Digital Signal Processing ••• (••••) •••–•••

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

# Non-negative tensor factorization models for Bayesian audio processing

Umut Şimşekli [a,*], Tuomas Virtanen [b], Ali Taylan Cemgil [a]

[a] Department of Computer Engineering, Boğaziçi University, 34342, Bebek, İstanbul, Turkey
[b] Department of Signal Processing, Tampere University of Technology, 33720 Tampere, Finland

ARTICLE INFO

*Article history:*
Available online xxxx

*Keywords:*
Nonnegative matrix and tensor factorization
Coupled factorization
Bayesian audio modeling
Bayesian inference

ABSTRACT

We provide an overview of matrix and tensor factorization methods from a Bayesian perspective, giving emphasis on both the inference methods and modeling techniques. Factorization based models and their many extensions such as tensor factorizations have proved useful in a broad range of applications, supporting a practical and computationally tractable framework for modeling. Especially in audio processing, tensor models help in a unified manner the use of prior knowledge about signals, the data generation processes as well as available data from different modalities. After a general review of tensor models, we describe the general statistical framework, give examples of several audio applications and describe modeling strategies for key problems such as deconvolution, source separation, and transcription.



## 1. Introduction

With the recent technological advances of sensor and communication technologies, the cost of data acquisition and storage is significantly reduced. Consequently, the last decade has witnessed the dramatic increase in the amount of data that can be easily collected. One important facet of data processing is extracting meaningful information from highly structured datasets that can be of interest for scientific, financial, or technological purposes.

The key to exploiting the potential of large datasets lies in developing computational techniques that can efficiently extract meaningful information. These computational methods must be scalable and tailored for the specifics of an application, but still be versatile enough to be useful in several scenarios. In this paper, we will focus on audio processing and review one particular class of such models, that provide a favorable balance between high modeling accuracy, ease of implementation and ease of management of required computational resources. This class of models, coined under the name of tensor factorization models along with their Bayesian interpretations, will be the focus of this tutorial paper. The mathematical setup may look somewhat abstract at a first sight, but the generic nature of the approach makes tensors suitable for a broad range of applications where complicated structured datasets need to be analyzed. In particular, we will show examples in the domain of audio processing where significant progress has been achieved using tensor methods. While the modeling and inference strategies can be applied in the broader context of general audio and other non-stationary time series analysis, the hierarchical Bayesian nature of the framework makes the approach particularly suitable for the analysis of acoustical signals.

In audio processing, an increasing number of applications are developed that can handle challenging acoustical conditions and highly variable sound sources. Here, one needs to exploit the inherent structure of acoustic signals to address some of the key problems such as denoising, restoration, interpolation, source separation, transcription, bandwidth extension, upmixing, coding, event recognition and classification. Not surprisingly, many different modeling techniques have been developed for those purposes. However, as is the case for computational modeling of all physical phenomena, we face here with a trade off: accuracy versus computational tractability – a physically realistic and accurate model may be too complex to meet the demands of a given application to be useful in practice.

Typically, there is a lot of *a-priori* knowledge available for acoustic signals. This includes knowledge of the physical or cognitive mechanisms by which sounds are generated or perceived, as well as the hierarchical nature by which they are organized in an acoustical scene. In more specific domains, such as music transcription or audio event recognition, more specialized assumptions about the hierarchical organization are needed. Yet, the

* Corresponding author.
*E-mail addresses:* umut.simsekli@boun.edu.tr (U. Şimşekli), tuomas.virtanen@tut.fi (T. Virtanen), taylan.cemgil@boun.edu.tr (A.T. Cemgil).

http://dx.doi.org/10.1016/j.dsp.2015.03.011
1051-2004/

resulting models often possess complex statistical structure and highly adaptive and powerful computational techniques are needed to perform inference.

Factorization-based modeling has been useful in addressing the modeling accuracy versus computational requirement trade off in various domains beyond audio signal processing [1], with prominent examples such as text processing [2], bioinformatics [3], computer vision [4], social media analysis [5], and network traffic analysis [6]. The aim in such modeling strategies is to decompose an observed matrix or tensor (multidimensional array) into semantically meaningful factors in order to obtain useful predictions. Meanwhile, the factors themselves also provide a useful feature representation about the specifics of the domain.

In this paper, we review tensor based statistical models and associated inference methods developed recently for audio and music processing and describe various extensions and applications of these models. In Section 2, we illustrate the ideas of factorization based modeling, and then in Section 3 we describe a probabilistic interpretation of these models. The probabilistic interpretation opens up the way for a full Bayesian treatment via Bayesian hierarchical modeling. This leads to a very natural means for unification, allowing the formulation of highly structured probabilistic models for audio data at the various levels of abstraction, as we will illustrate in Section 6. The paper concludes with remarks on future research directions.

## 2. Factorization-based data modeling

In this section, we will describe the basics of factorization based modeling, and describe extensions such as coupled tensor factorizations and nonnegative decompositions. This section will describe the main structure and the notation.

In many applications, data can be represented as a matrix, for example, the spectrogram of an audio signal (frequency vs time), a dataset of images (pixel coordinates vs instances), word frequencies among different documents (words vs documents), and the adjacency structure of a graph (nodes vs nodes) to name a few. Here the indices of the matrix correspond to the entities, and the matrix elements describe a relation between the two entities. Matrix Factorization (MF) models are one of the most widely used methods for analyzing the data that involve two entities [7–10]. The goal in these models is to calculate a factorization of the form:

$$X_1(i,j) \approx \hat{X}_1(i,j) = \sum_k Z_1(i,k) Z_2(k,j) \tag{1}$$

where $X_1$ is the given data matrix, $\hat{X}_1$ is an approximation to $X_1$, and $Z_1$, and $Z_2$ are factor matrices to be estimated. Even though we have a single observed matrix in this model, we use a subscript in $X_1$ since we will consider factorization models that involve more than one observed matrix or tensor, later in this section. Here, $X_1$ is expressed as the product of $Z_1$ and $Z_2$, where $Z_1$ is considered as the *dictionary* matrix and $Z_2$ contains the corresponding *weights*. From another perspective, $X_1$ is approximated as the sum of inner products of the columns of $Z_1$ and the rows of $Z_2$, as illustrated at the top of Fig. 2. Note that, if $Z_1$ would have been fixed, the problem would have been equivalent to basis regression where the weights (expansion coefficients) $Z_2$ are estimated [11]. In contrast, in matrix factorization the dictionary (the set of basis vectors) is estimated along with the coefficients. This modeling strategy has been shown to be successful in various fields including signal processing, finance, bioinformatics, and natural language processing [8].

Matrix factorization models are applicable when the observed data encapsulates the relation of two different entities (e.g., $i$ and $j$ in Eq. (1)). However, when the data involves multiple entities

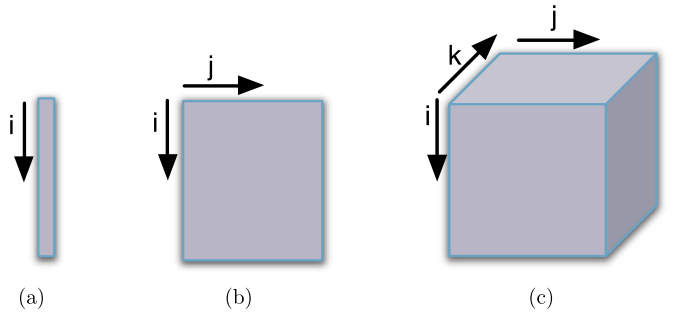


**Fig. 1.** Illustration of a) a vector $X(i)$: an array with one index, b) a matrix $X(i,j)$ an array with two indices, c) a tensor $X(i,j,k)$: an array with three or more indices. In this study, we refer vectors as tensors with one mode and matrices as tensors with two modes.

of interest, such as ternary or higher order relations it cannot be represented without loss of structure by using matrices. For example a multichannel sound library of several instances may be represented in the time-frequency domain conveniently as an object with several entities, say the power at each (frequency, time, channel, instance). One could in principle 'concatenate' each spectrogram across time and instances to obtain a big matrix, say (frequency × channel, time × instance) but this representation would obscure important structural information – compare simply with representing a matrix with a column vector. Hence one needs naturally multiway tables, the so-called *tensors*, where each element is denoted by $T(i,j,k,\ldots)$. Here, $T$ is the tensor and the indices $i,j,k,\ldots$ are the entities. The number of distinct entities dictates the mode of a tensor. Hence a vector and a matrix are tensors of mode one and two respectively. Tensors are illustrated in Fig. 1 and we will give a more precise and compact definition in Section 3.

For modeling multiway arrays with more than two entities the canonical polyadic decomposition [12,13] (also referred as, CP, PARAFAC, or CANDECOMP) is one of the most popular factorization models. The model, for three entities, is defined as follows:

$$X_2(i,m,r) \approx \hat{X}_2(i,m,r) = \sum_k Z_1(i,k) Z_3(m,k) Z_4(r,k) \tag{2}$$

where the observed tensor $X_2$ is decomposed as a product of three different matrices. Analogous to MF models, this model approximates $X_2$ as the sum of 'inner products' of the columns of $Z_1$, $Z_3$, and $Z_4$ as illustrated at the bottom of Fig. 2. This model has been shown to be useful in chemometrics [14], psychometrics [12], and signal processing [8].

Tucker model [15] is another important model for analyzing tensors with three modes, which is a generalization of the PARAFAC model. The model is defined as follows:

$$\begin{aligned} X_3(i,j,k) &\approx \hat{X}_3(i,j,k) \\ &= \sum_p \sum_q \sum_r Z_1(i,p) Z_2(j,q) Z_3(k,r) Z_4(p,q,r) \end{aligned} \tag{3}$$

where $X_3$ is expressed as the product of three matrices ($Z_{1:3}$) and a 'core tensor' ($Z_4$). When the core tensor $Z_4$ is chosen as super diagonal ($Z_4(p,q,r) \neq 0$ only if $p=q=r$), Tucker decomposition reduces to PARAFAC.

