PROBABILISTIC NON-NEGATIVE TENSOR FACTORIZATION USING MARKOV CHAIN MONTE CARLO

Mikkel N. Schmidt and Shakir Mohamed

Department of Engineering, University of Cambridge
Trumpington Street, Cambridge, CB2 1PZ, UK
email: mns@imm.dtu.dk, sm649@cam.ac.uk

ABSTRACT

We present a probabilistic model for learning non-negative tensor factorizations (NTF), in which the tensor factors are latent variables associated with each data dimension. The non-negativity constraint for the latent factors is handled by choosing priors with support on the non-negative numbers. Two Bayesian inference procedures based on Markov chain Monte Carlo sampling are described: Gibbs sampling and Hamiltonian Markov chain Monte Carlo. We evaluate the model on two food science data sets, and show that the probabilistic NTF model leads to better predictions and avoids overfitting compared to existing NTF approaches.

1. INTRODUCTION

Matrix factorization methods such as principal component analysis, singular value decomposition, factor analysis, independent component analysis, and non-negative matrix factorization (NMF) [1, 2] have become established tools in many data analysis tasks such as dimensionality reduction, missing data imputation, and data visualisation. These techniques restricts their analysis to that of two-dimensional matrix data, but frequently data occurs in the form of multiway arrays or tensors. When data have a natural multiway structure it is sensible to conserve this structure in the analysis of the data, as opposed to rearranging the data into a matrix and employing conventional matrix decomposition techniques.

Non-negative tensor factorization (NTF), which generalizes NMF, is an emerging technique for computing a non-negative low-rank approximation to multiway data array. Non-negativity is a natural constraint in many application areas; for example, when data are measurements of color intensities, counts, or spectral amplitudes, negative numbers do not have any physical interpretation. Furthermore, it has been shown for NMF that the non-negativity constraint often leads to sparse solutions and lends an intuitive parts based interpretation to the data.

In this light, we focus on non-negative factorizations of tensor data. The perhaps most well known model for tensor factorization is the PARAFAC model [3, 4], in which data tensor $\boldsymbol{X}$ is represented in polyadic form, i.e., as the sum of a finite number of rank one tensors. For a general I-dimensional tensor, $\boldsymbol{X} \in \mathbb{R}^{M_1 \times \cdots \times M_I}$ this decomposition can be written as

$$\boldsymbol{X} \approx \hat{\boldsymbol{X}} = \sum_{k=1}^{K} \boldsymbol{u}_1^k \otimes \cdots \otimes \boldsymbol{u}_I^k = \sum_{k=1}^{K} \bigotimes_{i=1}^{I} \boldsymbol{u}_i^k, \qquad (1)$$

where K is a positive integer, $\boldsymbol{u}_i^k$ are vectors in $\mathbb{R}^{M_i}$, and $\otimes$ denotes the outer product. In the general PARAFAC model the factors have no constraints, and the model has the desriable property that it is unique under mild conditions.

Variations of the PARAFAC model can be obtained by placing different constraints on the model factors and by considering different algorithms for parameter learning. Two such methods are the positive tensor factorization (PTF) [5] and non-negative tensor factorization (NTF) [6]. These algorithms are based on minimizing the reconstruction error,

$$\min_{\{\boldsymbol{u}_i^k\}} \frac{1}{2} \Big\| \boldsymbol{X} - \sum_{k=1}^{K} \bigotimes_{i=1}^{I} \boldsymbol{u}_i^k \Big\|_F^2, \quad \text{s.t. } u_i^k \geq 0, \qquad (2)$$

where $\|\cdot\|_F^2$ is the squared Frobenius norm, which is the sum of squares of all entries of the tensor elements. Welling and Weber [5] present a non-probabilistic approach where the factor matrices are learned using a fixed point algorithm similar to the multiplicative update rules for NMF proposed by Lee and Seung [2]. Shahsua and Hazan [6] present two algorithms: a direct approach using a gradient descent scheme and an expectation maximization (EM) approach using repeated rank-1 approximations.

We approach the non-negative tensor factorization problem by considering the PARAFAC model as a latent factor model. We describe a fully specified graphical model for the problem and employ Bayesian learning methods to infer the latent factors. This is important since the Bayesian approach has significant advantages in that it makes efficient use of the available data, allows prior information to be included into the model, avoids overfitting, allows a principled approach to model comparison and allows missing data to be handled easily. Explicit non-negativity constraints are not required on the latent factors, since this is naturally taken care of by the appropriate choice of prior distribution. We approach learning in this probabilistic NTF model through the use of Markov chain Monte Carlo (MCMC) techniques. Specifically, both Gibbs sampling and Hamiltonian Markov chain Monte Carlo (HMC) is discussed.

