_{-i}$, and the observed data $D$. If we sample new values in turn for each parameter $\theta_i$ from $p(\theta_i|\boldsymbol{\theta}_{-i}, D)$, we will eventually converge to draws from the posterior, which can be used to approximate the posterior $p(\boldsymbol{\theta}|D)$. We have to discard the first $n$ draws because it takes a while to converge (*burn-in*), and since consecutive draws are correlated we only use every $i$th value (*thinning*).

For NMF this means that we need to be able to draw from distributions

$$p(\tau|\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\lambda}, D) \qquad p(U_{ik}|\tau, \boldsymbol{U}_{-ik}, \boldsymbol{V}, \boldsymbol{\lambda}, D)$$
$$p(\lambda_k|\tau, \boldsymbol{U}, \boldsymbol{V}, D) \qquad p(V_{jk}|\tau, \boldsymbol{U}, \boldsymbol{V}_{-jk}, \boldsymbol{\lambda}, D).$$

where $\boldsymbol{U}_{-ik}$ denotes all elements in $\boldsymbol{U}$ except $U_{ik}$, and similarly for $\boldsymbol{V}_{-jk}$. $\boldsymbol{\lambda}$ is a vector including all $\lambda_k$ values. Using Bayes theorem we obtain the following posterior distributions:

$$p(\tau|\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\lambda}, D) = \mathcal{G}(\tau|\alpha^*_\tau, \beta^*_\tau) \qquad p(U_{ik}|\tau, \boldsymbol{U}_{-ik}, \boldsymbol{V}, \boldsymbol{\lambda}, D) = \mathcal{TN}(U_{ik}|\mu^U_{ik}, \tau^U_{ik})$$
$$p(\lambda_k|\tau, \boldsymbol{U}, \boldsymbol{V}, D) = \mathcal{G}(\lambda_k|\alpha^*_k, \beta^*_k) \quad p(V_{jk}|\tau, \boldsymbol{U}, \boldsymbol{V}_{-jk}, \boldsymbol{\lambda}, D) = \mathcal{TN}(V_{jk}|\mu^V_{jk}, \tau^V_{jk})$$

where

$$\mathcal{TN}(x|\mu, \tau) = \begin{cases} \frac{\sqrt{\frac{\tau}{2\pi}} \exp\left\{-\frac{\tau}{2}(x-\mu)^2\right\}}{1 - \Phi(-\mu\sqrt{\tau})} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

is a truncated normal: a normal distribution with zero density below $x = 0$ and renormalised to integrate to one. $\Phi(\cdot)$ is the cumulative distribution function of $\mathcal{N}(0, 1)$.

For NMTF we can derive a Gibbs sampling algorithm similarly, as done by Brouwer and Lió [4]. The posteriors, together with the parameter values for both Gibbs samplers, are given in the supplementary materials.

### 3.3 Iterated Conditional Modes

The iterated conditional models (ICM) algorithm for inference in the NMF model was given in Schmidt et al. [15]. It works very similarly to the Gibbs sampler, but instead of randomly drawing a value from the conditional posteriors, we take the mode at each iteration. This gives a maximum a posteriori (MAP) point estimate $\boldsymbol{\theta}_{\text{MAP}} = \max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|D)$, rather than a full posterior distribution. We furthermore still need to use thinning and burn-in. For random variables $X \sim \mathcal{G}(a, b)$, $Y \sim \mathcal{TN}(\mu, \tau)$, the modes are $\frac{a-1}{b}$ and $\max(0, \mu)$, respectively.

In practice ICM often converges to solutions where multiple columns in the matrices are all zeros, leading to poor approximations. We have addressed this issue by resetting zeros to a small positive value like 0.1 at each iteration.

