MARKOV CHAIN MONTE CARLO INFERENCE FOR PROBABILISTIC LATENT TENSOR FACTORIZATION

Umut Şimşekli, A. Taylan Cemgil

Dept. of Computer Engineering
Boğaziçi University
34342, Bebek, İstanbul, Turkey

ABSTRACT

Probabilistic Latent Tensor Factorization (PLTF) is a recently proposed probabilistic framework for modeling multiway data. Not only the popular tensor factorization models but also any arbitrary tensor factorization structure can be realized by the PLTF framework. This paper presents Markov Chain Monte Carlo procedures (namely the Gibbs sampler) for making inference on the PLTF framework. We provide the abstract algorithms that are derived for the general case and the overall procedure is illustrated on both synthetic and real data.

***Index Terms*—** Probabilistic Latent Tensor Factorization (PLTF), Markov Chain Monte Carlo (MCMC), Space Alternating Data Augmentation (SADA)

1. INTRODUCTION

Factorization based data modeling has become popular together with the advances in the computational power. Nonnegative Matrix Factorization (NMF) model, proposed by Lee and Seung [1], is one of the most popular factorization models where the aim is to estimate the matrices Z_1 and Z_2 as the matrix X is observed:

$$X(i,j) \approx \hat{X}(i,j) = \sum_k Z_1(i,k)Z_2(k,j). \quad (1)$$

Here X, Z_1, and Z_2 are all non-negative matrices. This modeling paradigm has found place in many fields including audio/music processing, image processing, and bioinformatics [2, 3, 4].

Although the NMF model has its own advantages, certain applications require more structured modeling and incorporation of prior knowledge where NMF can be inadequate. Accordingly, several complex factorization models have been proposed in the literature [4]. The Probabilistic Latent Tensor Factorization framework (PLTF) [5] enables one to incorporate domain specific information to any arbitrary factorization model and provides the update rules for multiplicative gradient descent and expectation-maximization algorithms.

The PLTF framework is defined as a natural extension of the matrix factorization model of (1):

$$X(v_0) \approx \hat{X}(v_0) = \sum_{\bar{v}_0} \prod_\alpha Z_\alpha(v_\alpha), \quad (2)$$

where $\alpha = 1 \dots K$ denotes the factor index. Here the aim is computing an approximate factorization of a given a multiway array X in terms of a product of individual factors Z_α, some of which are possibly fixed. The product $\prod_\alpha Z_\alpha(v_\alpha)$ is summed over a set of indices which makes the factorization latent.

In the PLTF framework, each tensor is described by an index set. Here we define V as the set of all indices in a model, V_0 as the set of visible indices, V_α as the set of indices in Z_α, and $\bar{V}_\alpha = V - V_\alpha$ as the set of all indices not in Z_α. We use small letters as v_α to refer to a particular setting of indices in V_α. For example, the NMF model of [1], introduced in (1), can be defined in the PLTF framework by selecting the index sets as $V = \{i,j,k\}$, $V_0 = \{i,j\}$, $V_1 = \{i,k\}$, and $V_2 = \{k,j\}$.

In this paper, we present Markov Chain Monte Carlo procedures (namely the Gibbs sampler) for making inference on the PLTF framework. We first provide a more conventional sampling schema, and then we describe how the sampling algorithm can be made more efficient by making use of space alternating data augmentation (SADA) [6]. We also describe how the marginal likelihood of a tensor factorization model can be estimated by using Chib's method. Finally, we illustrate our method on both synthetic and real data.

1.1. Probability Model

The usual approach to estimate the factors Z_α is trying to find the optimal $Z^*_{1:K} = \underset{Z_{1:K}}{\arg\min}\, d(X||\hat{X})$, where $d(\cdot)$ is a divergence typically taken as Euclidean, Kullback-Leibler or

Funded by the scientific and technological research council of Turkey (TÜBİTAK) grant number 110E292, project Bayesian matrix and tensor factorizations (BAYTEN). Umut Şimşekli is also supported by a Ph.D. scholarship from TÜBİTAK.