The structure of the paper is as follows: Section 2 provides additional notation and presents NTF as a probabilistic latent factor model, and section 3 discusses the details of Gibbs sampling and HMC for inference in the model. Section 4 presents experimental results on two food science datasets, and we conclude in section 5.

2. NON-NEGATIVE TENSOR FACTORIZATION

In non-negative tensor factorization (NTF), the observed data is an I-way array, $\boldsymbol{X} \in \mathbb{R}^{M_1 \times \cdots \times M_I}$, where the dimensions of each mode are denoted by $M_1, \ldots, M_I$. Let $\mathcal{M} = \{1, \ldots, M_1\} \times \cdots \times \{1, \ldots, M_I\}$ be the index set over all elements in $\boldsymbol{X}$ and let $\boldsymbol{m} = (m_1, \ldots, m_I)$ be an I-tuple index in $\mathcal{M}$. We denote the total number of elements in $\boldsymbol{X}$ by $M = \prod_i M_i$.

We seek to find a decomposition of the form given by Eq. (1) that approximates $\boldsymbol{X}$ as the sum of K rank-1 tensors that are outer products of I non-negative vectors, $\boldsymbol{u}_i^k \in \mathbb{R}^{M_i}$.

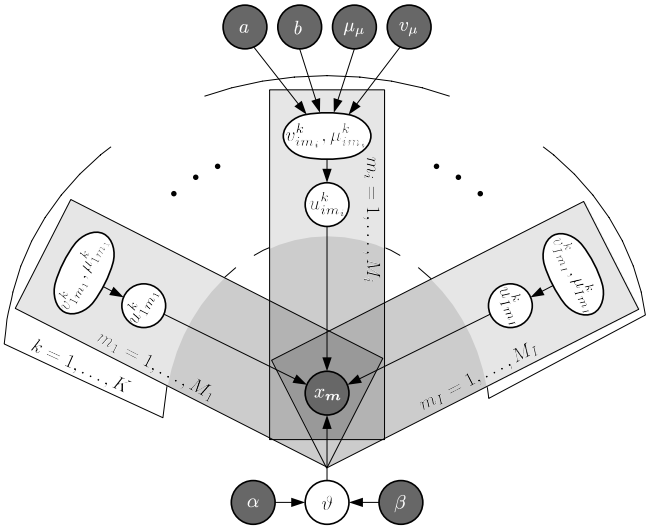

Figure 1: Graphical model of Bayesian NTF. Dark and white circles denote observed and unobserved variables respectively, and plates represent repeated variables. The hyperparameters are shared by all plates, but for clarity are shown connected only to the ith plate.

Each element of $\hat{\boldsymbol{X}}$ can thus be computed by

$$\hat{x}_{\boldsymbol{m}} = \sum_{k=1}^{K} \prod_{i=1}^{I} u^k_{im_i}, \tag{3}$$

where u^k_{im} denotes the mth element of $\boldsymbol{u}^k_i$. We view the vectors $\boldsymbol{u}^k_i$ as latent factors, and proceed by specifying a hierarchical Bayesian model.

2.1 Hierarchical Bayesian model

We view the data $\boldsymbol{X}$ as being produced according to the probabilistic generative process described in Figure 1. The observed data points, $x_{\boldsymbol{m}}$, are modelled using a Gaussian likelihood with variance ϑ and mean $\hat{x}_{\boldsymbol{m}}$ given by the decomposition in Eq. (3),

$$p(x_{\boldsymbol{m}}|\{u^k_{im}\}, \vartheta) = \mathcal{N}(x_{\boldsymbol{m}}|\hat{x}_{\boldsymbol{m}}, \vartheta) = \frac{1}{\sqrt{2\pi\vartheta}} \exp\left(-\frac{(x_{\boldsymbol{m}} - \hat{x}_{\boldsymbol{m}})^2}{2\vartheta}\right). \tag{4}$$

We choose a conjugate prior on the data variance, namely an inverse Gamma distribution with shape and scale parameters α and β,

$$p(\vartheta|\alpha, \beta) = \mathcal{IG}(\vartheta|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \vartheta^{-\alpha-1} \exp\left(\frac{-\beta}{\vartheta}\right). \tag{5}$$

We assume that the latent variables u^k_{im} are drawn from a rectified Gaussian prior with unknown mean μ^k_{im} and variance v^k_{im},

$$\begin{aligned} p\left(u^k_{im}\right) &= \mathcal{R}\left(u^k_{im}|\mu^k_{im}, v^k_{im}\right) \\ &= \frac{\sqrt{\frac{2}{\pi v^k_{im}}}}{\operatorname{erfc}\left(\frac{-\mu^k_{im}}{\sqrt{2v^k_{im}}}\right)} \exp\left(-\frac{(u^k_{im} - \mu^k_{im})^2}{2v^k_{im}}\right) h(u^k_{im}), \end{aligned} \tag{6}$$

where $h(x)$ is the Heaviside unit step function. This prior serves to enforce the non-negativity constraint, and is conjugate to the Gaussian likelihood.