### 3.4 Variational Bayesian Inference

Variational Bayesian inference (VB) has been used for other matrix factorisation models before [8], but not for the nonnegative model in this paper. We therefore

now introduce a new VB algorithm for our model. Like Gibbs sampling, VB is a way to approximate the true posterior $p(\boldsymbol{\theta}|D)$. The idea behind VB is to introduce an approximation $q(\boldsymbol{\theta})$ to the true posterior that is easier to compute, and to make our variational distribution $q(\boldsymbol{\theta})$ as similar to $p(\boldsymbol{\theta}|D)$ as possible (as measured by the KL-divergence). We assume the variational distribution $q(\boldsymbol{\theta})$ factorises completely, so all variables are independent in the posterior,

$$q(\boldsymbol{\theta}) = \prod_{\theta_i \in \boldsymbol{\theta}} q(\theta_i).$$

This is called the mean-field assumption. We use the same forms of $q(\theta_i)$ as we used in Gibbs sampling,

$$\begin{aligned} q(\tau) &= \mathcal{G}(\tau|\alpha^*_\tau, \beta^*_\tau) & q(\lambda_k) &= \mathcal{G}(\lambda_k|\alpha^*_k, \beta^*_k) \\ q(U_{ik}) &= \mathcal{TN}(U_{ik}|\mu^U_{ik}, \tau^U_{ik}) & q(V_{jk}) &= \mathcal{TN}(V_{jk}|\mu^V_{jk}, \tau^V_{jk}). \end{aligned}$$

It can be shown [3] that the optimal distribution for the $i$th parameter, $q^*(\theta_i)$, can be expressed as follows (for some constant $C$), allowing us to find the optimal updates for the variational parameters.

$$\log q^*(\theta_i) = \mathbb{E}_{q(\boldsymbol{\theta}_{-i})} \left[\log p(\boldsymbol{\theta}, D)\right] + C.$$

We now take the expectation with respect to the distribution $q(\boldsymbol{\theta}_{-i})$ over the parameters but excluding the $i$th one. This gives rise to an iterative algorithm: for each parameter $\theta_i$ we update its distribution to that of its optimal variational distribution, and then update the expectation and variance with respect to $q$. We therefore need updates for the variational parameters, and to be able to compute the expectations and variances of the random variables. This algorithm is guaranteed to maximise the Evidence Lower Bound (ELBO)

$$\mathcal{L} = \mathbb{E}_q \left[\log p(\boldsymbol{\theta}, D) - \log q(\boldsymbol{\theta})\right],$$

which is equivalent to minimising the KL-divergence.

We use $\widetilde{f(X)}$ as a shorthand for $\mathbb{E}_q\left[f(X)\right]$, where $X$ is a random variable and $f$ is a function over $X$. For random variables $X \sim \mathcal{G}(a, b)$ and $Y \sim \mathcal{TN}(\mu, \tau)$ the variance and expectation are

$$\widetilde{X} = \frac{a}{b} \qquad \widetilde{Y} = \mu + \frac{1}{\sqrt{\tau}} \lambda\left(-\mu\sqrt{\tau}\right) \qquad \mathrm{Var}\left[Y\right] = \frac{1}{\tau}\left[1 - \delta\left(-\mu\sqrt{\tau}\right)\right],$$

where $\psi(x) = \frac{d}{dx}\log\Gamma(x)$ is the digamma function, $\lambda(x) = \phi(x)/[1-\Phi(x)]$, and $\delta(x) = \lambda(x)[\lambda(x) - x]$. $\phi(x) = \frac{1}{\sqrt{2\pi}}\exp\{-\frac{1}{2}x^2\}$ is the density function of $\mathcal{N}(0,1)$.

The updates for NMF are given in the supplementary materials. Our VB algorithm for NMTF follows the same steps as before, but now has an added complexity due to the term $\mathbb{E}_q\left[(R_{ij} - \boldsymbol{F}_i \cdot \boldsymbol{S} \cdot \boldsymbol{G}_j)^2\right]$. Before, all covariance terms for $k' \neq k$ were zero due to the factorisation in $q$, but we now obtain some additional non-zero covariance terms:

$$\mathbb{E}_q\left[(R_{ij} - \boldsymbol{F}_i \cdot \boldsymbol{S} \cdot \boldsymbol{G}_j)^2\right] = \left(R_{ij} - \sum_{k=1}^{K}\sum_{l=1}^{L} \widetilde{F_{ik}}\widetilde{S_{kl}}\widetilde{G_{jl}}\right)^2$$
$$+ \sum_{k=1}^{K}\sum_{l=1}^{L} \mathrm{Var}_q\left[F_{ik}S_{kl}G_{jl}\right] \quad (1)$$
$$+ \sum_{k=1}^{K}\sum_{l=1}^{L}\sum_{k'\neq k} \mathrm{Cov}\left[F_{ik}S_{kl}G_{jl}, F_{ik'}S_{k'l}G_{jl}\right] \quad (2)$$
$$+ \sum_{k=1}^{K}\sum_{l=1}^{L}\sum_{l'\neq l} \mathrm{Cov}\left[F_{ik}S_{kl}G_{jl}, F_{ik}S_{kl'}G_{jl'}\right]. \quad (3)$$