978-1-4673-1026-0/12/$31.00 ©2012 IEEE

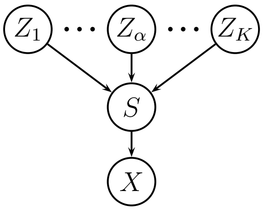

Fig. 1. The generative model of the PLTF framework as a Bayesian network. The directed acyclic graph describes the dependency structure of the variables: the full joint distribution can be written as $p(X, S, Z_{1:K}) = p(X|S)p(S|Z_{1:K}) \prod_\alpha p(Z_\alpha)$.

Itakura-Saito divergences. Since the analytical solution for this problem is intractable, one should refer to iterative or approximate inference methods.

In this study, we use the Kullback-Leibler (KL) divergence as the cost function which is equivalent to selecting the Poisson observation model [3, 5], while our approach can be extended to other observation models where a composite structure is present. The overall probabilistic model is defined as follows:

$$\begin{aligned} Z_\alpha(v_\alpha) &\sim \mathcal{G}(Z_\alpha(v_\alpha); A_\alpha(v_\alpha), B_\alpha(v_\alpha)) && \text{factor priors} \\ \Lambda(v) &= \prod_\alpha Z_\alpha(v_\alpha) && \text{intensity} \\ S(v) &\sim \mathcal{PO}(S(v); \Lambda(v)) && \text{components} \\ X(v_0) &= \sum_{\bar{v}_0} S(v) && \text{observation} \\ \hat{X}(v_0) &= \sum_{\bar{v}_0} \Lambda(v) && \text{parameter} \end{aligned}$$

where the symbols $\mathcal{PO}$ and $\mathcal{G}$ symbols refer to Poisson and Gamma distributions respectively, where

$$\mathcal{PO}(s; \lambda) = e^{-\lambda} \frac{\lambda^s}{s!} \tag{3}$$

$$\mathcal{G}(z; a, b) = e^{-bz} \frac{z^{a-1} b^a}{\Gamma(a)}. \tag{4}$$

The Gamma prior on the factors are chosen in order to preserve conjugacy. The graphical model for the PLTF framework is depicted in Fig 1. Note that $p(X|S)$ is a degenerate distribution that is defined as follows:

$$p(X|S) = \prod_{v_0} \delta \left(X(v_0) - \sum_{\bar{v}_0} S(v) \right). \tag{5}$$

Here, $\delta(\cdot)$ is the Kronecker delta function where $\delta(x) = 1$ when $x = 0$ and $\delta(x) = 0$ otherwise.

The rest of the paper is organized as follows: We describe the MCMC procedures and the method for estimating marginal likelihood ($\int dZ_{1:K} p(X|Z_{1:K}) p(Z_{1:K})$) in Section 2. In Section 3, we illustrate the proposed approach on three different factorization models. Section 4 concludes this paper.

2. MARKOV CHAIN MONTE CARLO, THE GIBBS SAMPLER

Monte Carlo methods are a set of numerical techniques to estimate expectations of the form:

$$\langle \varphi(x) \rangle_{\pi(x)} \approx \frac{1}{N} \sum_{n=1}^{N} \varphi(x^{(i)}) \tag{6}$$

where $x^{(i)}$ are independent samples drawn from the target $\pi(x)$. Under mild conditions on the test function φ, the estimate converges to the true expectation for $N \to \infty$. The difficulty here is obtaining independent samples from a non-standard target density π.

The Markov Chain Monte Carlo (MCMC) techniques generate subsequent samples from a Markov chain defined by a transition kernel $\mathcal{T}$, that is, one generates $x^{(i+1)}$ conditioned on $x^{(i)}$ as follows:

$$x^{(i+1)} \sim \mathcal{T}(x|x^{(i)}). \tag{7}$$

The transition kernel $\mathcal{T}$ is not needed explicitly in practice; all is needed is a procedure to sample a new configuration, given the previous one. Perhaps surprisingly, even though subsequent samples are correlated, provided that $\mathcal{T}$ satisfies certain ergodicity conditions, (6) remains still valid, and estimated expectations converge to their true values when number of steps i goes to infinity. To design a transition kernel $\mathcal{T}$ such that the desired distribution is the stationary distribution, that is, $\pi(x) = \int \mathcal{T}(x|x')\pi(x')dx'$, many alternative strategies can be employed; the most popular one being the Metropolis-Hastings (MH) algorithm [7]. One particularly convenient and simple MH strategy is the Gibbs sampler where one samples each block of variables from the so called full conditional distributions. The Gibbs sampler for the PLTF model may be formed by iteratively drawing samples from the full conditional distributions as follows:

$$S^{(i+1)} \sim p(S|Z_{1:K}^{(i)}, X, \Theta) \tag{8}$$

$$Z_\alpha^{(i+1)} \sim p(Z_\alpha|S^{(i)}, Z'_{\neg\alpha}, X, \Theta) \quad \alpha = 1 \ldots K \tag{9}$$

where $Z'_{\neg\alpha}$ denotes the most recent values of all the factors but Z_α, Θ denotes the prior distribution parameters $\{A_\alpha, B_\alpha\}_{\alpha=1}^K$, and the full conditionals are defined as:

$$p(S|\cdot) = \prod_{v_0} \mathcal{M} \left(S(v_0, \bar{V}_0); X(v_0), \frac{\Lambda(v_0, \bar{V}_0)}{\hat{X}(v_0)} \right) \tag{10}$$

$$p(Z_\alpha|\cdot) = \prod_{v_\alpha} \mathcal{G} \left(Z_\alpha(v_\alpha); \Sigma_\alpha(v_\alpha), \Phi_\alpha(v_\alpha) \right) \tag{11}$$

Algorithm 1 Block Gibbs Sampler for PLTF

```
Input: Observed data $X$, $\Theta$
Initialize factors: $Z_\alpha^{(0)} \sim \mathcal{G}(Z_\alpha; A_\alpha, B_\alpha)\ \forall\alpha = 1 \dots K$
for $i = 1 \dots$ MAXITER do
    Compute the intensity and parameter tensors:
    $\Lambda(v) = \prod_\alpha Z_\alpha^{(i-1)}(v_\alpha)$
    $\hat{X}(v_0) = \sum_{\bar{v}_0} \Lambda(v)$
    Sample Sources:
    for all $v_0 \in V_0$ do
        $S(v_0, \bar{V}_0)^{(i)} \sim \mathcal{M}\left(\cdot; X(v_0), \frac{\Lambda(v_0,\bar{V}_0)}{\hat{X}(v_0)}\right)$
    end for
    Sample Factors:
    for $\alpha = 1 \dots K$ do
        for all $v_\alpha \in V_\alpha$ do
            $\Sigma_\alpha = A_\alpha(v_\alpha) + \sum_{\bar{v}_\alpha} S^{(i)}(v)$
            $\Phi_\alpha = B_\alpha(v_\alpha) + \sum_{\bar{v}_\alpha} \prod_{\alpha' \neq \alpha} Z'_{\alpha'}(v_{\alpha'})$
            ($Z'_\alpha$ refers to the most recent value of $Z_\alpha$)
            $Z_\alpha^{(i)}(v_\alpha) \sim \mathcal{G}\left(Z_\alpha(v_\alpha); \Sigma_\alpha, \Phi_\alpha\right)$
        end for
    end for
end for
```

where

$$\Sigma_\alpha(v_\alpha) = A_\alpha(v_\alpha) + \sum_{\bar{v}_\alpha} S(v) \tag{12}$$

$$\Phi_\alpha(v_\alpha) = B_\alpha(v_\alpha) + \sum_{\bar{v}_\alpha} \prod_{\alpha' \neq \alpha} Z_{\alpha'}(v_{\alpha'}). \tag{13}$$

Here, Λ is the intensity tensor that is defined in Section 1.1, $\mathcal{G}$ is the Gamma distribution and $\mathcal{M}$ refers to the Multinomial distribution that is defined as follows:

$$\mathcal{M}(\mathbf{s}; x, \mathbf{p}) = \delta(x - \sum_i s_i) x! \prod_{i=1}^{I} \frac{p_i^{s_i}}{s_i!} \tag{14}$$

where $\mathbf{s} = \{s_1, \dots, s_I\}$ and $\mathbf{p} = \{p_1, \dots, p_I\}$. Verbally, given a particular instance of observed indices v_0, the full conditional of S is a Multinomial distribution over all the latent indices $\bar{V}_0$.