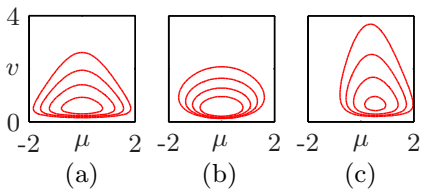

Figure 2: Illustration of different priors for μ and v where μ_μ=0 and v_μ=a=b=1. a) Independent Normal and inverse Gamma, $\mathcal{N}(\mu|\mu_\mu, v_\mu)\mathcal{IG}(v|a, b)$. b) Normal-inverse-Gamma, $\mathcal{N}(\mu|\mu_\mu, v_\mu v)\mathcal{IG}(v|a, b)$. c) Proposed prior in Eq. (8).

If the prior over u^k_{im} had been a Gaussian, appropriate conjugate priors for the mean μ^k_{im}, and variance v^k_{im}, would be a Gaussian and inverse Gamma; however, these priors are not conjugate to the rectified Gaussian, and instead we choose a convenient joint prior density,

$$\begin{aligned} p(\mu^k_{im}, v^k_{im}|\mu_\mu, v_\mu, a, b) = \frac{1}{c}\sqrt{v^k_{im}} \operatorname{erfc}\left(-\frac{\mu^k_{im}}{\sqrt{2v^k_{im}}}\right) \\ \times \mathcal{N}(\mu^k_{im}|\mu_\mu, v_\mu)\, \mathcal{IG}\left(v^k_{im}|a, b\right), \end{aligned} \tag{7}$$

where c is a normalization constant. With this prior, μ^k_{im} and v^k_{im} decouple, and the posterior conditional densities of μ^k_{im} and v^k_{im} are Gaussian and inverse Gamma respectively. This non-standard density is illustrated in Figure 2 and compared with two commonly used priors over mean and variance parameters.

We denote the set of all unknown variables in the model by $\boldsymbol{\theta} = \left\{\{\boldsymbol{u}^k_i\}, \vartheta, \{\boldsymbol{\mu}^k_i\}, \{\boldsymbol{v}^k_i\}\right\}$, and the set of hyperparemeters by $\boldsymbol{\Psi} = \{\alpha, \beta, a, b, \mu_\mu, v_\mu\}$. Following from the graphical model and Eq. (5–8) the joint probability of data and parameters is given by

$$\begin{aligned} p(\boldsymbol{X}, \boldsymbol{\theta}) &= p(\boldsymbol{X}|\{\boldsymbol{u}^k_i\}, \vartheta) p(\vartheta|\alpha, \beta) p(\{\boldsymbol{u}_i\}|\{\boldsymbol{\mu}^k_i\}, \{\boldsymbol{v}^k_i\}) \\ &\quad \times p(\{\boldsymbol{\mu}^k_i\}, \{\boldsymbol{v}^k_i\}|\mu_\mu, v_\mu, a, b) \\ &\propto \vartheta^{-\frac{M}{2}-\alpha-1} \prod_{\boldsymbol{m}\in\mathcal{M}} \exp\left\{-\frac{1}{2\vartheta}\left(x_{\boldsymbol{m}} - \sum_{k=1}^{K}\prod_{i=1}^{I} u^k_{im_i}\right)^2\right\} \\ &\quad \times \exp\left(\frac{-\beta}{\vartheta}\right) \prod_{k=1}^{K}\prod_{i=1}^{I}\prod_{m=1}^{M_i} \left\{\exp\left(-\frac{(u^k_{im} - \mu^k_{im})^2}{2v^k_{im}}\right) h(u^k_{im}) \right. \\ &\quad \left. \times \exp\left(-\frac{(\mu^k_{im} - \mu_\mu)^2}{2v_\mu}\right) (v^k_{im})^{-a-1} \exp\left(\frac{-b}{v^k_{im}}\right)\right\}. \end{aligned} \tag{8}$$

3. MARKOV CHAIN MONTE CARLO INFERENCE

3.1 Gibbs sampling

Gibbs sampling is one of the simplest MCMC techniques and plays a prominent role in modern Bayesian inference. The Gibbs sampler is widely applicable, particularly in the case where we deal with conditional distributions that have a parametric form that can easily be sampled from [7]. With our choice of conjugate priors and hyper priors, Gibbs sampling is particularly applicable for inference in the probabilistic NTF model.