The above variance and covariance terms are equal to the following, respectively, leading to the variational updates given in the supplementary materials.

$$\widetilde{F_{ik}^2}\widetilde{S_{kl}^2}\widetilde{G_{jl}^2} - \widetilde{F_{ik}}^2\widetilde{S_{kl}}^2\widetilde{G_{jl}}^2, \quad \mathrm{Var}_q\left[F_{ik}\right]\widetilde{S_{kl}}\widetilde{G_{jl}}\widetilde{S_{kl'}}\widetilde{G_{jl'}}, \quad \widetilde{F_{ik}}\widetilde{S_{kl}}\mathrm{Var}_q\left[G_{jl}\right]\widetilde{F_{ik'}}\widetilde{S_{k'l}}.$$

### 3.5 Complexity

Each of the four approaches have the same time complexities, but vary in how efficiently the updates can be computed, and how quickly they converge. The time complexity per iteration for NMF is $\mathcal{O}(IJK^2)$, and $\mathcal{O}(IJ(K^2L + KL^2))$ for NMTF. However, the updates in each column of $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{F}, \boldsymbol{G}$ are independent of each other and can therefore be updated in parallel. For Gibbs and ICM this means we can draw these values in parallel, but for VB and NP we can jointly update the columns using a single matrix operation. Modern computer architectures can exploit this using vector processors, leading to a great speedup.

Furthermore, after the VB algorithm converges we have our approximation to the posterior distributions immediately, whereas with Gibbs and ICM we need to obtain further draws after convergence and use a thinning rate to obtain an accurate MAP (ICM) or posterior (Gibbs) estimate. This deterministic behaviour of VB and NP makes them easier to use. Although additional variables need to be stored to represent the posteriors, this does not result in a worse space complexity, as the Gibbs sampler needs to store draws over time.

### 3.6 Initialisation

Initialising the parameters of the models can vastly influence the quality of convergence. This can be done by using the hyperparameters $\lambda_{ik}^U$, $\lambda_{jk}^V$, $\lambda_{ik}^F$, $\lambda_{kl}^S$, $\lambda_{jl}^G$, $\alpha$, $\beta$, $\alpha_0$, $\beta_0$, $\alpha_0^F$, $\beta_0^F$, $\alpha_0^G$, $\beta_0^G$ to set the initial values to the mean of the priors of the model, or using random draws. We found that random draws tend to give faster and better convergence than the expectation, as it provides a better initial guess of the right patterns in the matrices. For matrix tri-factorisation we

Table 1: Overview of the four drug sensitivity datasets, giving the number of cell lines (rows), drugs (columns), and the fraction of entries that are observed.

| Dataset | Cell lines | Drugs | Fraction observed |
|---|---|---|---|
| GDSC $IC_{50}$ | 707 | 139 | 0.806 |
| CTRP $EC_{50}$ | 887 | 545 | 0.801 |
| CCLE $IC_{50}$ | 504 | 24 | 0.965 |
| CCLE $EC_{50}$ | 504 | 24 | 0.630 |

can initialise $\boldsymbol{F}$ by running the K-means clustering algorithm on the rows as datapoints, and similarly $\boldsymbol{G}$ for the columns, as suggested by Ding et al. (2006) [6]. For the VB and NP algorithms we then set the $\mu$ parameters to the cluster indicators, and for Gibbs and ICM we add 0.2 for smoothing. We found that this improved the convergence as well, with $\boldsymbol{S}$ initialised using random draws.

### 3.7 Software

Implementations of all methods, the datasets, and experiments described in the next section, are available at `https://github.com/ThomasBrouwer/BNMTF_ARD`.