### 2.1. Coupled factorization models

In certain applications, information from different sources are available and need to be combined for obtaining more accurate predictions [16–20]. In musical audio processing, one example is having a large collection of annotated audio data and a collection of symbolic music scores as side information. Similarly, in product recommendation systems, a customer–product rating matrix

Please cite this article in press as: U. Şimşekli et al., Non-negative tensor factorization models for Bayesian audio processing, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.011

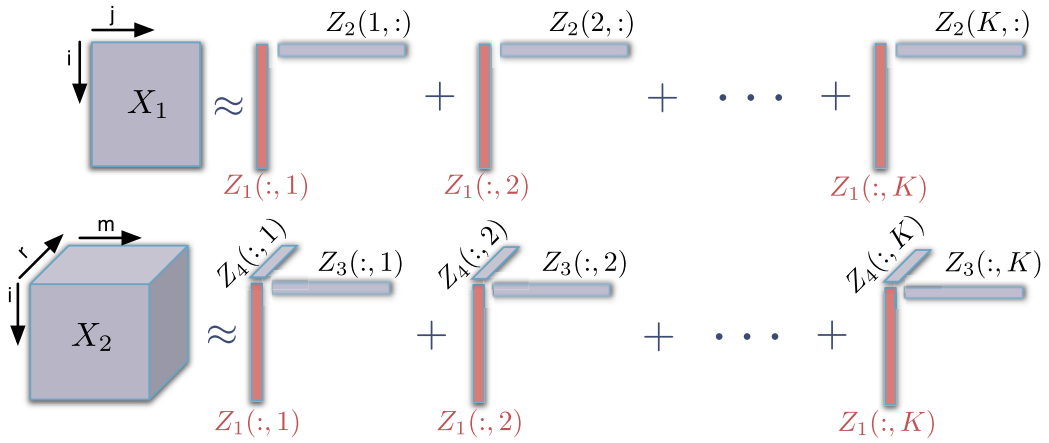


**Fig. 2.** Coupled MF-PARAFAC illustration. The observed matrix $X_1$ is approximated as the sum of inner products of the columns of $Z_1$ and the rows of $Z_2$. Similarly, $X_2$ is approximated as the sum of 'inner products' of the columns of $Z_1$, $Z_3$, and $Z_4$. The overall model is coupled since the matrix $Z_1$ is shared in both factorizations. Here $K$ denotes the size of the index $k$: $k \in \{1, \ldots, K\}$.

can be enhanced with connectivity information from a social network and demographic information from the customer. For these problems, a single factorization model would not be sufficient for exploiting all the information in the data and we need to develop more comprehensive modeling strategies in order to be able to combine different data sources in a single factorization model. Such models are called as *coupled tensor factorization* methods, where the aim is to simultaneously factorize multiple observed tensors that share a set of latent factors.

Let us consider an example coupled matrix–tensor factorization model where two observed tensors $X_1$ and $X_2$ are collectively decomposed as

$$X_1(i,j) \approx \hat{X}_1(i,j) = \sum_k Z_1(i,k) Z_2(k,j)$$
$$X_2(i,m,r) \approx \hat{X}_2(i,m,r) = \sum_k Z_1(i,k) Z_3(m,k) Z_4(r,k) \qquad (4)$$

where $X_1$ is decomposed by using an MF model and $X_2$ is decomposed by using a PARAFAC model. The factor $Z_1$ is the 'shared factor' in both decompositions, making the overall model coupled. Fig. 2 illustrates this model. In Section 6, we will illustrate the usefulness of various factorization models on audio processing applications.

### 2.2. Non-negative tensor factorizations

So far, we have described some of the most important matrix and tensor factorization models. However, even if two factorization models have the exact same topology (e.g., both models are MF models with the same number of parameters), depending on the constraints placed over the latent factors, the factorizations might have completely different interpretations. For instance, in the MF models, one option is not to restrict the factors by not placing any constraints over them. On the other hand, we can have orthogonality constraints on the factors, where the factorization would turn into the principal component analysis.

Even if we place highly restrictive constraints such as orthogonality, the estimated factors would be dense and their physical interpretations in applications would be quite limited as long as their elements are allowed to take any positive and negative values. In this study, we will consider *non-negative* factorization models [8], where we will restrict all the elements of the factors to be non-negative. Here, the non-negativity constraint implicitly imposes an *additive* structure on the model, where the contributions of the latent factors are always added since there will not be any cancellations due to negative values, as opposed to the aforementioned cases. Therefore, this strategy promotes sparsity on the factors since most of the entries in the factors would be close to zero in order the model to be able to fit the data, and more importantly the estimated factors will have physical interpretations that might be essential in many fields, such as audio processing. On the other hand, as we will describe in more detail in Sections 3 and 4.2, by modeling tensors with probabilistic tensor factorization models, we essentially decompose the parameters of a probabilistic model that are non-negative by definition (e.g., the intensity of a Poisson distribution or the mean of a gamma distribution) and are constructed as the sum of non-negative sources [9]. In this modeling strategy, the non-negativity constraint on the factors is rather a necessity than an option.

In audio processing, which is the main application focus of this study, we model the energies of signals in the time-frequency domain that are known as the magnitude or power spectra. For modeling these spectra, the non-negativity constraint turns out to be very natural, since realistic sounds can be viewed as being composed of purely additive components, as will be detailed in Section 5. For example, music signals consist of multiple instruments, and the signal of each instrument consists of multiple notes played by the instrument. Speech signals consist of basic units such as phonemes and words. Cancellation of sounds happens only intentionally and in very specific scenarios, for example in echo cancellation systems.

## 3. Probabilistic modeling of non-negative tensor factorizations

In applications, as we will demonstrate in Section 6, we often need to come up with custom model topologies, where either the observed tensors or the latent factors involve multiple entities and cannot be represented by using matrices without loss of important structural information. In order to be able to model the real world data sets that might consist of several tensors and require custom factorization models, we need to handle a broad variety of model topologies.

The Generalized Coupled Tensor Factorization (GCTF) framework [21] is a generalization of matrix and tensor factorization models to jointly factorize multiple tensors. The formal definition of the GCTF framework is as follows:

$$X_\nu(u_\nu) \approx \hat{X}_\nu(u_\nu) = \sum_{\bar{u}_\nu} \prod_\alpha Z_\alpha(v_\alpha)^{R^{\nu,\alpha}} \qquad (5)$$

where $\nu = 1, \ldots, N_x$ is the observed tensor index and $\alpha = 1, \ldots, N_z$ is the factor index. In this framework, the goal is to com-

Please cite this article in press as: U. Şimşekli et al., Non-negative tensor factorization models for Bayesian audio processing, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.011

**Table 1**
Illustration of different factorization models in the GCTF notation. Here $N_x$ is the number of observed tensors, $N_z$ is the number of latent factors, $V$ is the set of all indices in the model, $U_\nu$ are the set of indices of $X_\nu$, $V_\alpha$ are the set of indices of $Z_\alpha$, $R$ is the coupling matrix.

| | $N_x$ | $N_z$ | $V$ | $U_\nu$ | $V_\alpha$ | $R$ |
|---|---|---|---|---|---|---|
| MF (Eq. (1)) | 1 | 2 | $\{i,j,k\}$ | $\{i,j\}$ | $\{i,k\}$, $\{k,j\}$ | $[1,1]$ |
| PARAFAC (Eq. (2)) | 1 | 3 | $\{i,j,k,r\}$ | $\{i,j,k\}$ | $\{i,r\}$, $\{j,r\}$, $\{k,r\}$ | $[1,1,1]$ |
| TUCKER (Eq. (3)) | 1 | 4 | $\{i,j,k,p,q,r\}$ | $\{i,j,k\}$ | $\{i,p\}$, $\{j,q\}$, $\{k,r\}$,$\{p,q,r\}$ | $[1,1,1,1]$ |
| MF-PARAFAC (Eq. (4)) | 2 | 4 | $\{i,j,k,m,r\}$ | $\{i,j\}$, $\{i,m,r\}$ | $\{i,k\}$, $\{k,j\}$, $\{m,k\}$,$\{r,k\}$ | $\begin{bmatrix}1&1&0&0\\1&0&1&1\end{bmatrix}$ |

**Table 2**
Tweedie distributions with corresponding normalizing constants and divergence forms. The general form of the distribution is given in Eq. (7).

| $p$ | Divergence | Distribution | Normalizing constant | Divergence form |
|---|---|---|---|---|
| $2-\beta$ | $\beta$-divergence | Tweedie | $K(x,\phi,p)$ | $d_p(x\|\|\hat{x})$ |
| 0 | Euclidean | Gaussian | $(2\pi\phi)^{1/2}$ | $\frac{1}{2}(x-\hat{x})^2$ |
| 1 | Kullback–Leibler | Poisson | $\frac{e^{x/\phi}\Gamma(x/\phi+1)}{(x/\phi)^{x/\phi}}$ | $x\log\left(\frac{x}{\hat{x}}\right)-x+\hat{x}$ |
| 2 | Itakura–Saito | Gamma | $\Gamma(1/\phi)(e\phi)^{1/\phi}x$ | $\frac{x}{\hat{x}}-\log\left(\frac{x}{\hat{x}}\right)-1$ |
| 3 | – | Inverse Gaussian | $(2\pi x^3\phi)^{1/2}$ | $\frac{1}{2}\frac{(x-\hat{x})^2}{x\hat{x}^2}$ |

pute an approximate factorization of given observed tensors $X_\nu$ in terms of a product of individual factors $Z_\alpha$, some of which are possibly shared. Here, we define

$V$ as the set of all indices found in a model,
$U_\nu$ as the set of visible indices of the tensor $X_\nu$,
$V_\alpha$ as the set of indices in $Z_\alpha$,
$\bar{U}_\nu = V \setminus U_\nu$ as the set of invisible indices that are not present in $X_\nu$.

We use small letters as $v_\alpha$ to refer to a particular setting of indices in $V_\alpha$ (similarly $u_\nu \in U_\nu$ and $\bar{u}_\nu \in \bar{U}_\nu$). Furthermore, $R$ is a *coupling matrix* that is defined as follows: $R^{\nu,\alpha}=1$ if $X_\nu$ and $Z_\alpha$ are connected and $R^{\nu,\alpha}=0$ otherwise. In other words, the coupling matrix $R^{\nu,\alpha}$ specifies the factors $Z_\alpha$ that affect the observed tensor $X_\nu$. As the product $\prod_\alpha Z_\alpha(v_\alpha)$ is collapsed over a set of indices, the factorization is latent. Here, we consider non-negative observations and factors: $X_\nu(u_\nu)\geq 0$ and $Z_\alpha(v_\alpha)\geq 0$. This rather abstract notation is needed to express increasingly complicated tensor models without limiting one to a particular topology.