Note that, it can easily be verified that the Gibbs sampler for the NMF model that is presented in [3] is a special case of our method. The pseudo-code is given in Algorithm 1 and this procedure will be illustrated with an example in Section 3.

2.1. Efficient Inference with Space Alternating Data Augmentation

Space alternating data augmentation (SADA) was first presented in [8] for making inference in Gaussian mixture models. In a recent work [6], Fevotte et al. presented a MCMC procedure with SADA for making inference in composite models including NMF. In this section we will generalize this procedure to the PLTF framework.

The main idea behind SADA is sampling each slice of the components from their marginal distribution instead of sampling all the slices from their full conditional at the same time. This approach significantly reduces the memory requirements of a sampler since it only requires storing $|V_0|$ elements instead of $|V|$ elements of the latent components at each iteration of the sampling procedure.

Applying the SADA algorithm to the PLTF framework is not straightforward since the index structure for different models can lead to different conditional independence structures. Therefore, we rewrite the original PLTF model (2) as a collection of several 'marginal' models, one for each latent factor Z_α as follows:

$$\hat{X}(v_0) = \sum_{\bar{v}_0 \cap v_\alpha} \underbrace{Z_\alpha(v_\alpha) \sum_{\bar{v}_0 \cap \bar{v}_\alpha} \prod_{\alpha' \neq \alpha} Z_{\alpha'}(v_{\alpha'})}_{\equiv \Lambda_\alpha(\lambda_\alpha)} \tag{15}$$

where $\lambda_\alpha = v \setminus (\bar{v}_0 \cap \bar{v}_\alpha)$. We also define $S_\alpha(\lambda_\alpha)$ by using the additivity property of Poisson distribution:

$$S_\alpha(\lambda_\alpha) \sim \mathcal{PO}(S_\alpha(\lambda_\alpha); \Lambda_\alpha(\lambda_\alpha)) \tag{16}$$

$$X(v_0) = \sum_{\bar{v}_0 \cap v_\alpha} S_\alpha(\lambda_\alpha). \tag{17}$$

In the SADA algorithm, each slice of S_α is drawn from its marginal distribution and then each Z_α is drawn by conditioning on S_α. Curious reader is referred to [6] for a detailed description and the proof of convergence for the matrix case. The pseudo-code is given in Algorithm 2. Note that the $\mathcal{BI}$ symbol in the pseudocode refers to the Binomial distribution:

$$\mathcal{BI}(s; x, p) = \frac{x!}{s!(x-s)!} p^s (1-p)^{(x-s)}. \tag{18}$$

2.2. Marginal Likelihood Estimation with Chib's Method

The marginal likelihood of the observed data under a tensor factorization model $p(X)$ is often necessary for certain problems such as model selection. This quantity can be estimated from the Gibbs output and it is known as the Chib's method [9]. This method is applied in [3] for NMF; here we generalize it in order to estimate the marginal likelihood for the PLTF framework.

Suppose the Gibbs sampler has been run until convergence and we have N samples for each variable. The marginal likelihood is defined as:

$$p(X) = \frac{p(S, Z_{1:K}, X)}{p(S, Z_{1:K}|X)}. \tag{19}$$

This equation holds for all points $(S, Z_{1:K})$. Provided that the distribution is unimodal, a good candidate point in the configuration space is a configuration near the mode $(\tilde{S}, \tilde{Z}_{1:K})$.