## 4 Experiments

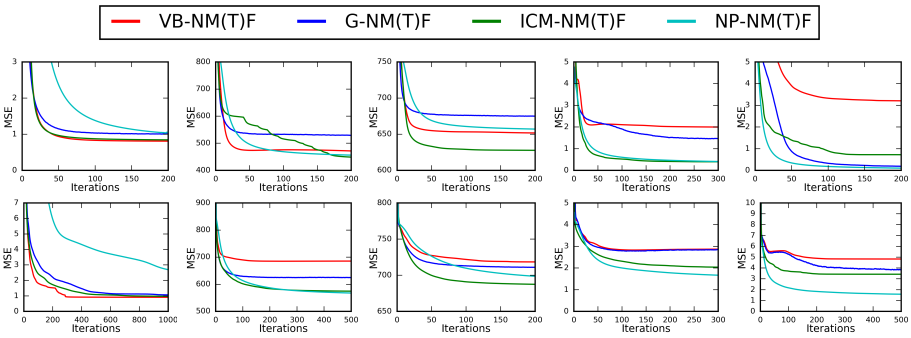
To demonstrate the trade-offs between the four inference methods presented, we conducted experiments on synthetic data and four real-world drug sensitivity datasets. We compare the convergence speed, robustness to noise, and robustness to sparsity.

### 4.1 Datasets

For the synthetic datasets we generated the latent matrices using unit mean exponential distributions, and adding zero mean unit variance Gaussian noise to the resulting product. For the matrix factorisation model we used $I = 100, J = 80, K = 10$, and for the matrix tri-factorisation $I = 100, J = 80, K = 5, L = 5$.

We considered four drug sensitivity datasets, each detailing the effectiveness ($IC_{50}$ or $EC_{50}$ values) of a range of drugs on different cell lines for cancer and tissue types, where some of the entries are missing. We consider the Genomics of Drug Sensitivity in Cancer (GDSC v5.0 [21], $IC_{50}$), Cancer Therapeutics Response Portal (CTRP v2 [16], $EC_{50}$), and Cancer Cell Line Encyclopedia (CCLE [2], $IC_{50}$ and $EC_{50}$). The four datasets are summarised in Table 1, giving the number of cell lines, drugs, and the fraction of entries that are observed.

In some experiments we focused on a selection of the datasets, but results for all can be found in the supplementary materials, together with preprocessing details. For all models we used weak priors ($\lambda = 0.1, \alpha_\tau = \beta_\tau = \alpha_0 = \beta_0 = 1$).




(a) Synthetic (b) GDSC (c) CTRP (d) CCLE $IC_{50}$ (e) CCLE $EC_{50}$

Fig. 3: Convergence of algorithms on the synthetic and drug sensitivity datasets, measuring the training data fit (mean square error) across iterations, for each of the inference approaches for NMF (top row) and NMTF (bottom row).

### 4.2 Convergence Speed

We firstly measured the convergence speeds of the different inference methods on the datasets, using the versions of NMF and NMTF without ARD. Convergence plots on all datasets are given in Figure 3, plotting the mean squared error on the training data against the number of iterations, for NMF (top row) and NMTF (bottom row). For the synthetic data we used the correct number of factors, and for the drug sensitivity datasets we used $K = 20$ for NMF and $K = L = 10$ for NMTF. We ran each method 20 times, taking the average training errors.

Although the results are empirical, they show that the inference approaches have different convergence speeds and depths (final training error reached). On the synthetic data VB is the fastest, followed by ICM and Gibbs, and finally NP. All methods reach the optimal MSE of 1 (which is the level of noise added). On the real-world drug sensitivity datasets, all methods reach their lowest depth at roughly the same number of iterations. However, ICM and NP generally converge much deeper than VB and Gibbs. Although this initially seems good, this is a sign of overfitting to the training data, and can lead to poor predictions for unseen data. We will see this later in the noise and sparsity experiments (Sections 4.4 and 4.5), where VB and Gibbs are more robust than ICM and NP.

In the supplementary materials we also give the convergence speed against time taken, which shows that the NP approach takes the least amount of time per iteration, followed by ICM, VB, and then Gibbs. In summary, ICM and NP give the fastest convergence, followed by VB, and then Gibbs.