In order to illustrate the framework, we define the coupled PARAFAC-MF model of Eq. (4) in the GCTF notation as follows. The observed index sets are given as $U_1=\{i,j\}$ and $U_2=\{i,m,r\}$. The index sets of the factors are given as: $V_1=\{i,k\}$, $V_2=\{k,j\}$, $V_3=\{m,k\}$, $V_4=\{r,k\}$. The coupling matrix is given as $R=[1100;1011]$ and indicates that $\hat{X}_1$ is a function of $Z_1$ and $Z_2$ and $\hat{X}_2$ is a function of $Z_1$, $Z_3$, and $Z_4$. Table 1 lists all the factorization models described earlier in the paper as specific instances of the GCTF notation.

The GCTF framework assumes the following probabilistic model over each scalar element of the observed tensors [21]:

$$X_\nu(u_\nu)\sim \mathcal{TW}_{p_\nu}\left(X_\nu(u_\nu);\hat{X}_\nu(u_\nu),\phi_\nu\right),\qquad \nu=1,\ldots,N_x \tag{6}$$

where $\hat{X}_{1:N_x}$ are the model output tensors that are defined in Eq. (5) and $\mathcal{TW}$ denotes the so-called Tweedie distribution. Tweedie densities $\mathcal{TW}_p(x;\hat{x},\phi)$ can be written in the following moment form:

$$\mathbb{P}(x;\hat{x},\phi,p)=\frac{1}{K(x,\phi,p)}\exp\left(-\frac{1}{\phi}d_p(x||\hat{x})\right) \tag{7}$$

where $\hat{x}$ is the mean, $\phi$ is the dispersion, $p$ is the power parameter and $d_p(\cdot)$ denotes the $\beta$-divergence defined as follows:

$$d_p(x||\hat{x})=\frac{x^{2-p}}{(1-p)(2-p)}-\frac{x\hat{x}^{1-p}}{1-p}+\frac{\hat{x}^{2-p}}{2-p} \tag{8}$$

The Tweedie distribution is an important special case of exponential dispersion models, characterized by three parameters: the mean, dispersion, and power. This model is also known as the *power variance* model, since the variance of the data has the following form: $\mathrm{var}(x)=\phi\hat{x}^p$.

By taking appropriate limits, we can verify that the rather arbitrary looking divergence $d_p(\cdot)$ defined in Eq. (8) yields a lot more familiar divergence functions such as the Euclidean distance square, Kullback–Leibler (KL) divergence, and Itakura–Saito (IS) divergence for $p=0,1,2$, respectively. Using a suitable divergence is important in applications; for example the IS divergence is scale invariant, where the divergence between two points $x$ and $y$ does not change when both points are multiplied by the same constant, i.e. $d_2(x||y)=d_2(ax||ay)$ [9]. Physically, this translates to our intuitive notion that the discrepancy between two sound signals should not change by just turning up their volume.

From the probabilistic perspective, different choices of $p$ yield important distributions such as Gaussian ($p=0$), Poisson ($p=1$), compound Poisson ($1<p<2$), gamma ($p=2$) and inverse Gaussian ($p=3$) distributions. Due to a rather technical condition, no Tweedie model exists for the interval $0<p<1$, but for all other values of $p$, one obtains the very rich family of Tweedie stable distributions [22]. Table 2 illustrates the Tweedie distribution for $p\in\{0,1,2,3\}$.

An important property of the Tweedie models is that the normalizing constant $K(\cdot)$ does not depend on the mean parameter $\hat{x}$. Therefore, provided that $p$ and $\phi$ are given, it is easy to see that solving a maximum likelihood problem for $\hat{x}$ is equivalent to minimization of the $\beta$-divergence. For instance, for the Gaussian case (see Table 2), the divergence function is the squared Euclidean distance and the dispersion is simply the variance. For all possible $p$ values, we have a similar form; the Tweedie models generalize the established theory of least squares linear regression to more general noise models.

For regularization and incorporating prior knowledge, we place prior distributions over the latent factors. Depending on the application, we might consider different prior distributions on the latent factors. For example, we can choose an exponential prior over the latent factors that can be used for all the cases of the power parameter $p$, shown as follows:

Please cite this article in press as: U. Şimşekli et al., Non-negative tensor factorization models for Bayesian audio processing, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.011

$$Z_\alpha(v_\alpha) \sim \mathcal{E}(Z_\alpha(v_\alpha); A_\alpha(v_\alpha))$$

where $\mathcal{E}$ denotes the exponential distribution. We can also choose more sophisticated priors for individual cases of $p$. For instance, in the case of Poisson observations ($p = 1$), we can choose a gamma prior model

$$Z_\alpha(v_\alpha) \sim \mathcal{G}(Z_\alpha(v_\alpha); A_\alpha(v_\alpha), B_\alpha(v_\alpha))$$

where $\mathcal{G}$ denotes the gamma distribution. The definitions of the distributions that are mentioned in this paper are given in Appendix A.

## 4. Inference

Once we observe the tensors $X_{1:N_x}$, we would like to make inference in the probabilistic model defined in Eq. (6). Depending on the application, we might be interested in obtaining

- Point estimates, such as maximum likelihood (ML) or maximum a-posteriori (MAP):
  - ML: $Z^\star_{1:N_z} = \underset{Z_{1:N_z}}{\operatorname{argmax}} \log\left[\mathbb{P}(X_{1:N_x}|Z_{1:N_z})\right]$
  - MAP: $Z^\star_{1:N_z} = \underset{Z_{1:N_z}}{\operatorname{argmax}} \log\left[\mathbb{P}(X_{1:N_x}|Z_{1:N_z})\mathbb{P}(Z_{1:N_z})\right]$
- The full posterior distribution: $\mathbb{P}(Z_{1:N_z}|X_{1:N_x})$

where $X_{1:N_x}$ and $Z_{1:N_z}$ denote all the observed tensors and all the latent factors, respectively.

In this section, we will explain inference algorithms for each of these problems. Firstly, we will explain a gradient-based inference algorithm for maximum likelihood or a-posteriori estimation. Then we will explain a Markov Chain Monte Carlo procedure, namely the Gibbs sampler, for sampling the posterior distribution over the latent factors.

### 4.1. Maximum likelihood and maximum a-posteriori estimation

Given the dispersion $\phi_v$ and power parameters $p_v$, ML estimation of the factors $Z_{1:N_z}$ reduces to the problem of minimizing the $\beta$-divergence between the observations and the product of the latent factors, given as follows:

$$\begin{aligned} \text{minimize} \quad & \sum_v \sum_{u_v} \frac{1}{\phi_v} d_{p_v}\Big(X_v(u_v) || \sum_{\bar{u}_v} \prod_\alpha Z_\alpha(v_\alpha)^{R^{v,\alpha}}\Big) \\ \text{subject to} \quad & Z_\alpha(v_\alpha) \geq 0, \quad \forall \alpha \in [N_z], v_\alpha \in V_\alpha \end{aligned} \tag{9}$$

where, the power parameter $p_v$ determines the cost function to be used for $X_v$ and the dispersion parameter $\phi_v$ determines the relative weight of the approximation error to $X_v$.

ML estimation of the latent factors $Z_\alpha$ can be achieved via iterative methods, by fixing all factors $Z_{\alpha'}$ for $\alpha' \neq \alpha$ but one $Z_\alpha$ and updating in an alternating fashion. For non-negative data and factors, we present multiplicative update rules that have the following form [21]:

$$Z_\alpha \leftarrow Z_\alpha \circ \frac{\sum_v R^{v,\alpha} \phi_v^{-1} \Delta_{\alpha,v}(\hat{X}_v^{-p_v} \circ X_v)}{\sum_v R^{v,\alpha} \phi_v^{-1} \Delta_{\alpha,v}(\hat{X}_v^{1-p_v})}, \tag{10}$$

where $\circ$ is the element-wise product and the division operator is also element-wise. The key quantity in the above update equation is the $\Delta_{\alpha,v}$ function that is defined as follows:

$$\Delta_{\alpha,v}(A) = \left[ \sum_{u_v \cap \bar{v}_\alpha} A(u_v) \sum_{\bar{u}_v \cap \bar{v}_\alpha} \prod_{\alpha' \neq \alpha} Z_{\alpha'}(v_{\alpha'})^{R^{v,\alpha'}} \right] \tag{11}$$

For updating $Z_\alpha$, we need to compute this function twice for arguments $A = \hat{X}_v^{-p_v} \circ X_v$ and $A = \hat{X}_v^{1-p_v}$. Even though this function seems complicated, $\Delta_{\alpha,v}(\hat{X}_v^{1-p_v}) - \Delta_{\alpha,v}(\hat{X}_v^{-p_v} \circ X_v)$ is nothing but the gradient of $d_\beta(X_v||\hat{X}_v)$ with respect to $Z_\alpha$, as detailed in Appendix B. Intuitively, the terms $(\hat{X}_v^{-p_v} \circ X_v)$ and $\hat{X}_v^{1-p_v}$ come from the derivative of the $\beta$-divergence and the term $\sum_{\bar{u}_v \cap \bar{v}_\alpha} \prod_{\alpha' \neq \alpha} Z_{\alpha'}(v_{\alpha'})^{R^{v,\alpha'}}$ is just the derivative of $\hat{X}_v$ with respect to $Z_\alpha$, where the product is over all the factors but $Z_\alpha$ (i.e., $\alpha' \neq \alpha$). The monotonicity of these update rules for $N_x = 1$ is analyzed in [23].

For MAP inference, the objective given in Eq. (9) will have an additional regularization term that involves $Z_\alpha$. The resulting algorithm for MAP inference turns out to be similar to the ML schema. For exponential priors over the factors, the update equation becomes a simple modification:

$$Z_\alpha \leftarrow Z_\alpha \circ \frac{\sum_v R^{v,\alpha} \phi_v^{-1} \Delta_{\alpha,v}(\hat{X}_v^{-p_v} \circ X_v)}{A_\alpha + \sum_v R^{v,\alpha} \phi_v^{-1} \Delta_{\alpha,v}(\hat{X}_v^{1-p_v})}. \tag{12}$$

For other conjugate priors, the update rules have similar forms [24].

The benefit of the multiplicative updates is that they can be applied on any tensor factorization model and are rather simple to implement. The downside of the multiplicative updates is that they typically require a large number of iterations to convergence. Alternative optimization methods based on second-order optimization [25–27] and active-set methods [28,27] can also be applied for specific types of models.