In Gibbs sampling we assume that the latent variables in the model, $\boldsymbol{\theta}$, are partitioned in N groups, $\theta_1, \ldots, \theta_N$, and that it is possible to draw samples from the posterior conditional densities, $p(\theta_n|\boldsymbol{\theta}\backslash\theta_n)$, for each of these groups. Given some initial value of the parameters, we proceed by

iteratively sampling each group of latent variables, θ_n, while keeping all other variables fixed. This procedure forms a homogeneous Markov chain that can be shown to sample from the full posterior distribution. Gibbs sampling explores the posterior distribution in a random walk manner, and this may result in slow mixing of the Markov chain. Thus, in practice samples are highly correlated and a large number of iterations and subsampling is required in order to obtain independent samples from the target distribution.

To apply the Gibbs sampling procedure to probabilistic NTF we derive the relevant posterior conditional distributions, based on the joint distribution in Eq. (9). Our choice of conjugate priors simplifies this process, and implies the functional form of the posterior conditional distributions for all unknown variables. For each conditional posterior distribution we denote the posterior parameters by the same symbols as the prior parameters with a bar.

The conditional distribution for $u^{k'}_{i'm'}$ is a rectified Gaussian, $p(u^{k'}_{i'm'}|\boldsymbol{X},\boldsymbol{\theta}\backslash u^{k'}_{i'm'}) = \mathcal{R}(u^{k'}_{i'm'}|\bar{\mu}^{k'}_{i'm'},\bar{v}^{k'}_{i'm'})$ with variance and mean

$$\bar{v}^{k'}_{i'm'} = \left(\frac{1}{\vartheta}\sum_{\substack{\boldsymbol{m}\in\mathcal{M}\\ m_{i'}=m'}}\prod_{i\neq i'}\left(u^{k'}_{im_i}\right)^2 + \frac{1}{v^{k'}_{i'm'}}\right)^{-1}, \tag{9}$$

$$\begin{aligned}\bar{m}^{k'}_{i'm'} = \bar{v}^{k'}_{i'm'}\Bigg\{&\frac{1}{\vartheta}\sum_{\substack{\boldsymbol{m}\in\mathcal{M}\\ m_{i'}=m'}}\left(\sum_{k\neq k'}\prod_{i=1}^{I}u^{k}_{im_i} - x_{\boldsymbol{m}}\right)\\ &\times\prod_{i\neq i'}u^{k'}_{im_i} + \frac{\mu^{k'}_{i'm'}}{v^{k'}_{i'm'}}\Bigg\},\end{aligned} \tag{10}$$

from which it is possible to draw samples using standard methods such as inverse transform sampling. The mean of the density is unconstrained but samples are non-negative due to the rectification in the distribution. The conditional posterior distribution of the data variance is an inverse Gamma distribution, $p(\vartheta|\boldsymbol{X},\boldsymbol{\theta}\backslash\vartheta) = \mathcal{IG}(\vartheta|\bar{\alpha},\bar{\beta})$, with shape and scale

$$\bar{\alpha} = \alpha + \tfrac{M}{2} \qquad \bar{\beta} = \beta + \tfrac{1}{2}\chi^2, \tag{11}$$

where $\chi^2 = \|\boldsymbol{X} - \hat{\boldsymbol{X}}\|_F^2$ is the sum of squared errors. The conditional posterior distribution for $\mu^{k'}_{i'm'}$ is Gaussian, $p(\mu^{k'}_{i'm'}|\boldsymbol{X},\boldsymbol{\theta}\backslash\mu^{k'}_{i'm'}) = \mathcal{N}(\mu^{k'}_{i'm'}|\bar{m}_\mu,\bar{v}_\mu)$, with variance and mean

$$\bar{v}_\mu = \left(\frac{1}{v^k_{im}} + \frac{1}{v_\mu}\right)^{-1} \qquad \bar{m}_\mu = \bar{v}_\mu\left(\frac{u^{k'}_{i'm'}}{v^{k'}_{i'm'}} + \frac{\mu_\mu}{v_\mu}\right), \tag{12}$$

and the conditional posterior distribution for $v^{k'}_{i'm'}$ is an inverse Gamma, $p(v^{k'}_{i'm'}|\boldsymbol{X},\boldsymbol{\theta}\backslash v^{k'}_{i'm'}) = \mathcal{IG}(v^{k'}_{i'm'}|\bar{a},\bar{b})$, with shape and scale parameters

$$\bar{a} = a + \tfrac{1}{2} \qquad \bar{b} = b + \tfrac{1}{2}(u^{k'}_{i'm'} - \mu^{k'}_{i'm'})^2. \tag{13}$$

The latent variables in the probabilistic NTF model can thus be learned by sequentially drawing samples from these conditional densities.