### 4.3 Cross-Validation

Next we measured the cross-validation performances of the methods on the four drug sensitivity datasets. For each method we performed 10-fold nested cross-validation (nested to pick the dimensionality $K$—for simplicity we used $L = K$

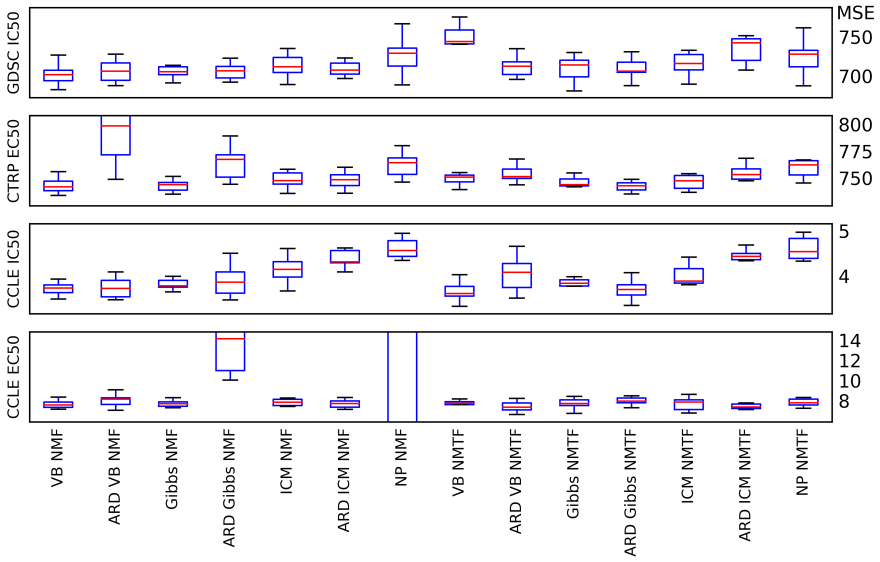



Fig. 4: 10-fold cross-validation results (mean squared error) for drug sensitivity predictions on each of the four datasets. Each boxplot gives the median (red line), standard deviation (blue box), and upper quartiles (black lines).

for the NMTF models), giving the average performance in Figure 4. For the ARD models we did not need to pick the dimensionality, instead using $K = 20$ for NMF, and $K = 10, L = 10$ for NMTF.

We can see that most models perform very similarly, with little to no difference between the matrix factorisation and tri-factorisation versions. Using the ARD models often works equally well as without ARD, but with the added benefit of not having to run nested cross-validation to choose the dimensionality, reducing the running time from hours to minutes. However, sometimes ARD fails to prevent overfitting, such as for VB NMF on CTRP $EC_{50}$, and Gibbs NMF on CCLE $EC_{50}$). This is unsurprising as the ARD models are given dimensionalities that are way too high. We will see in Section 4.6 that the ARD is actually very efficient at turning off unnecessary factors and reducing overfitting.

We can also see that the VB and Gibbs models often do a bit better than the NP and ICM versions. This is especially obvious on the CCLE $IC_{50}$ dataset, and also on GDSC $IC_{50}$. On the CCLE $EC_{50}$ dataset the NP NMF model completely overfits on one of the folds, leading to extremely high predictive errors.

### 4.4 Noise Test

We conducted a noise test on the synthetic data to measure the robustness of the methods. We add different levels of Gaussian noise to the data, with the noise-to-signal ratio being given by the ratio of the standard deviation of the

Gaussian noise we add, to the standard deviation of the generated data. For each noise level we split the datapoints randomly into ten folds, and measure the predictive performance of the models on one held-out set. The results are given in Figures 5a (NMF) and 5b (NMTF), where we can see that the non-probabilistic approach starts overfitting heavily at low levels of noise, whereas the Bayesian approaches achieve the best possible predictive powers even at high levels of noise. In the supplementary materials we also show that adding ARD did not make a difference for the robustness of the Bayesian models.

### 4.5 Sparsity Test

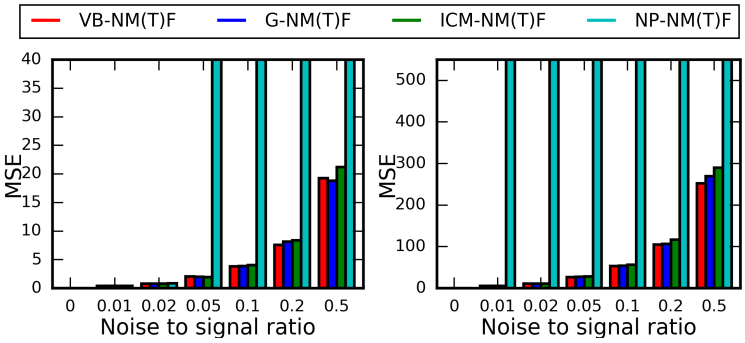
We furthermore measured the robustness of each inference technique to sparsity of the data. For different fractions of missing values we randomly split the data ten times into train and test sets using those proportions, and measured the average predictive error. We conducted this experiment on the synthetic data, using the true dimensionality $K$ (and $L$) for each model. We also performed it on the GDSC and CTRP datasets, using the most common dimensionalities in the cross-validation from Section 4.3 (given in supplementary materials).