### 4.2. Full Bayesian inference via the Gibbs sampler

Maximum likelihood and a-posteriori estimation methods provide us useful and practical tools that can be used in various applications. However, in certain cases, they are prone to over-fitting since they fall short at capturing uncertainties that arise in the inference process. Instead of aiming to obtain single point estimates of the latent variables, full Bayesian inference aims to characterize the full posterior distribution over the latent variables. Apart from being able to handle the uncertainties and therefore being robust to over-fitting, full Bayesian inference has many advantages over the point estimation methods in various tasks such as the model selection problem. In this section, we will develop a Monte Carlo method for characterizing the full posterior over the latent factors.

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

$$\langle \varphi(x) \rangle_{\pi(x)} = \int \varphi(x)\pi(x)dx \approx \frac{1}{N} \sum_{i=1}^{N} \varphi(x^{(i)}) \tag{13}$$

where $\langle \varphi(x) \rangle_{\pi(x)}$ denotes the expectation of the function $\varphi(x)$ under the distribution $\pi(x)$ and $x^{(i)}$ are independent samples drawn from the target distribution $\pi(x)$, that will be the posterior distribution in our case. Under mild conditions on the test function $\varphi$, this estimate converges to the true expectation as $N$ goes to infinity. The challenge here is obtaining independent samples from a nonstandard target density $\pi$.

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:

$$x^{(i+1)} \sim \mathcal{T}(x|x^{(i)}). \tag{14}$$

The transition kernel $\mathcal{T}$ does not need to be formed explicitly in practice; we only need a procedure that samples a new configuration, given the previous one. Perhaps surprisingly, even though

Please cite this article in press as: U. Şimşekli et al., Non-negative tensor factorization models for Bayesian audio processing, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.011

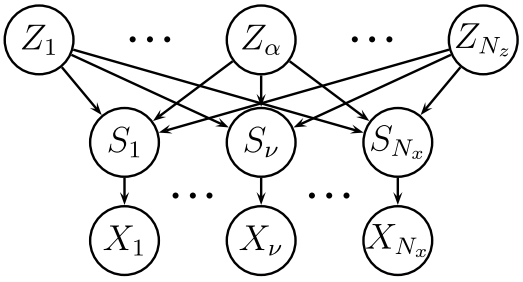


**Fig. 3.** Graphical representation of the GCTF framework. The nodes represent the random variables and the arrows represent the conditional independence structure.

these subsequent samples are correlated, Eq. (13) remains still valid, and estimated expectations converge to their true values when number of samples $i$ goes to infinity, provided that $\mathcal{T}$ satisfies certain ergodicity conditions. In order to design a transition kernel $\mathcal{T}$ such that the desired distribution is the stationary distribution, that is, $\pi(x) = \int \mathcal{T}(x|x')\pi(x')dx'$, various strategies can be applied; the most popular one being the Metropolis–Hastings (MH) algorithm [29]. 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.

Deriving a single Gibbs sampler for all the cases of the power parameter $p$ is not straightforward, therefore one needs to focus on individual cases of the Tweedie family. For the Poisson model ($p = 1$), we define the following augmented generative model:

$$\begin{aligned} Z_\alpha(v_\alpha) &\sim \mathcal{G}(Z_\alpha(v_\alpha); A_\alpha(v_\alpha), B_\alpha(v_\alpha)) && \text{factor priors} \\ \Lambda_\nu(v) &= \prod_\alpha Z_\alpha(v_\alpha)^{R_{v,\alpha}} && \text{intensities} \\ S_\nu(v)|Z_{1:N_z} &\sim \mathcal{PO}(S_\nu(v); \Lambda(v)) && \text{sources} \\ X_\nu(u_\nu) &= \sum_{\bar{u}_\nu} S_\nu(v) && \text{outputs} \end{aligned}$$

In the model defined in Eq. (6), the observed tensors $X_\nu$ directly depend on the latent factors $Z_\alpha$. Here, we form a so called composite model where we augment the model in Eq. (6) by defining the intensity tensors $\Lambda_\nu$ and the source tensors $S_\nu$ as intermediate layers, where in this model, the observed tensors are deterministic functions of the sources. Fig. 3 illustrates this model. Note that, the Tweedie models with $p = 0$ and $p = 2$, can also be represented as composite models [30]; however, the other cases have not been explored in the literature.

The Gibbs sampler for the GCTF model with Poisson observations can be formed by iteratively drawing samples from the full conditional distributions as follows:

$$S_\nu^{(i+1)} \sim p(S|Z_{1:N_z}^{(i)}, X_\nu, \Theta) \qquad \nu = 1 \dots N_x \tag{15}$$

$$Z_\alpha^{(i+1)} \sim p(Z_\alpha|S_{1:N_x}^{(i)}, Z'_{\neg\alpha}, X_{1:N_x}, \Theta) \qquad \alpha = 1 \dots N_z \tag{16}$$

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

$$p(S_\nu|\cdot) = \prod_{u_\nu} \mathcal{M}\left(S_\nu(u_\nu, \bar{U}_\nu); \frac{\Lambda_\nu(u_\nu, \bar{U}_\nu)}{\hat{X}_\nu(u_\nu)}\right)$$

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

where

$$\Sigma_\alpha(v_\alpha) = A_\alpha(v_\alpha) + \left[\sum_\nu R_{\nu,\alpha}\left(\sum_{\bar{v}_\alpha} S_\nu(v)\right)\right]$$

$$\Phi_\alpha(v_\alpha) = B_\alpha(v_\alpha) + \left[\sum_\nu \left(\sum_{\bar{v}_\alpha} \prod_{\alpha' \neq \alpha} Z_{\alpha'}(v_{\alpha'})^{R_{v,\alpha'}}\right)\right]$$

Here, $\mathcal{M}$ denotes the multinomial distribution. Verbally, given a particular instance of observed indices $u_\nu$, the full conditional of $S_\nu$ is a multinomial distribution over all the latent indices $\bar{U}_\nu$. Note that, the algorithms presented in [10] and [31] are special cases of the presented sampling algorithm.

The marginal likelihood of the observed data under a tensor factorization model $p(X_{1:N_x})$ is often necessary for certain problems such as model selection. By using the samples generated by the Gibbs sampler, the marginal likelihood $P(X_{1:N_x})$ can be estimated by using Chib's method [32]. Besides, more efficient samplers can be formed by using space alternating data augmentation [33,31].

## 5. Audio processing applications

In addition to the rich theoretical structure of non-negative tensor factorizations, they have a plenty of practical applications. Because of their ability to efficiently model the high-level structure of audio signals, tensor factorizations have been used to solve grand signal processing challenges such as source separation and robust pattern recognition. They have also provided completely new solutions to problems that are more specific to audio signals, such as audio bandwidth extension, dereverberation, and audio upmixing. We will first describe the audio representations that are used in tensor factorizations, and then describe different audio processing applications where tensor factorizations have successfully been applied.

### 5.1. Time-frequency representation

An audio signal represents the air pressure as the function of time and contains both positive and negative values. In the simplest scenario the signal is recorded with only one microphone, and the signal is therefore one dimensional. In order to enable modeling these kinds of audio signals with non-negative tensor factorization, we typically represent them using a *spectrogram* that characterizes the intensity of sound as the function of time and frequency. Fig. 4 represents an example audio signal and its spectrogram.

Unlike raw audio signals that can show huge variations, spectrogram representations of sounds possess typically easily perceivable patterns. For example, in the example figure, one can see that the notes are harmonic (there are large amplitudes at regular frequency intervals), and the spectrum of each note is rather static over time. This allows modeling them efficiently with non-negative tensor factorizations. Sound types with more diverse characteristics can also be modeled with tensor factorizations, provided that an appropriate model that takes into account the structure of the sound is used.

A standard procedure to calculate the spectrogram consists of the following steps:

1. Segment the signal into fixed-length frames. Typical frame length range between 10 ms and 100 ms. Adjacent frames are typically overlapping 50% or 25%.
2. Window each frame with a window function such as the Hamming window to smooth discontinuities at frame boundaries.
3. Calculate the spectrum within each frame by applying discrete Fourier transform.
4. Calculate the magnitude spectrum by taking the absolute values of each entry of the spectrum. The spectrum can also be decimated e.g. by integrating magnitudes within Mel bands.

Please cite this article in press as: U. Şimşekli et al., Non-negative tensor factorization models for Bayesian audio processing, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.011

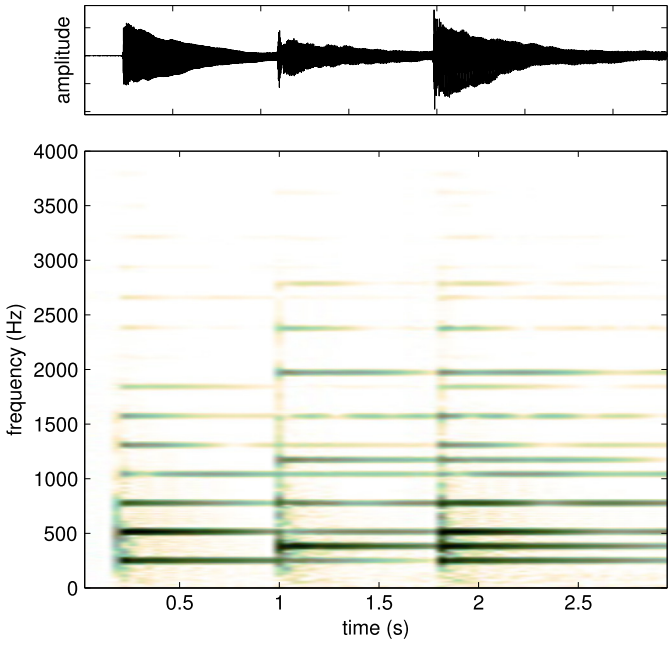


**Fig. 4.** An example audio signal (top panel) consisting of note sequence C4, G4, C4 + G4 played by piano, and its spectrogram (bottom panel).

The resulting data matrix $X(i, j)$ is indexed by frequency index $i$ and frame index $j$. In the case of multichannel audio where multiple microphones are available, the above procedure would be applied separately to each channel, so result in a 3D data tensor indexed by time, frequency, and channel.