3.2 Hamiltonian Markov chain Monte Carlo

A second MCMC sampling method is Hamiltonian Markov chain Monte Carlo (HMC) [8, 7], which is a suitable sampler for use with this model since all the variables are continuous and it is possible to compute the derivative of the log joint probability. HMC is also an attractive scheme for sampling since it avoids the random walk behaviour of the Metropolis or the Gibbs sampling algorithms [7].

HMC is an auxiliary variable sampler that uses gradient information to improve mixing. The gradient acts as a force that causes the sampler to explore the sample space more effectively. The gradient acts on the momentum $\boldsymbol{q}$, of the system which is included as the auxiliary variable, such that we sample from the augmented distribution $p(\boldsymbol{\theta},\boldsymbol{q}|\boldsymbol{X})$ rather than the target distribution $p(\boldsymbol{\theta}|\boldsymbol{X})$. The sampling requires that we are able to compute a potential and kinetic energy as well as the gradient of the potential energy with respect to the sampling variables. The potential energy function is the negative log joint probability of the probabilistic NTF model, $\mathcal{E}(\boldsymbol{\theta}|\boldsymbol{\Psi}) = -\ln p(\boldsymbol{X},\boldsymbol{\theta}|\boldsymbol{\Psi})$, given by the negative logarithm of Eq. (9). The auxiliary momentum variable $\boldsymbol{q}$ is Gaussian and is used to define the kinetic energy $\boldsymbol{K}(\boldsymbol{q}) = \frac{1}{2}\boldsymbol{q}^\top\boldsymbol{q}$, and the gradients $\boldsymbol{\Delta\theta} \triangleq \frac{\partial\mathcal{E}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}$ can be derived from Eq. (9). The sum of the kinetic and the potential energy defines the Hamiltonian $\mathcal{H}$. Samples of $\boldsymbol{\theta}$ and $\boldsymbol{q}$ are obtained by combining the Hamiltonian with the gradient information in the simulation of so-called "leapfrog" steps which simulate the Hamiltonian dynamics. We defer these details and the general pseudocode for HMC to the works of MacKay [9] and Neal [7].

3.2.1 Change of variables

To simplify the HMC sampling procedure, we ensure that the simulation dynamics for the model parameters are performed in an unconstrained space. For probabilistic NTF, the parameters $u^k_{im} \geq 0$, $\vartheta \geq 0$ and $v^k_{im} \geq 0$ can be transformed to unconstrined variables using the transformations: $u^k_{im} = \exp(\tilde{u}^k_{im})$, $\vartheta = \exp(\tilde{\vartheta})$, and $v^k_{im} = \exp(\tilde{v}^k_{im})$. These changes of variables requires the joint probability in Eq. (9) to be multiplied by the Jacobian determinants of the transformations, which are given by

$$\left|\tfrac{\partial u^k_{im}}{\partial\tilde{u}^k_{im}}\right| = \exp(\tilde{u}^k_{im}),\ \left|\tfrac{\partial\vartheta}{\partial\tilde{\vartheta}}\right| = \exp(\tilde{\vartheta}),\ \left|\tfrac{\partial v^k_{im}}{\partial\tilde{v}^k_{im}}\right| = \exp(\tilde{v}^k_{im}). \tag{14}$$

Including the Jacobian terms and computing the negative of the logarithm of Eq. (9), we arrive at the following negative log joint probability density

$$\begin{aligned}\mathcal{L} = &\frac{1}{2\exp(\tilde{\vartheta})}\sum_{\boldsymbol{m}\in\mathcal{M}}\left(x_{\boldsymbol{m}} - \sum_{k=1}^{K}\prod_{i=1}^{I}\exp(\tilde{u}^k_{im_i})\right)^2\\ &+\left(\frac{M}{2}+\alpha\right)\tilde{\vartheta} + \frac{\beta}{\exp(\tilde{\vartheta})}\\ &+\sum_{k=1}^{K}\sum_{i=1}^{I}\sum_{m=1}^{M_i}\left\{\frac{\left(\exp(\tilde{u}^k_{im}) - \mu^k_{im}\right)^2 + 2b}{2\exp(\tilde{v}^k_{im})}\right.\\ &\left.+\frac{(\mu^k_{im}-\mu_\mu)^2}{2v_\mu} + a\tilde{v}^k_{im} - \tilde{u}^k_{im}\right\}.\end{aligned} \tag{15}$$