The results are given in Figures 6a (NMF) and 6d (NMTF) for the synthetic data, 6b and 6e for GDSC, and 6c and 6f for CTRP. We can see that the non-probabilistic models start overfitting even on very low sparsity levels (with the exception of 6d)—in Figure 6a we cannot even see the line. The ICM models are also less robust when the sparsity is high. In contrast, the Gibbs sampling model achieves very good predictive performance even under extreme sparsity. The VB models are similar, but for sparser data it can sometimes not find the best solution, as can be seen in Figure 6d. We conducted this experiment for the models with ARD as well (results given in supplementary materials), where we show that ARD makes no difference to the robustness of Gibbs and VB (which are already very robust), but for ICM it can sometimes improve results.

### 4.6 Model Selection

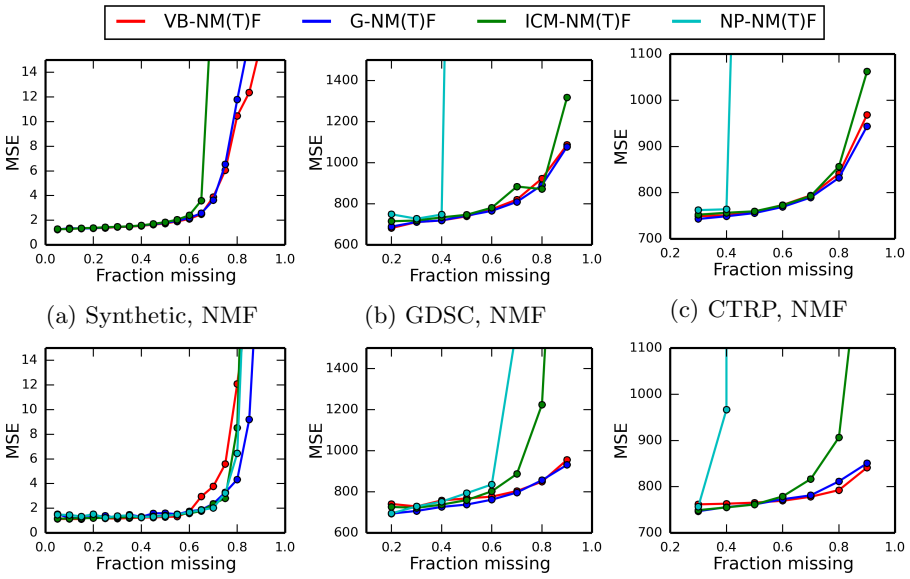
Finally, we conducted an experiment to see the extent of overfitting if the model is given a high dimensionality $K$, and whether this is remedied through the use of ARD. If we give a model a higher dimensionality, it can fit more to the data, but this can lead to overfitting and a higher predictive error. ARD can remedy this by turning off scarsely used factors, hopefully leading to less overfitting.

On the GDSC dataset, we performed 10-fold cross-validation for different values of $K$ (and $L$ for NMTF, using $K = L$) for Gibbs, VB, and ICM. We show these results in Figures 7a to 7f, where the results for models without ARD are given by crosses (x) and with ARD by circles (o). We can see that in most graphs, the models with ARD have a much flatter line as the dimensionality increases, hence reducing overfitting. This effect is more apparent for the NMF models than for the NTMF ones. The only exception is NMTF ICM, where the ARD is preventing the model from fitting as much to the data, hence leading to poor predictive results. Results for this experiment on the other three drug