Algorithm 2 SADA Sampler for PLTF

```
Input: Observed data $X$, $\Theta$
Initialize factors: $Z_\alpha^{(0)} \sim \mathcal{G}(Z_\alpha; A_\alpha, B_\alpha)\ \forall \alpha = 1 \dots K$
for $i = 1 \dots$ MAXITER do
    for $\alpha = 1 \dots K$ do
        for all $v_\alpha \in V_\alpha$ do
            $\hat{X}(v_0) = \sum_{\bar{v}_0} \prod_\alpha Z'_\alpha(v_\alpha)$
            Sample Slices of Components
            for all $v_0 \in V_0$ do
                $\Lambda_\alpha(v_\alpha \cup v_0) = \sum_{\bar{v}_\alpha \cap \bar{v}_0} \prod_{\alpha'} Z'_{\alpha'}(v_{\alpha'})$
                $S_\alpha^{(i)}(v_\alpha \cup v_0) \sim \mathcal{BI}\left(\cdot; X(v_0), \frac{\Lambda_\alpha(v_\alpha \cup v_0)}{\hat{X}(v_0)}\right)$
            end for
            Sample Factors
            $\Sigma_\alpha = A_\alpha(v_\alpha) + \sum_{\bar{v}_\alpha} S_\alpha^{(i)}(\lambda_\alpha)$
            $\Phi_\alpha = B_\alpha(v_\alpha) + \sum_{\bar{v}_\alpha} \prod_{\alpha' \neq \alpha} Z'_{\alpha'}(v_{\alpha'})$
            ($Z'_\alpha$ refers to the most recent value of $Z_\alpha$)
            $Z_\alpha^{(i)}(v_\alpha) \sim \mathcal{G}\left(Z_\alpha(v_\alpha); \Sigma_\alpha, \Phi_\alpha\right)$
        end for
    end for
end for
```

The numerator (the full joint distribution) is straightforward to evaluate. We can expand the denominator as follows:

$$
\begin{aligned}
p(\tilde{S}, \tilde{Z}_{1:K}|X) =& p(\tilde{Z}_1|\tilde{Z}_{2:K}, \tilde{S}) p(\tilde{Z}_2|\tilde{Z}_{3:K}, \tilde{S}) \dots \\
& p(\tilde{Z}_{K-1}|\tilde{Z}_K, \tilde{S}) p(\tilde{Z}_K|\tilde{S}) p(\tilde{S}|X) \qquad (20) \\
=& p(\tilde{S}|X) p(\tilde{Z}_1|\tilde{Z}_{2:K}, \tilde{S}) \\
& \prod_{\alpha=2}^{K} p(\tilde{Z}_\alpha|\tilde{Z}_{\alpha+1:K}, \tilde{S}), \qquad (21)
\end{aligned}
$$

where $p(\tilde{Z}_K|\tilde{Z}_{K+1}, \tilde{S}) = p(\tilde{Z}_K|\tilde{S})$. The ordering of the variables at this expansion step can be changed, however without loss of generality we assume that the ordering is $Z_1 \dots Z_K$.

The term $p(\tilde{Z}_1|\tilde{Z}_{2:K}, \tilde{S})$ is full conditional, so it is available for the Gibbs sampler. We can also approximate $p(\tilde{S}|X)$ as:

$$
\begin{aligned}
p(\tilde{S}|X) &= \int dZ_{1:K}\, p(\tilde{S}|Z_{1:K}, X) p(Z_{1:K}|X) \qquad (22) \\
&\approx \frac{1}{N} \sum_{i=1}^{N} p(\tilde{S}|Z_{1:K}^{(i)}, X). \qquad (23)
\end{aligned}
$$

Evaluating the term $p(\tilde{Z}_\alpha|\tilde{Z}_{\alpha+1:K}, \tilde{S})$ is more complicated. Firstly, we start by rewriting the term $p(\tilde{Z}_K|\tilde{S})$ as:

$$
p(\tilde{Z}_K|\tilde{S}) = \int dZ_{1:K-1}\, p(\tilde{Z}_K|Z_{1:K-1}, \tilde{S}) p(Z_{1:K-1}|\tilde{S}) \qquad (24)
$$

The first term here is again full conditional. However, we do not have samples from the distribution $p(Z_{1:K-1}|\tilde{S})$ since the sampler gives us samples from $p(Z_{1:K-1}|X)$. The solution is approximating this term by running the Gibbs sampler M more iterations and clamping S at $\tilde{S}$: $(Z_{1:K}^{(N+m)}) \sim p(Z_{1:K}|S = \tilde{S})$. The estimate is as follows:

$$
p(\tilde{Z}_K|\tilde{S}) \approx \frac{1}{M} \sum_{m=N+1}^{N+M} p(\tilde{Z}_K|Z_{1:K-1}^{(m)}, \tilde{S}). \qquad (25)
$$