In some applications such as signal classification [34] or polyphonic music transcription [35] it suffices to apply tensor factorizations on the data matrices, without the need to reconstruct time-domain audio signals. However, in many applications such as signal denoising and source separation [36,37], the target output is an audio signal, and there is need to reconstruct an audio signal based on the factor matrices. Steps 1–3 above are invertible, but Step 4 discards the phase information, and is therefore not invertible. There exist methods for reconstructing signals from magnitude spectrograms [38,39], but a simple and efficient approach uses simply the phases of the complex spectrogram obtained at Step 3, and assigns them for the output data matrices.

### 5.2. Source separation and signal denoising

Many real-world audio signals are *mixtures* consisting of multiple sources. For example, in music recordings there are typically multiple instruments playing, and speech signals captured by mobile phones contain some interfering sources from the background. Many signal processing algorithms (classification, coding, etc.) assume a single source, and therefore there is a large need for *source separation* algorithms that extract the signal produced by an individual source from a mixture.

In the context of tensor factorizations, mixture data matrix $X(i, j)$ is modeled as the sum of sources $S(i, j, n)$ (see Section 4.2) as

$$X(i, j) = \sum_{n=1}^{N} S(i, j, n), \tag{17}$$

where $n$ is the source index and $N$ is the number of sources. For example in the case of speech denoising, we would have $N = 2$, and source $n = 1$ would correspond to speech, and $n = 2$ would correspond to noise [40].

Each source is modeled with tensor factorization. For example, the NMF factorization for source $n$ is given as

$$S(i, j, n) \approx \hat{S}(i, j, n) = \sum_{k} Z_1(i, k, n) Z_2(k, j, n). \tag{18}$$

By combining the above equations, we can conveniently write the model for the mixture data matrix also as a tensor factorization as

$$X(i, j) \approx \hat{X}(i, j) = \sum_{n=1}^{N} \sum_{k} Z_1(i, k, n) Z_2(k, j, n). \tag{19}$$

Provided that the data matrices of individual sources become correctly estimated, each source can be reconstructed separately according to Eq. (18).

Depending how much a priori knowledge about the sources is utilized, a source separation problem can be termed as being unsupervised, supervised, or semisupervised:

- *Unsupervised:* all the factor matrices are estimated using the mixture signal only [37], and no training data is used to estimate them.
- *Supervised:* isolated training data of each source exists and has been used to obtain factor matrices $Z_1(i, k, n)$ representing the source spectra at a training stage, which are then kept fixed [41,40].
- *Semi-supervised:* isolated training data exists at least for one of the sources which is used to estimate entries of $Z_1(i, k, n)$ for those $n$ for which training data is available. However, isolated training data does not exist for all the sources and $Z_1(i, k, n)$ for the rest of the sources needs to be estimated from the mixture data matrix [42,43]. Furthermore, in some cases there might be side information available for some sources, such as a collection of symbolic music data that is related to a certain source. Incorporating such side information to the estimation process via coupled factorization models can further improve the separation accuracy [44].

### 5.3. Dereverberation

In natural environments the signal recorded by a microphone is a convolution of a source signal and the impulse response from the source to the microphone. Large amounts of reverberation make the signal less intelligible and difficult to process and analyze with algorithms that assume undistorted source signals. Therefore there is need for algorithms that either *dereverberate* the signal, or those that are able to process and analyze reverberant signals. Tensor factorizations can be used for both purposes.

The data matrix $X(i, j)$ of reverberant signal can be modeled as

$$\begin{aligned} X(i, j) \approx \hat{X}(i, j) &= \sum_{t} U(i, t) H(i, \overbrace{j - t}^{d}) \\ &= \sum_{t} \sum_{d} U(i, t) H(i, d) \delta(d - j + t) \\ &= \sum_{t,d} U(i, t) H(i, d) Z_3(d, i, t) \end{aligned} \tag{20}$$

where $\delta(x) = 1$ if $x = 0$ and $\delta(x) = 0$ otherwise, $U(i, t)$ is the data matrix of unreverberant, dry signal, and $H(i, d)$ is the convolution filter in the time-frequency domain [45,46]. Tensor $Z_3(d, i, t) = \delta(d - j + t)$ is fixed.

Note that above we assume that the unreverberant signal matrix $U(i, t)$ and the convolution filter response matrix $H(i, d)$ are

Please cite this article in press as: U. Şimşekli et al., Non-negative tensor factorization models for Bayesian audio processing, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.011

non-negative. Modeling arbitrary time-domain signals and convolutions would require using complex-valued signal and response matrices, and therefore the above formulation provides only an approximation of realistic reverberations. On the other hand, the non-negativity is an efficient regularization, that overcomes some problems present in standard deconvolution problems.

The above formulation is a specific instance of non-negative tensor factorizations, which allows estimating $U(i,j)$ and $H(i,d)$ directly using techniques described in Section 4. In practice, better reverberation results are obtained by modeling $U(i,j)$ with another tensor factorization (for example NMF, [45]), or by regularizing it by using a sparse prior for its entries [46].

### 5.4. Robust classification

Pattern recognition of audio signals is the core of many applications, for example automatic speech recognition, automatic music transcription, and multimedia information retrieval. As was explained in Section 5.2, many real-world audio signals consist of multiple sources. This makes the recognition of an individual source within a mixture challenging, since acoustic features extracted from the mixture do not represent the target source only, but are distorted by other sources.

Source separation techniques described in Section 5.2 could naturally be used as a pre-processing step to separate the target signal from a mixture. However, the artifacts produced by source separation may have a negative effect on the recognition accuracy, and classification approaches that directly utilize tensor factorization have been found more efficient in some scenarios [34].

Provided that there is isolated training material from each of the target sources classes, we can model the mixture signal as the sum of source and class specific tensor models as in the source separation approach in Eq. (19). For frame $j$, the likelihood for each class $n$ can then be calculated as $\sum_k Z_2(k,j,n)$ [34], without the need to actually reconstruct source signals. Non-linear mappings from tensor $Z_2(k,j,n)$ to likelihoods can also be used [47].

The coupled tensor factorization framework allows also learning mappings between acoustic data and class likelihoods from mixture signals [48,49]. At a training stage, this approach factors acoustic data matrix $X_1(i,j)$ and reference class likelihood matrix $X_2(c,j)$ jointly as

$$X_1(i,j) \approx \sum_k Z_1(i,k) Z_2(k,j) \tag{21}$$

$$X_2(c,j) \approx \sum_k Z_3(c,k) Z_2(k,j). \tag{22}$$

At the actual usage stage, the learned $Z_1(i,k)$ is kept fixed, and $Z_2(k,j)$ is estimated to model the observed acoustic data matrix according to Eq. (21). After that $Z_2(k,j)$ is projected to give an estimate of the class likelihood matrix according to Eq. (22) with fixed $Z_3(i,k)$.

### 5.5. Restoration and missing data imputation

In some scenarios only a part of an audio signal has been reliably observed, which affects its perceptual quality as well as suitability for computational analysis. For example, when audio streamed from the Internet, a segment of the signal may be unavailable because of packet losses. Similarly, physical damage inflicted to a vinyl recording may introduce 'clicks' where a short audio segment is lost. Part of audio can also be considered lost, if a recording includes interfering sources which dominate a time-frequency segment of a signal. Redundancy in the audio signal can be used to get an estimate of the missing data.

Within the tensor factorization framework, a missing data scenario can be formulated in terms of the observed data matrix $X(i,j)$ (that may contain missing segments whose values are assumed to be zero), and a fixed binary mask $M(i,j)$, where entries with values 1 correspond to reliable, observed data points of $X$, and values 0 indicate lost data points. A generic model for $X$ is given as

$$M(i,j)X(i,j) \approx M(i,j)\hat{X}(i,j), \tag{23}$$

where $\hat{X}(i,j)$ is a tensor model. From the probabilistic point of view, when a data sample is missing, i.e., $M(i,j)=0$, this is equivalent that the noise variance on $X(i,j)$ is infinite. Therefore, this particular sample does not contribute to the likelihood. When the latent factors are being estimated, only the observed elements are taken into account; in other words we minimize the following discrepancy:

$$\sum_{\substack{i,j,\\ M(i,j)=1}} d_\beta(X(i,j)||\hat{X}(i,j)). \tag{24}$$

Here, the binary mask $M$ enables the estimation process to use only the reliable, observed data points for which $M(i,j)=1$. An estimate for missing data points for which $M(i,j)=0$ is simply given by $\hat{X}(i,j)$. Since typical audio data matrices are low rank and can be approximated with tensor factorizations, redundancy in the data allows recovering the missing entries [50,51].

It should be noted that when complete frames of data are missing, i.e. for some $j$ all the values $X(i,j), i=1\dots I$ are missing, basic NMF cannot recover the data, and one needs to use a tensor factorization that explicitly models the relation between frames. For example convolutive techniques [52,36] that will be discussed in Section 6.1 can be used for this purpose. For music signals, the restoration quality can be further improved by introducing symbolic music data as side information [21,53]. It should also be noted that the above techniques can only recover the magnitudes of the missing data, and there is need to use other methods to estimate missing phases.

### 5.6. Coding and upmixing

Even though perceptual audio codecs are widely adopted and can provide significant compression gains while preserving high audio quality, there is always need for higher compression gains. Existing audio coding methods are mainly based on perceptual coding, that rely on coding perceptually less relevant parts of a signal coarsely. They do not utilize long-term redundancies, which are present in most of the audio signals to be coded. Since tensor factorizations effectively model the redundancy in the data, they have potential in audio coding as well.

Tensor factorizations can be used for audio coding simply by representing the audio data matrix $X(i,j)$ to be encoded with a tensor model, such as the one in Eq. (1). The parameters of the tensor matrices are then quantized and transmitted [54]. Criteria that maximize the perceptual quality of the factorization can be used to estimate the factor matrices [54].

A major drawback of the discussed tensor models for audio coding is their capability to model magnitude spectrogram data matrices only. For signal reconstruction, the phases need to be coded separately. Since the phases are much more stochastic, they cannot be compressed with tensor factorizations, and coding them requires a significant numbers of bits [54].