(a) NMF (b) NMTF

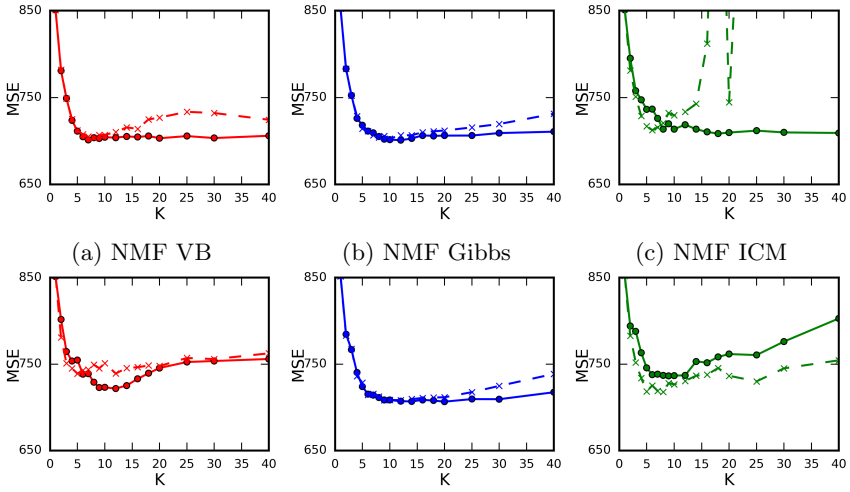
Fig. 5: Noise test performances, measured by average predictive performance on test set (mean square error) for different noise-to-signal ratios.




(a) Synthetic, NMF (b) GDSC, NMF (c) CTRP, NMF

(d) Synthetic, NMTF (e) GDSC, NMTF (f) CTRP, NMTF

Fig. 6: Sparsity test performances, measured by average predictive performance on test set (mean square error) for different sparsity levels. The top row gives the performances for NMF, and the bottom for NMTF, for the synthetic data (left), GDSC dataset (middle), and CTRP dataset (right).




(a) NMF VB (b) NMF Gibbs (c) NMF ICM

(d) NMTF VB (e) NMTF Gibbs (f) NMTF ICM

Fig. 7: 10-fold cross-validation performances of the Bayesian models on the GDSC dataset, where we vary the dimensionality $K$ (using $L = K$ for NMTF). The top row gives the performances for NMF, the bottom row for NMTF. Performances for models without ARD are given by dotted lines and crosses (x), with ARD by circles (o).

sensitivity datasets is given in the supplementary materials, which show that this problem only occurred for NMTF ICM on the GDSC dataset.

## 5 Conclusion

We have studied the trade-offs between different inference approaches for Bayesian nonnegative matrix factorisation and tri-factorisation models. We considered three methods, namely Gibbs sampling, iterated conditional modes, and non-probabilistic inference, and introduced a fourth one based on variational Bayesian inference. We furthermore extended these models with the Bayesian automatic relevance determination prior, to perform automatic model selection. Through experiments on both synthetic data, and real-world drug sensitivity datasets, we explored the trade-offs in convergence, robustness to noise, and robustness to sparsity.

A qualitative summary based on our quantitative findings can be found in Table 2. We found that the non-probabilistic methods are not very robust to noise and sparsity. Gibbs sampling is the most robust of the methods, especially for sparse datasets, and gives a full Bayesian posterior estimate. However, it converges slowly, and requires additional samples to estimate the posterior. Iterated conditional modes offers a much faster convergence and run-time speed, but sacrifices some robustness, still requires sampling, and no longer returns a full

Table 2: Qualitative comparison of inference methods.

| Method | Estimate | Requires sampling | Speed of convergence | Robustness |
|---|---|---|---|---|
| Non-probabilistic | Point | No | High | Low |
| Iterated conditional modes | Point (MAP) | Yes | High | Medium |
| Gibbs sampling | Full posterior | Yes | Low | High |
| Variational Bayes | Full posterior | No | Medium | Fairly high |

posterior (giving a MAP estimate instead). Our variational Bayesian inference gives good convergence speeds while maintaining more robustness properties.

Finally, we have shown that ARD is an effective way of reducing overfitting when using the wrong dimensionality in matrix factorisation models. This can eliminate the use for performing model selection, or nested cross-validation—although it is not perfect. We also discovered that adding ARD has little impact on performance, or on the robustness of the models to sparsity and noise (except for iterated conditional modes, where ARD increases its robustness to sparsity).

Our experiments were conducted for a specific version of Bayesian matrix factorisation and tri-factorisation, but we believe they offer insights into the trade-offs between different inference techniques in other matrix factorisation models, as well as tensor and Tucker decomposition methods.

**Acknowledgements.** This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC), grant reference EP/M506485/1. JF acknowledge funding from the Danish Council for Independent Research 0602-02909B.