We can apply the same idea to the rest of the terms in (21) by clamping some of the factors and running the sampler M more iterations for each $\alpha = (K-1), \dots, 2$. The resulting estimation is as follows:

$$
\begin{aligned}
& p(\tilde{Z}_\alpha|\tilde{Z}_{\alpha+1:K}, \tilde{S}) \\
& = \int dZ_{1:\alpha-1}\, p(\tilde{Z}_\alpha|Z_{1:\alpha-1}, \tilde{Z}_{\alpha+1:K}, \tilde{S}) p(Z_{1:\alpha-1}|\tilde{Z}_{\alpha+1:K}\tilde{S}) \\
& \approx \frac{1}{M} \sum_{m=l_\alpha}^{u_\alpha} p(\tilde{Z}_\alpha|Z_{1:\alpha-1}^{(m)}, \tilde{Z}_{\alpha+1:K}, \tilde{S}) \qquad (26)
\end{aligned}
$$

where l_α and u_α denote the first and the last indices of the drawn samples while $p(\tilde{Z}_\alpha|\tilde{Z}_{\alpha+1:K}, \tilde{S})$ is being estimated and they are defined as $l_\alpha = N + (K - \alpha)M + 1$ and $u_\alpha = N + (K - \alpha + 1)M$.

After replacing the terms in (21) with their estimates that are defined in (23) and (26), Chib's method estimates the marginal likelihood as follows:

$$
\begin{aligned}
\log p(X) =& \log p(\tilde{S}, \tilde{Z}_{1:K}, X) - \log p(\tilde{S}, \tilde{Z}_{1:K}|X) \qquad (27) \\
\approx& \log p(\tilde{S}, \tilde{Z}_{1:K}, X) - \log p(\tilde{Z}_1|\tilde{Z}_{2:K}, \tilde{S}) \\
& - \log \sum_{i=1}^{N} p(\tilde{S}|Z_{1:K}^{(i)}, X) \\
& - \sum_{\alpha=2}^{K} \log \sum_{m=l_\alpha}^{u_\alpha} p(\tilde{Z}_\alpha|Z_{1:\alpha-1}^{(m)}, \tilde{Z}_{\alpha+1:K}, \tilde{S}) \\
& + \log(K-1)MN. \qquad (28)
\end{aligned}
$$

3. EXPERIMENTS

In this section, we will illustrate the block and SADA sampler and Chib's method on three different tensor factorization models: a deconvolution model, a Parafac model, and an extended version of the NMF model.

3.1. Model I

Convolutive models emerge in various fields such as audio processing, image processing or seismic sciences. In order to illustrate the proposed sampling schemata, we give the deconvolution problem as an example and define it as a tensor

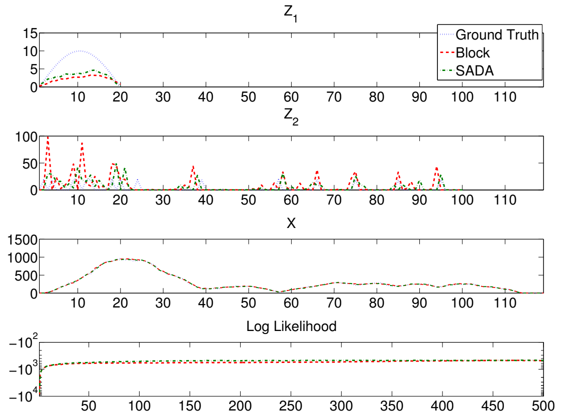

Fig. 2. Inference results for Model I. From top to bottom: the first and second figures show the real and the estimated values for the first factor (Z_1) and the second factor (Z_2), respectively. Third: Observed signal (X) and the model predictions ($\hat{X}$). Fourth: Log likelihood vs iteration plots of the samplers.

factorization problem as follows:

$$X(t) \approx \hat{X}(t) = \sum_r Z_1(r) Z_2(\overbrace{t-r}^{d})$$
$$= \sum_{r,d} Z_1(r) Z_2(d) Z_3(d,t,r) \qquad (29)$$

where Z_1 and Z_2 are the convolved signals and $\hat{X}$ is the output signal. In order to be able to define this model in the PLTF framework, we define a dummy index d and a dummy tensor $Z_3(d,t,r) = \delta(d-t+r)$, where $\delta(\cdot)$ is the Kronecker delta function.