3.2.2 Derivatives

Using Eq. (16) we can compute the required derivatives $\boldsymbol{\Delta\theta}$ for HMC sampling. The derivative of the negative log joint probability density w.r.t. the variable $\tilde{u}^{k'}_{i'm'}$ is given by

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\tilde{u}^{k'}_{i'm'}} = &\sum_{\substack{\boldsymbol{m}\in\mathcal{M}\\ m_{i'}=m'}}\left(\sum_{k=1}^{K}\prod_{i=1}^{I}\exp(\tilde{u}^k_{im_i}) - x_{\boldsymbol{m}}\right)\\ &\times\frac{1}{\exp(\tilde{\vartheta})}\prod_{i=1}^{I}\exp(\tilde{u}^{k'}_{im_i})\\ &+\frac{\exp(\tilde{u}^{k'}_{i'm'}) - \mu^{k'}_{i'm'}}{\exp(\tilde{v}^{k'}_{i'm'})}\exp(\tilde{u}^{k'}_{i'm'}) - 1.\end{aligned} \tag{16}$$

Recalling that χ^2 is the sum of squared errors, defined previously for Eq. (12), the derivative w.r.t the likelihood variance is given by

$$\frac{\partial \mathcal{L}}{\partial \tilde{\vartheta}} = -\frac{\frac{1}{2}\chi^2 + \beta}{\vartheta} + \frac{M}{2} + \alpha. \tag{17}$$

The remaining derivatives are those w.r.t the unknown mean and variance of the latent variables given by

$$\frac{\partial \mathcal{L}}{\partial \mu^{k'}_{i'm'}} = -\frac{u^{k'}_{i'm'} - \mu^{k'}_{i'm'}}{v^{k'}_{i'm'}} + \frac{\mu^{k'}_{i'm'} - \mu_\mu}{v_\mu}, \tag{18}$$

$$\frac{\partial \mathcal{L}}{\partial \tilde{v}^{k'}_{i'm'}} = -\frac{\frac{1}{2}(u^{k'}_{i'm'} - \mu^{k'}_{i'm'})^2 + b}{v^{k'}_{i'm'}} + a. \tag{19}$$

3.3 Notes on computation

The major part of the computational complexity in the Gibbs sampler and the HMC lies in computing Eq. (10–11) and Eq. (17) respectively, which are the only expressions that require a sum over all data points. In the following, we only discuss the efficient computation of Eq. (17) for HMC, but note that the computations for the Gibbs sampler are similar.

The two central terms in the computation are inner products between the model factors and the data tensor as well as the approximated data tensor. The first term can be computed efficiently by forming the outer product of $\boldsymbol{u}^k_{im}$ for all i except i', and computing a tensor product

$$\sum_{\substack{\boldsymbol{m}\in\mathcal{M}\\ m_{i'}=m'}} x_{\boldsymbol{m}} \prod_{i=1}^{I} u^{k'}_{im_i} = \Big[\boldsymbol{u}^{k'}_{i'} \odot \big(\boldsymbol{X} \bigotimes_{i\neq i'} \boldsymbol{u}^{k'}_i\big)\Big]_{m'} \tag{20}$$

where $\odot$ denotes the elementwise product. This can be implemented using efficient standard routines for matrix multiplication and has complexity $O(MKI)$ to compute for the whole model. The second term can be computed efficiently as

$$\sum_{\substack{\boldsymbol{m}\in\mathcal{M}\\ m_{i'}=m'}} \left(\sum_{k=1}^{K}\prod_{i=1}^{I} u^k_{im_i}\right) \prod_{i=1}^{I} u^{k'}_{im_i} = \Big[\boldsymbol{U}_{i'} \odot \big(\boldsymbol{U}_{i'} \bigodot_{i\neq i'} \boldsymbol{U}_i^\top \boldsymbol{U}_i\big)\Big]_{m'k'} \tag{21}$$

where $\boldsymbol{U}_i = [\boldsymbol{u}^1_i, \ldots, \boldsymbol{u}^K_i]$, and it has a computational complexity of $O(\sum_i M_i K^2)$ to compute for the whole model. Note that this formulation avoids the explicit computation of the approximation to the data tensor, and thus allows the algorithms to scale to large data tensors without excessive memory requirements.

4. EXPERIMENTS

We evaluate the performance of the proposed probabilistic NTF methods on two publicly aviailable food science data sets: The first data set is five-way tensor of measurements of the color of fresh beef [10] as it changes due to storage conditions. The second data set is a three-way tensor of sensory profiles of bread [11]. Detailed descriptions of the data sets are available at `http://www.models.kvl.dk/research/data`. We compare the reconstructive ability of the probabilistic NTF techniques to that of the non-negative PARAFAC model, which we compute using the algorithm of Bro et al. [12] implemented in the N-Way Toolbox [13].