## References

1. Asuncion, A., Welling, M., Smyth, P., Teh, Y.W.: On smoothing and inference for topic models. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (2009)
2. Barretina, J., Caponigro, G., Stransky, N., Venkatesan, K., Margolin, A.A., Kim, S., Wilson, C.J., et al.: The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. Nature 483(7391), 603–7 (2012)
3. Beal, M., Ghahramani, Z.: The Variational Bayesian EM Algorithm for Incomplete Data: with Application to Scoring Graphical Model Structures. Bayesian Statistics 7, Oxford University Press (2003)
4. Brouwer, T., Lió, P.: Bayesian Hybrid Matrix Factorisation for Data Integration. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS) (2017)
5. Chen, G., Wang, F., Zhang, C.: Collaborative filtering using orthogonal nonnegative matrix tri-factorization. Information Processing and Management 45(3) (2009)
6. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix t-factorizations for clustering. In: Proceedings of the 12th ACM SIGKDD (2006)

7. Figueiredo, M., Jain, A.: Unsupervised learning of finite mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(3), 381–396 (mar 2002)
8. Gönen, M.: Predicting drug-target interactions from chemical and genomic kernels using Bayesian matrix factorization. Bioinformatics 28(18) (2012)
9. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401(6755), 788–791 (1999)
10. Lee, D.D., Seung, H.S.: Algorithms for Non-negative Matrix Factorization. NIPS, MIT Press pp. 556–562 (2000)
11. Li, T., Zhang, Y., Sindhwani, V.: A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge. Proceeding of the 47th Annual Meeting of the Association for Computational Linguistics (2009)
12. Lippert, C., Weber, S., Huang, Y.: Relation prediction in multi-relational domains using matrix factorization. In: NIPS workshop on structured input, structured output (2008)
13. Salakhutdinov, R., Mnih, A.: Probabilistic Matrix Factorization. In: Advances in Neural Information Processing Systems (NIPS). pp. 1257–1264 (2008)
14. Salimans, T., Kingma, D.P., Welling, M.: Markov Chain Monte Carlo and Variational Inference: Bridging the Gap. In: Proceedings of the 32nd International Conference on Machine Learning (2015)
15. Schmidt, M.N., Winther, O., Hansen, L.K.: Bayesian non-negative matrix factorization. In: International Conference on Independent Component Analysis and Signal Separation, Springer Lecture Notes in Computer Science, Vol. 5441 (2009)
16. Seashore-Ludlow, B., Rees, M.G., Cheah, J.H., Cokol, M., Price, E.V., Coletti, M.E., Jones, V., et al.: Harnessing Connectivity in a Large-Scale Small-Molecule Sensitivity Dataset. Cancer discovery 5(11), 1210–23 (2015)
17. Tan, V.Y.F., Févotte, C.: Automatic relevance determination in nonnegative matrix factorization with the ($\beta$)-divergence. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(7), 1592–1605 (2013)
18. Virtanen, S., Klami, A., Khan, S., Kaski, S.: Bayesian group factor analysis. In: Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS) (2012)
19. Virtanen, S., Klami, A., Kaski, S.: Bayesian CCA via Group Sparsity. In: Proceedings of the 28th International Conference on Machine Learning (2011)
20. Wang, J.J.Y., Wang, X., Gao, X.: Non-negative matrix factorization by maximizing correntropy for cancer clustering. BMC bioinformatics 14(1), 107 (2013)
21. Yang, W., Soares, J., Greninger, P., Edelman, E.J., Lightfoot, H., Forbes, S., Bindal, N., et al.: Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. Nucleic acids research 41(Database issue), D955–61 (2013)
22. Yoo, J., Choi, S.: Probabilistic matrix tri-factorization. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (2009)
23. Zhang, D.Q., Chen, S.C., Zhou, Z.H.: Two-dimensional non-negative matrix factorization for face representation and recognition. In: Analysis and Modelling of Faces and Gestures. vol. 3723, pp. 350–363 (2005)
24. Zhong, M., Girolami, M.: Reversible Jump MCMC for Non-Negative Matrix Factorization. In: Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS-09). pp. 663–670 (2009)
25. Zhong, M., Girolami, M., Faulds, K., Graham, D.: Bayesian methods to detect dye-labelled DNA oligonucleotides in multiplexed Raman spectra. Journal of the Royal Statistical Society: Series C (Applied Statistics) 60(2), 187–206 (mar 2011)