In order to build the samplers, we first start by defining the index sets for this particular model: $V = \{t,r,d\}$, $V_0 = \{t\}$, $V_1 = \{r\}$, $V_2 = \{d\}$, and $V_3 = \{t,r,d\}$. After placing these index sets in Algorithm 1, we obtain the block Gibbs sampler as presented in Algorithm 3. Similarly, we can also obtain the SADA sampler for this model by placing these index sets in Algorithm 2. The inference results of block and SADA samplers on a toy problem are illustrated in Figure 2.

3.2. Model II

Parafac (also known as Candecomp or CP) is a popular model for decomposing three-way data and has been used in many fields including chemometrics, psychometrics, and signal processing [4]. The model is defined as:

$$\hat{X}(i,j,k) = \sum_m Z_1(i,m) Z_2(j,m) Z_3(k,m) \qquad (30)$$

where the three-way tensor X is decomposed into three matrices, Z_1, Z_2, and Z_3. In practice, the optimal number of components (indexed with m above) is not known beforehand and should be estimated.

Algorithm 3 Block Gibbs Sampler for Model I

```
Input: Observed data $X$, $A_\alpha, B_\alpha$ $\forall\alpha = 1,2$
Initialize factors: $Z_\alpha^{(0)} \sim \mathcal{G}(Z_\alpha; A_\alpha, B_\alpha)$ $\forall\alpha = 1,2$
for $i = 1 \ldots$ MAXITER do
    Compute the intensity and parameter tensors:
        $\Lambda(t,r,d) = Z_1'(r) Z_2'(d) Z_3(d,t,r)$
        $\hat{X}(t) = \sum_{r,d} \Lambda(t,r,d)$
    Sample Sources:
    for all $i$ do
        $S(t,:,:)^{(i)} \sim \mathcal{M}\left(\cdot; X(t), \frac{\Lambda(t,:,:)}{\hat{X}(t)}\right)$
    end for
    Sample $Z_1$:
    for all $r$ do
        $\Sigma_1 = A_1(r) + \sum_{t,d} S^{(i)}(t,r,d)$
        $\Phi_1 = B_1(r) + \sum_{t,d} Z_2'(d) Z_3(d,t,r)$
        $Z_1^{(i)}(r) \sim \mathcal{G}(Z_1(r); \Sigma_1, \Phi_1)$
    end for
    Sample $Z_2$:
    for all $d$ do
        $\Sigma_2 = A_2(d) + \sum_{t,r} S^{(i)}(t,r,d)$
        $\Phi_2 = B_2(d) + \sum_{t,r} Z_1'(r) Z_3(d,t,r)$
        $Z_2^{(i)}(d) \sim \mathcal{G}(Z_2(d); \Sigma_2, \Phi_2)$
    end for
end for
```

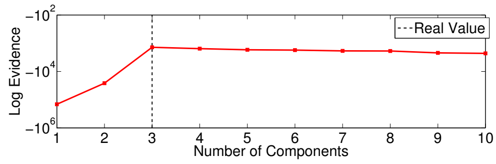

Fig. 3. Model selection results for Model II. The marginal Likelihood of the three-way observed data under the CP model ($\int dZ_{1:3}\ p(X|Z_{1:3})p(Z_{1:3})$) is estimated by using Chib's method.

In order to test our approach for model selection, we generated synthetic data where $|i| = 10$, $|j| = 5$, $|k| = 8$, and $|m| = 3$ and applied Chib's method to estimate marginal likelihood of the observed tensor under CP models with different number of components. Figure 3 shows the marginal likelihood estimates of the synthetic data for different number of components. It can be seen that the marginal likelihood estimate is at the highest when the correct number of components is selected.

3.3. Model III

Pioneering work on NMF for audio processing [10] has demonstrated that factorization based audio modeling can be very powerful. Many factorization based models have been

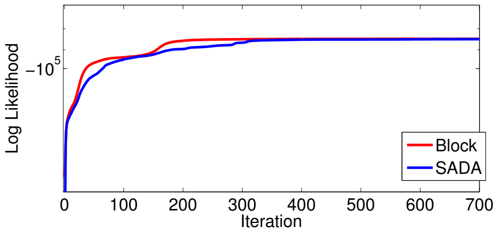

Fig. 4. Log-likelihood vs iteration plots for the block sampler and SADA sampler.

proposed for several audio/music processing applications such as source separation, transcription, and restoration.