For each dataset, we separate the available data into training and test data. The test data is created by randomly selecting 10% of the data points and setting them as missing data in the training set. We repeat this process to create 10 such datasets. We do inference in the presence of missing data for a range of different model orders, and compute root mean squared error (RMSE) on the held out data.

Table 1: Root mean squared error and standard deviation results on *color of beef* and *sensory bread* data for PARAFAC and probabilistic NTF using different model orders.

Data	K	PARAFAC	Probabilistic NTF
Color of beef	1	2.90 ± 0.43	2.90 ± 0.43
	2	1.66 ± 0.57	1.50 ± 0.23
	3	$\mathbf{1.52 \pm 0.38}$	1.53 ± 0.31
	4	2.54 ± 2.92	1.53 ± 0.36
	5	2.16 ± 1.13	$\mathbf{1.47 \pm 0.45}$
	6	1.99 ± 1.30	1.47 ± 0.33
	7	2.21 ± 0.63	1.54 ± 0.37
Sensory bread	1	1.50 ± 0.11	1.50 ± 0.11
	2	1.32 ± 0.10	1.30 ± 0.09
	3	1.23 ± 0.09	1.22 ± 0.09
	4	1.26 ± 0.07	1.22 ± 0.08
	5	$\mathbf{1.21 \pm 0.08}$	1.19 ± 0.09
	6	1.23 ± 0.08	1.17 ± 0.09
	7	1.26 ± 0.11	1.17 ± 0.09
	8	1.34 ± 0.30	$\mathbf{1.16 \pm 0.09}$

Missing data is easily handled in the inference procedure for the Bayesian NTF model by excluding the missing elements in the likelihood term, which corresponds to dealing with the "missing at random" data assumption. For the non-negative PARAFAC model, missing data is handled by coupling the learning with built-in EM iterations [14].

HMC has two free parameters, the step size δ, and the number of leapfrog steps τ. The selection of these parameters is a design choice. A large step sizes causes the sampler to make large steps in the sample space and may result in oscillatory sampling behaviour. Small step sizes cause the sampler to take very small steps and may result in slow convergence, requiring a large number of iterations. In general we choose the step-size to ensure that the reject rate is less than 25%. It is also preferable to have a large number of leapfrog steps since this reduces the random walk behaviour of the sampling [7]. In our experiments we used $\tau = 20$ leapfrog steps of size $\delta = 0.001$.

We computed 50,000 samples and discarded the first half to allow for the samplers to burn in. Results using the Gibbs sampler and the HMC were similar; thus, for compactness we only show results obtained using the Gibbs sampler. We found that HMC exhibited faster convergence in terms of iterations but slower in terms of computation time, since each iteration of the HMC is approximately τ times slower than a Gibbs sweep.

Results for the two data sets are given in Table 1 and Figure 3. In the color of beef data, the non-negative PARAFAC model predicts missing data well for model orders $K = 2$ and $K = 3$ in accordance with previous results on this data set [10]. For larger model orders, however, the PARAFAC model overfits, evidenced by a decreasing training error and increasing test error. The probabilistic NTF model predicts missing data equally well or better at all model orders, and does not overfit. In the sensory bread data, the results for the non-negative PARAFAC model suggest that the data is reasonably modelled using around 3–7 components. The probabilistic NTF model predicts missing data better than non-negative PARAFAC for all model orders and does not lead to overfitting. Figure 4 shows autocorrelation coefficients of samples computed using Gibbs sampling and HMC

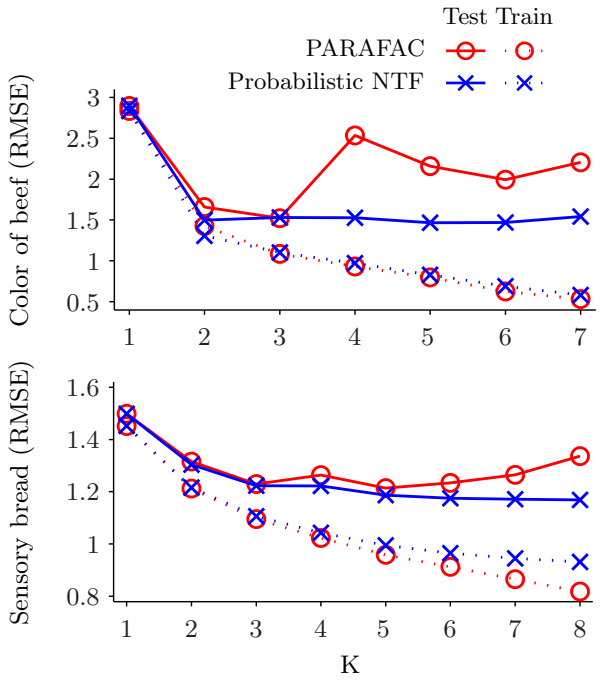

Figure 3: Root mean squared error (RMSE) on *color of beef* and *sensory bread* data for PARAFAC and probabilistic NTF using different model orders. Solid and dotted lines indicate held-out test data and training data respectively.