There exist also ways that avoid coding the phases. For example, we can do *audio upmixing* to reconstruct multichannel audio signals represented using a tensor model as follows. At the encoding stage, a multichannel audio recording is modeled with

Please cite this article in press as: U. Şimşekli et al., Non-negative tensor factorization models for Bayesian audio processing, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.011

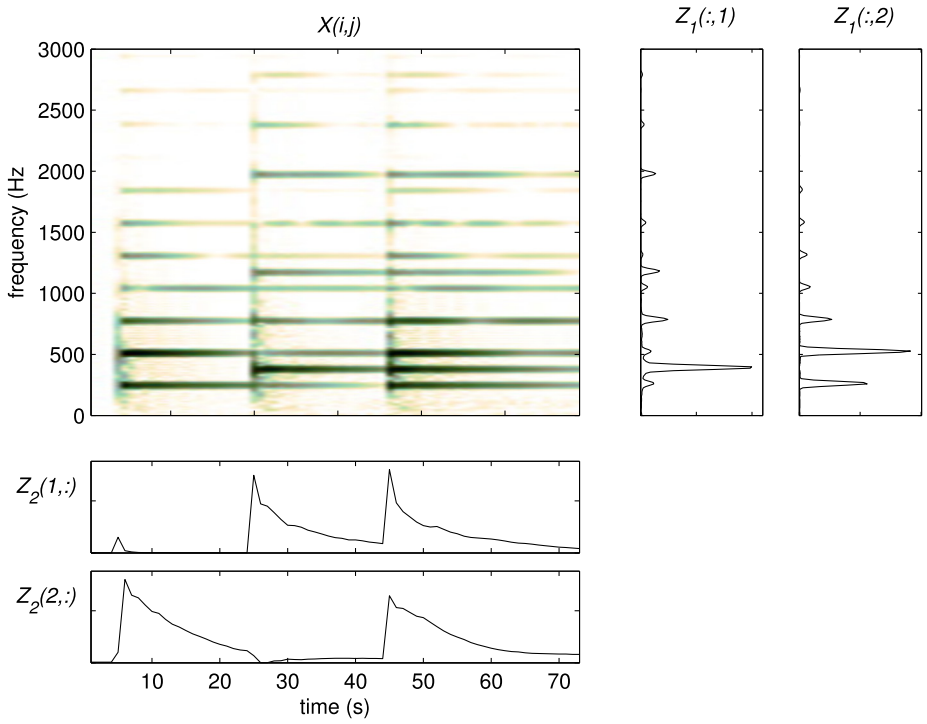


**Fig. 5.** Example spectrogram factored into a product of two factor matrices with the basic NMF model.

3D tensor factorization to learn the prominent repeating spectral structures and their amplitudes within each channel. Then, the multichannel audio is downmixed to one or two channels, and the resulting signal is coded using a conventional audio codec. Also the estimated data tensor matrices are coded, and transmitted together with the coded downmixed signal. At the decoder, the downmixed signal can be upmixed to multiple channels using the tensor factorization parameters. Even with fairly simple tensor model quantizations the approach produces performance that is comparable to state-of-the-art multichannel audio codecs [55].

A slightly similar idea can be used in *informed source separation*, where an encoder estimates the parameters of individual sound sources with a tensor model using data from isolated recordings, before they are mixed to form the target mixture signal. The tensor model parameters and the mixture signal are then coded and transmitted. The decoder can extract individual sources from the mixture using the tensor model parameters [56]. Estimating the tensor model parameters from isolated material allows better separation quality in comparison to approaches where the separation is done blindly, i.e, not using information about the signals before mixing.

## 6. Audio processing models

The general tensor factorization framework presented in Section 3 allows a wide variety of different models. The goal of this section is to illustrate the general framework by giving examples of some commonly used models and their interpretations, and also show how the models can be made increasingly more complex but still realistic for audio processing.

Let us start with the simple and most commonly used model, NMF, presented already in Eq. (1), which is given as

$$\hat{X}(i,j) = \sum_k Z_1(i,k) Z_2(k,j). \tag{25}$$

The interpretation of this model is that the spectrogram is modeled as a sum of components indexed by $k$, each of which has a static spectrum $Z_1(i,k)$ and time-varying gain $Z_2(k,j)$. Many realistic audio sources can be modeled with this model: for example, individual tones produced by musical instruments have often rather stationary spectrum, which amplitude just changes over time [37]. Speech is composed of basic units such as phonemes, and can also be modeled as a sum of components [34]. An example factorization of a simple music signal spectrogram with this model is illustrated in Fig. 5.

### 6.1. Convolutive matrix factorization

A shortcoming of the basic NMF model is that each frame is modeled independently from each other, and rearranging the frames does not affect the factorization. However, natural sounds have very strong temporal dynamics, meaning that adjacent frames are dependent on each other, and it is often advantageous to explicitly model these dependencies. There exist several dynamic extensions of NMF, including smooth NMF [37,9], non-negative hidden Markov model [57], non-negative dynamical systems [58–60], factorial scaled hidden Markov model [61], and high-resolution NMF [62,63].

The most commonly used extension is the *non-negative matrix factor deconvolution* [64], given as

$$\hat{X}(i,j) = \sum_k \sum_d Z_1(i,k,d) Z_2(k, j-d). \tag{26}$$

In this model, instead of static component spectra $Z_1(i,k)$ that were used in Eq. (25), we use spectrogram patches $Z_1(i,k,d)$ having also a time dimension $d = 0,\ldots,D$. Each patch represents a short spectrogram object. The factor $Z_2$ defines the amplitude for these spectral patches for each occurrence of each patch. The model is illustrated in Fig. 6.

In order to make the convolutive model match with the general tensor factorization formulation given in Eq. (5), we can rewrite Eq. (26) as

Please cite this article in press as: U. Şimşekli et al., Non-negative tensor factorization models for Bayesian audio processing, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.011

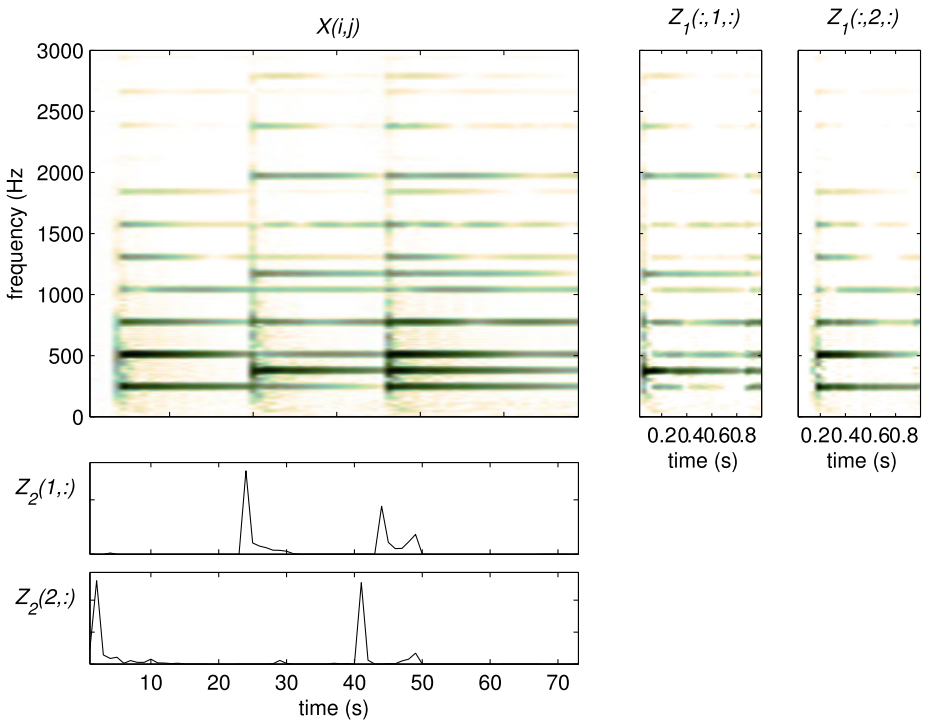


**Fig. 6.** Example spectrogram factored into a product of two factor matrices with the non-negative matrix factor deconvolution.

$$\begin{aligned}\hat{X}(i,j) &= \sum_k \sum_d Z_1(i,k,d) Z_2(k, \overbrace{j-d}^{\tau}) \\ &= \sum_k \sum_d \sum_\tau Z_1(i,k,d) Z_2(k,\tau) \delta(\tau - j + d) \\ &= \sum_k \sum_d \sum_\tau Z_1(i,k,d) Z_2(k,\tau) Z_3(\tau, j, d), \end{aligned} \tag{27}$$

where $Z_3(\tau, j, d) = \delta(\tau - j + d)$ is fixed. Methods presented in Section 4 can be used to infer factor matrices $Z_1(i,k,d)$ and $Z_2(k,\tau)$ of the model that are not fixed.

The above formulation uses convolution in time and is therefore able to model the temporal context. Convolution can equally well be done in frequency. When this model is applied on a constant-Q time-frequency representation where the frequencies are distributed logarithmically, we can model different pitches of an instrument with a single component [65]. Convolution in time and frequency can also be combined into a 2D deconvolution model, which takes into account both the temporal and spectral context [66].

### 6.2. PARAFAC model for multichannel audio

The models presented in the previous sections assumed a one-channel audio signal, for example recorded with a single microphone. In this case the data matrix $X(i,j)$ is the time-frequency magnitude spectrogram of the signal. On the other hand, we can also have multichannel audio recordings captured with multiple microphones, or produced in a studio by mixing tracks of individual sound sources.

The most straightforward way to represent multichannel signals is to calculate the time-frequency representation similarly from each channel, to obtain 3D data tensor matrix $X(i,j,c)$, where $c$ is the channel index. The simplest tensor factorization model for this data is the PARAFAC model presented in Eq. (2). Similarly to the basic NMF model, in this model, we assume that the data is an additive combination of components, each of which has a fixed spectrum $Z_1(i,k)$ and time-varying gain $Z_2(k,j)$. In addition to this, there is a channel gain factor matrix $Z_3(k,c)$, which indicates the gain of each component $k$ in each channel $c$ [67]. An illustration of this model is given in Fig. 7.

The above multichannel extension matches well with realistic multichannel audio recordings. For example when multichannel audio is produced at a recording studio, sources are positioned to different spatial locations by choosing their amplitude for each channel appropriately. When material is recorded with different microphones, the amplitude of a source captured by a microphone depends on the distance between the source and the microphone.

It should be noted that the above model cannot model phase differences between channels. Especially when multichannel material is recorded with microphones that are close to each other, phase differences between channels are an important cue that enables e.g. localization of sound sources. To enable modeling phase differences (dependencies between channels) in the tensor factorization framework, complex-valued extensions of non-negative matrix factorization have been used [68–70]. The magnitude of the data is typically modeled using the standard NMF model, whereas for the phases a separate phase model needs to be used.