We slightly modify the NMF model and define the following audio model:

$$\hat{X}(f,t) = \sum_i Z_1(f,i) \quad \overbrace{Z(i,k)}^{\sum_k Z_2(i,k)Z_3(k,t)}$$
$$= \sum_{i,k} Z_1(f,i)Z_2(i,k)Z_3(k,t) \qquad (31)$$

where $X(f,t)$ is the observed magnitude spectrum of the audio, f is the frequency index, and t is the time-frame index. When the musical signals are considered, i is called the note index. Here, Z_1 is called the 'spectral dictionary' since it encapsulates the spectral information for each musical note and Z is called the 'excitation' matrix. In this particular model we hierarchically decompose the excitation matrix as multiplication of a chord dictionary matrix Z_2 and its weights Z_3. Here the basis matrix Z_2 encapsulates the harmonic structure of the music and incorporates additional information to the factorization model. Note that, similar models to this model have been applied to different audio/music processing applications such as audio restoration [11] and musical source separation [12] and promising results have been reported.

We ran both the block sampler and the SADA sampler on a short polyphonic piano sound. We first estimated and fixed the spectral dictionary Z_1 than ran the inference algorithms. We used 4 spectral templates and 3 chord templates while having 1025 frequency bins and 86 time frames. Figure 4 shows the log-likelihoods of the algorithms. It can be observed that, both algorithms converge smoothly. Matlab implementations of these algorithms are available at `http://www.cmpe.boun.edu.tr/~umut/pltf_mcmc/`.

4. CONCLUSION

In this paper, we presented Markov Chain Monte Carlo procedures for making inference on the PLTF framework. We first provided a conventional sampling schema, and a more efficient sampling algorithm that makes use of space alternating data augmentation. We also described how the marginal likelihood of a tensor factorization model can be estimated by using Chib's method. The proposed methods were illustrated on three different tensor factorization models.

As a future direction and a next step of this work, we aim to extend our method in order to be able to make inference on tensor factorization models where multiple observed tensors $(X_1, \dots, X_K)$ can share a set of factors [11]. As another extension to the proposed approach, we also aim to apply more complex MCMC methods to the tensor factorization problem, such as the reversible jump algorithm.

5. REFERENCES

[1] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization.," *Nature*, vol. 401, pp. 788–791, 1999.

[2] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *WASPAA*, 2003, pp. 177–180.

[3] A. T. Cemgil, "Bayesian inference in non-negative matrix factorisation models," *Computational Intelligence and Neuroscience*, 2009.

[4] A. Cichoki, R. Zdunek, A.H. Phan, and S. Amari, *Non-negative Matrix and Tensor Factorization*, Wiley, 2009.

[5] Y. K. Yilmaz and A. T. Cemgil, "Algorithms for probabilistic latent tensor factorisation with beta divergence," *Signal Processing*, 2011.

[6] C. Fevotte, O. Cappe, and A. T. Cemgil, "Efficient markov chain monte carlo inference in composite models with space alternating data augmentation," in *SSP*, 2011.

[7] Jun S. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer, Jan. 2008.

[8] A. Doucet, S. Senecal, , and T. Matsui, "Space alternating data augmentation: Application to finite mixtures of gaussians and speaker recognition," in *ICASSP*, 2005.

[9] S. Chib, "Marginal likelihood from the gibbs output," *Journal of the Acoustical Society of America*, vol. 90, no. 432, pp. 1313–1321, 1995.

[10] P. Smaragdis, "Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs," in *ICA*, 2004, pp. 494–499.

[11] Y. K. Yilmaz, A. T. Cemgil, and U. Simsekli, "Generalised coupled tensor factorisation," in *NIPS*, 2011.

[12] U. Simsekli and A. T. Cemgil, "Score guided musical source separation using generalized coupled tensor factorization," in *EUSIPCO*, 2012.