on the sensory bread data, indicating that the HMC sampler mixes slightly better than the Gibbs sampler. The figure also shows the root mean squared error on the training data as a function of the iteration number, which indicates that both algorithms quickly converge to sample from a high density region of the posterior.

5. CONCLUSIONS

We have presented a model for probabilistic non-negative matrix factorization. The model is formulated by considering each of the tensor factors as a latent variable in a probabilistic graphical model. We have described two MCMC inference procedures: Gibbs sampling and Hamiltonian Monte Carlo. Both inference methods provide the same level of performance and have been shown to provide consistent predictions even as the number of tensor factors is increased. Probabilistic NTF is able to avoid problems of overfitting which is common in the standard non-negative PARAFAC model and other maximum likelihood learning approaches.

Future work includes a more rigorous comparison of the two sampling methods to evaluate which may be better for practical applications of this model. Other work includes exploring methods for the automatic discovery of the number of tensor factors through other MCMC techniques such as reversible jump MCMC or by non-parametric modelling.

REFERENCES

[1] P. Paatero and U. Tapper, "Positive matrix factorization: A nonnegative factor model with optimal utilization of error-estimates of data values," *Environmetrics*, vol. 5, pp. 111–126, Jun 1994.

[2] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[3] J. D. Carroll and J. J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart-Young decomposition," *Psychometrika*, vol. 35, pp. 283–319, 1970.

[4] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.

[5] M. Welling and M. Weber, "Positive tensor factorization," *Pattern Recognition Letters*, vol. 22, no. 12, pp. 1255–1261, 2001.

[6] A. Shashua and T. Hazan, "Non-negative tensor factorization with applications to statistics and computer vision," in *Machine Learning, International Conference on (ICML)*, pp. 792–799, Aug 2005.

[7] R. M. Neal, "Probabilistic inference using Markov Chain Monte Carlo methods," tech. rep., Dept. of Computer Science, University of Toronto, 1993.

[8] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, "Hybrid Monte Carlo," *Physics Letters B*, vol. 195, pp. 216–222, Sep 1987.

[9] D. J. C. MacKay, *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, June 2003.

[10] R. Bro and M. Jakobsen, "Exploring complex interactions in designed data using GEMANOVA. Color changes in fresh beef during storage," *Chemometrics, Journal of*, vol. 16, pp. 294–304, May 2002.

[11] R. Bro, *Multi-way Analysis in the Food Industry*. PhD thesis, University of Amsterdam and Royal Veterinary and Agricultural University, 1998.

[12] R. Bro and S. de Jong, "A fast non-negativity-constrained least squares algorithm," *Journal of Chemometrics*, vol. 11, pp. 393–401, Apr 1999.

[13] C. Andersson and R. Bro, "The N-way toolbox for MATLAB," *Chemometrics & Intelligent Laboratory Systems*, vol. 52, pp. 1–4, Aug 2000.

[14] G. Tomasi and R. Bro, "PARAFAC and missing values," *Chemometrics and Intelligent Laboratory Systems*, vol. 75, pp. 163–180, Feb 2002. 10.1016/j.chemolab.2004.07.003.

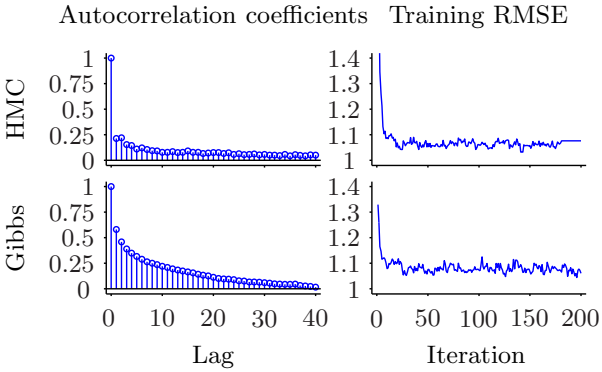

Figure 4: Left: Averaged autocorrelation coefficients of samples of u_{im}^k computed using Gibbs sampling and HMC ($\tau = 100$) on the sensory bread data. Right: Root mean squared error on the training set as a function of number of iterations for a five-component probabilistic NTF model.