### 6.3. Excitation-filter models

The models discussed in the previous sections are quite generic and could be used to analyze many other types of signals as well. On the other hand, audio signals have certain properties that can also be taken into account in more detail when designing a tensor factorization model. One such example is the excitation-filter model, where the signal is modeled as an excitation signal being filtered by a filter. For example in speech production, excitation signal generated by the lungs and vocal folds is acoustically filtered by the vocal tract. In the signal domain this can be modeled by the convolution of the excitation signal and the impulse response of the filter. In the magnitude spectrum domain that is typically used by tensor factorization models, this corresponds to point-wise multiplication of the excitation spectrum and the magnitude response of the filter [71].

Please cite this article in press as: U. Şimşekli et al., Non-negative tensor factorization models for Bayesian audio processing, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.011

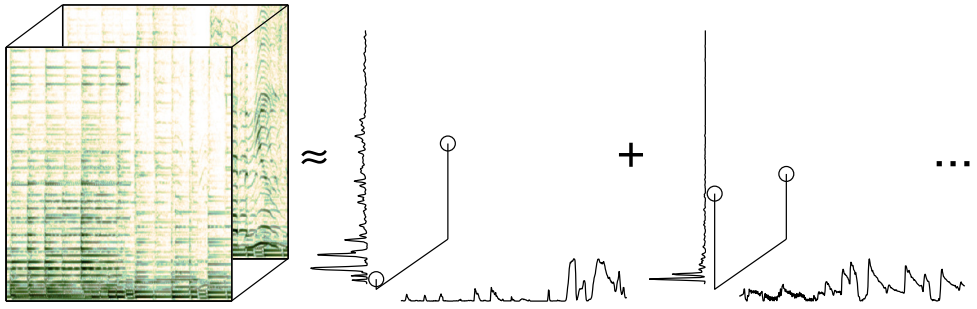

**Fig. 7.** An example stereo audio recording represented as a 3D tensor that consists of 2D spectrograms of the left and right channel is factored into a product of component spectral, gains and channel gains.

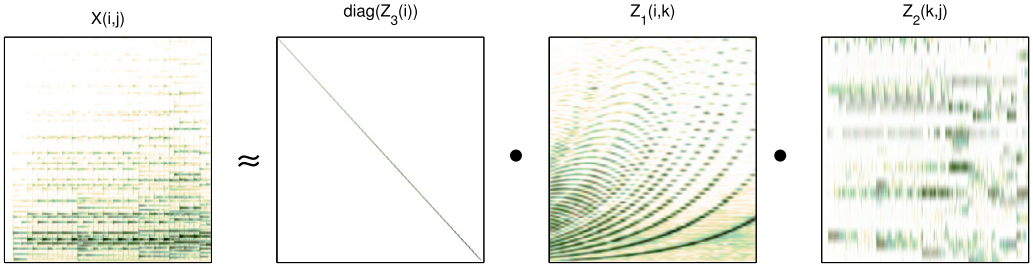


**Fig. 8.** Example spectrogram of piano music spectrogram $X(i, j)$ factored into a product of a diagonal matrix with the learned filter $Z_3(i)$ on the diagonal, matrix $Z_1(i, k)$ consisting of spectra of individual piano notes, and the matrix $Z_2(k, j)$ of frame-wise gains of the notes.

We can extend the basic NMF model in (1) by assuming that the spectrum $Z_1(i, k)$ of each component $k$ corresponds to an excitation spectrum, which are filtered with a single filter $Z_3(i)$. This model can be written as

$$\hat{X}(i, j) = \sum_k Z_1(i, k) Z_2(k, j) Z_3(i), \tag{28}$$

where, $Z_2(k, j)$ is the gain of each component in each frame. An illustration of this model is given in Fig. 8.

The above model is handy for example in adapting component spectra $Z_1(i, k)$ acquired at a training stage to model test data from different environment.

A generic excitation model for harmonic sound sources can for example consist of harmonic combs with different fundamental frequencies. The body responses of natural instruments are smooth in frequency, and in order to constrain the filter $Z_3(i)$ to be smooth, it can be modeled as a sum of spectral smooth elementary responses $Z_4(k, i)$ [72] as $Z_3(i) = \sum_l Z_4(l, i) Z_5(l)$, where $Z_5(l)$ are the response weights. This leads to total model

$$\hat{X}(i, j) = \sum_{k,l} Z_1(i, k) Z_2(k, j) Z_4(l, i) Z_5(l), \tag{29}$$

which is still a specific instance of the GCTF framework. More sophisticated excitation-filter models for music processing are presented in [73,74]. A generalized NMF-based framework for excitation-filter models is presented in [75].

## 7. Discussion and conclusions

In this paper, we reviewed a general framework for modeling and computing non-negative tensor decompositions. In particular, we described hierarchical modeling strategies that can be used to model structured domains such as audio. Given the model specification, an inference method can be derived in a straightforward manner. In modern applications, it is often required that side information is employed in an effective manner. The coupled tensor factorizations are useful in such settings. While such models are a lot more powerful, the inference algorithms are conceptually quite similar to the basic matrix factorizations, if the necessary computational primitives are carefully defined. We described the general form of the multiplicative update algorithms and a Markov Chain Monte Carlo procedure, namely the Gibbs sampler, that are applicable to any model topology that has multilinear structure.

We have also described a full family of probabilistic models, known as the Tweedie models [22], that is an important special case of the exponential dispersion models. One can obtain important distributions as special cases of the Tweedie distribution, such as the Gaussian, Poisson and gamma distributions. Modeling tensor factorizations with Tweedie models enabled us to have a flexible algorithmic framework that covers a broad range of cost functions such as the Euclidean, Kullback–Leibler, and Itakura–Saito divergences.

In coupled factorizations, the dispersion and power parameters of a model play an important role, since they together determine the cost function to be optimized. Typically, these parameters are selected manually. One possible future direction is to explore the applications of divergence learning methods [76] on audio processing, where the dispersion and power parameters are also estimated along with the latent factors. Similarly, the model order typically has a large effect on the interpretability and accuracy of a model. Even though methods for automatic estimation of model order have been proposed [10], there is still need for development.

In the last decade, matrix and tensor decomposition based methods had a big impact on audio processing applications. Yet, there are still significant challenges for future research. One important obstacle is scalability. Unfortunately, the inference algorithms described in this paper do not scale well with the size of the data, since they need to store all the data in the memory and pass over the whole dataset multiple times during the estimation process, making the inference impractical for very large data sets. Recently, parallel and distributed algorithms for making ML and MAP inference in matrix and tensor factorizations have been proposed [77,78] for certain model topologies (MF and PARAFAC). However, parallel and distributed inference methods for factorization models with arbitrary topologies are need to be explored.

Another important future research direction is online inference. In many audio applications, it is desirable to work with stream-

Please cite this article in press as: U. Şimşekli et al., Non-negative tensor factorization models for Bayesian audio processing, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.011

ing data to generate estimations on the fly. This is a topic in adaptive signal processing, where recursive estimation is the key mechanism. Currently, matrix and tensor decomposition methods are naturally formulated and solved as batch processing methods. It is desirable to develop theoretically sound methods that provide recursive solutions that work with a single pass over data. There is also need for developing more efficient optimization algorithms for basic tensor models so that they can be implemented in real-time systems.

In this paper we reviewed some of the most commonly used tensor models for audio signal processing. There exists a large number of models that were not covered, and it is expected that different variants of models will still be needed to model the structure of realistic audio signals and other information coupled with them. Data from audio and other domains have so far been coupled with tensor models in a relatively small number of studies, and it is expected that significant scientific advancements can be achieved by coupling information from multiple domains with tensor models.

This paper has described several models of increasing complexity for various audio processing tasks. Yet, in most models, modeling time frequency decompositions or merely power spectra is a simplification and discards detailed structure about the underlying signals. We anticipate that starting from the tensor factorization formalism, it would be possible to devise physically realistic models without compromising inferential efficiency.

## Acknowledgments

Umut Şimşekli and Taylan Cemgil are supported by the Turkish Scientific Research Council (TUBITAK) Grant 113M492, Advanced Parallel Optimization Algorithms for Big Data Analysis Using Machine Learning. Tuomas Virtanen is funded by the Academy of Finland, grant 258708. Umut Şimşekli is also funded by a PhD Scholarship from TUBITAK.

## Appendix A. Probability density and mass functions

- Exponential Distribution:

$$\mathcal{E}(x;a) = a\exp(-ax) \tag{A.1}$$

- Gamma Distribution:

$$\mathcal{G}(x;a,b) = x^{a-1}b^a\frac{\exp(-bx)}{\Gamma(a)} \tag{A.2}$$

- Multinomial Distribution:

$$\mathcal{M}(\mathbf{s};x,\mathbf{p}) = \delta(x-\sum_i s_i)x!\prod_{i=1}^{I}\frac{p_i^{s_i}}{s_i!} \tag{A.3}$$

where $\mathbf{s}=\{s_1,\ldots,s_I\}$ and $\mathbf{p}=\{p_1,\ldots,p_I\}$.

## Appendix B. Derivation of the multiplicative update rules

In order to solve the optimization problem given in Eq. (9), we make use of a gradient descent (GD) algorithm that is defined as follows:

$$Z_\alpha^{(i)} = Z_\alpha^{(i-1)} - \eta^{(i)}\nabla_{Z_\alpha}\mathcal{L}(X_{1:N_x}) \tag{B.1}$$

where $i$ denotes the number and $\nabla_{Z_\alpha}\mathcal{L}(\cdot)$ is the gradient of the objective function defined in Eq. (9). In this section, we will derive the multiplicative update rules (MUR) that are given in Eq. (10), that is actually a GD algorithm whose step sizes are chosen adaptively at each iteration.

Before deriving the partial derivatives with respect to $Z_\alpha$, let us write down the following derivatives that will become handy at the final derivation step. The general form the derivative of the $\beta$ divergence with respect to the second parameter is given as follows:

$$\frac{\partial d_p(x||\hat{x})}{\partial\hat{x}} = -x\hat{x}^{-p}+\hat{x}^{1-p} = \frac{\hat{x}-x}{\hat{x}^p}$$

Similarly, the derivative of the $\beta$-divergence $d_{p_\nu}(X_\nu(u_\nu);\hat{X}_\nu(u_\nu))$ with respect to an element of the model output tensor $\hat{X}_\nu(u_\nu)$ is given as follows:

$$\frac{\partial d_{p_\nu}(X_\nu(u_\nu);\hat{X}_\nu(u_\nu))}{\partial\hat{X}_\nu(u_\nu)} = \frac{\hat{X}_\nu(u_\nu)-X_\nu(u_\nu)}{\hat{X}_\nu(u_\nu)^{p_\nu}} \tag{B.2}$$

By using Eq. (B.2), we can obtain the partial derivatives that are required in the GD algorithm as follows:

$$\begin{aligned}\frac{\partial\mathcal{L}(X_{1:N_x})}{\partial Z_\alpha(v_\alpha)} &= \sum_\nu\frac{1}{\phi_\nu}\sum_{u_\nu}\frac{\partial d_{p_\nu}(X_\nu(u_\nu);\hat{X}_\nu(u_\nu))}{\partial\hat{X}_\nu(u_\nu)}\frac{\partial\hat{X}_\nu(u_\nu)}{\partial Z_\alpha(v_\alpha)}\\ &= \sum_\nu\left[R^{\nu,\alpha}\frac{1}{\phi_\nu}\sum_{\bar{v}_\alpha}\left(\frac{\hat{X}_\nu(u_\nu)-X_\nu(u_\nu)}{\hat{X}_\nu(u_\nu)^{p_\nu}}\right)\right.\\ &\quad\left.\times\prod_{\alpha'\neq\alpha}Z_{\alpha'}(v_{\alpha'})^{R^{\nu,\alpha}}\right]\end{aligned} \tag{B.3}$$

It is easy to verify that Eq. (B.3) can be re-written in the following form:

$$\begin{aligned}\nabla_{Z_\alpha}\mathcal{L}(X_{1:N_x}) &= \sum_\nu\left[R^{\nu,\alpha}\phi_\nu^{-1}\Delta_{\alpha,\nu}(\hat{X}_\nu^{1-p_\nu})\right]\\ &\quad-\sum_\nu\left[R^{\nu,\alpha}\phi_\nu^{-1}\Delta_{\alpha,\nu}(\hat{X}_\nu^{-p_\nu}\circ X_\nu)\right]\end{aligned}$$

where the function $\Delta_{\alpha,\nu}(\cdot)$ is defined in Eq. (11). Provided all the factors and observed tensors are non-negative, we can observe that in the right hand side of the above equation, both terms are non-negative. By making use of this observation, MUR method makes use of the following step size in order to preserve the non-negativity of the latent factors during the estimation process:

$$\eta^{(i)} = \frac{Z_\alpha^{(i-1)}}{\sum_\nu\left[R^{\nu,\alpha}\phi_\nu^{-1}\Delta_{\alpha,\nu}(\hat{X}_\nu^{1-p_\nu})\right]} \tag{B.4}$$

where the division is element-wise. By plugging this step size into Eq. (B.1), we obtain the multiplicative update rules given in Eq. (10). For MAP inference, the derivation procedure is the same up to adding one more term to the overall objective function.

## References

[1] A.T. Cemgil, U. Simsekli, Y.C. Subakan, Probabilistic latent tensor factorization framework for audio modeling, in: Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2011.
[2] M.W. Berry, M. Browne, Email surveillance using non-negative matrix factorization, Comput. Math. Organ. Theory 11 (3) (2005) 249–264.
[3] K. Devarajan, Nonnegative matrix factorization: an analytical and interpretive tool in computational biology, PLoS Comput. Biol. 4 (7) (2008) e1000029.
[4] A. Shashua, T. Hazan, Non-negative tensor factorization with applications to statistics and computer vision, in: Proceedings of the 22nd International Conference on Machine Learning, ACM, 2005, pp. 792–799.
[5] Y.-R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, A. Kelliher, Metafac: community discovery via relational hypergraph factorization, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2009, pp. 527–536.

Please cite this article in press as: U. Şimşekli et al., Non-negative tensor factorization models for Bayesian audio processing, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.011

[6] Y. Zhang, M. Roughan, W. Willinger, L. Qiu, Spatio-temporal compressive sensing and Internet traffic matrices, in: ACM SIGCOMM Comput. Commun. Rev., vol. 39, ACM, 2009, pp. 267–278.
[7] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (1999) 788–791.
[8] A. Cichocki, R. Zdunek, A.H. Phan, S. Amari, Nonnegative Matrix and Tensor Factorization, Wiley, 2009.
[9] C. Févotte, N. Bertin, J.-L. Durrieu, Nonnegative matrix factorization with the Itakura–Saito divergence. With application to music analysis, Neural Comput. 21 (3) (2009) 793–830.
[10] A.T. Cemgil, Bayesian inference for nonnegative matrix factorisation models, Comput. Intell. Neurosci. (2009).
[11] C.M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
[12] J. Carroll, J.-J. Chang, Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart–Young decomposition, Psychometrika 35 (3) (1970) 283–319.
[13] R.A. Harshman, Foundations of the PARAFAC procedure: models and conditions for an "explanatory" multi-modal factor analysis, UCLA Work. Pap. Phon. 16 (1) (1970) 1–84.
[14] R. Bro, PARAFAC. Tutorial and applications, Chemom. Intell. Lab. Syst. 38 (2) (1997) 149–171.
[15] L. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika 31 (3) (1966) 279–311.
[16] L. Le Magoarou, A. Ozerov, N.Q.K. Duong, Text-informed audio source separation using nonnegative matrix partial co-factorization, in: IEEE International Workshop on Machine Learning for Signal Processing, 2013.
[17] T. Wilderjans, E. Ceulemans, I. Van Mechelen, R. van den Berg, Simultaneous analysis of coupled data matrices subject to different amounts of noise, Br. J. Math. Stat. Psychol. 64 (2011) 277–290.
[18] E. Acar, G. Gurdeniz, M.A. Rasmussen, D. Rago, L.O. Dragsted, R. Bro, Coupled matrix factorization with sparse factors to identify potential biomarkers in metabolomics, Int. J. Knowl. Discov. Bioinform. 3 (3) (2012) 22–43.
[19] V.W. Zheng, B. Cao, Y. Zheng, X. Xie, Q. Yang, Collaborative filtering meets mobile recommendation: a user-centered approach, in: AAAI Conference on Artificial Intelligence, 2010.
[20] N. Seichepine, S. Essid, C. Févotte, O. Cappe, Soft nonnegative matrix co-factorization, IEEE Trans. Signal Process. 62 (22) (2014) 5940–5949.
[21] Y.K. Yılmaz, A.T. Cemgil, U. Şimşekli, Generalised coupled tensor factorisation, in: Advances in Neural Information Processing Systems, 2011.
[22] B. Jørgensen, The Theory of Dispersion Models, Chapman & Hall/CRC Monographs on Statistics & Applied Probability, 1997.
[23] A. Ozerov, N. Duong, L. Chevallier, On monotonicity of multiplicative update rules for weighted nonnegative tensor factorization, in: International Symposium on Nonlinear Theory and Its Applications, 2014.
[24] Y.K. Yilmaz, Generalized tensor factorization, Ph.D. thesis, Bogazici University, 2012.
[25] C.-J. Hsieh, I.S. Dhillon, Fast coordinate descent methods with variable selection for non-negative matrix factorization, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, USA, 2011.
[26] H. Van hamme, A diagonalized Newton algorithm for non-negative sparse coding, in: Proceedings of IEEE International Conference on Audio, Speech and Signal Processing, Vancouver, Canada, 2013.
[27] T. Virtanen, J. Gemmeke, B. Raj, Active-set Newton algorithm for overcomplete non-negative representations of audio, IEEE Trans. Audio Speech Lang. Process. 21 (11) (2013) 2277–2289.
[28] J. Kim, H. Park, Fast nonnegative matrix factorization: an active-set-like method and comparisons, SIAM J. Sci. Comput. 33 (6) (2011) 3261–3281.
[29] J.S. Liu, Monte Carlo Strategies in Scientific Computing, Springer, 2008.
[30] C. Févotte, A. Cemgil, Nonnegative matrix factorisations as probabilistic inference in composite models, in: Proceedings of European Signal Processing Conference, Glasgow, Scotland, 2009.
[31] U. Simsekli, A.T. Cemgil, Markov chain Monte Carlo inference for probabilistic latent tensor factorization, in: IEEE International Workshop on Machine Learning for Signal Processing, 2012, pp. 1–6.
[32] S. Chib, Marginal likelihood from the Gibbs output, J. Acoust. Soc. Am. 90 (432) (1995) 1313–1321.
[33] C. Fevotte, O. Cappe, A.T. Cemgil, Efficient Markov chain Monte Carlo inference in composite models with space alternating data augmentation, in: IEEE Statistical Signal Processing Workshop, 2011.
[34] J. Gemmeke, T. Virtanen, A. Hurmalainen, Exemplar-based sparse representations for noise robust automatic speech recognition, IEEE Trans. Audio Speech Lang. Process. 19 (7) (2011) 2067–2080.
[35] N. Bertin, R. Badeau, E. Vincent, Enforcing harmonicity and smoothness in Bayesian non-negative matrix factorization applied to polyphonic music transcription, IEEE Trans. Audio Speech Lang. Process. 18 (3) (2010) 538–549.
[36] P. Smaragdis, Convolutive speech bases and their application to supervised speech separation, IEEE Trans. Audio Speech Lang. Process. 15 (1) (2007) 1–12.
[37] T. Virtanen, Monaural sound source separation by non-negative matrix factorization with temporal continuity and sparseness criteria, IEEE Trans. Audio Speech Lang. Process. 15 (3) (2007) 1066–1074.
[38] D. Griffin, J. Lim, Signal estimation from modified short-time Fourier transform, IEEE Trans. Acoust. Speech Signal Process. 32 (1984) 236–242.
[39] J. Le Roux, E. Vincent, Consistent Wiener filtering for audio source separation, IEEE Signal Process. Lett. 20 (3) (2013) 217–220.
[40] B. Raj, T. Virtanen, S. Chaudhuri, R. Singh, Non-negative matrix factorization based compensation of music for automatic speech recognition, in: Interspeech, 2010, pp. 717–720.
[41] M.N. Schmidt, R.K. Olsson, Single-channel speech separation using sparse non-negative matrix factorization, in: Proceedings of the International Conference on Spoken Language Processing, Pittsburgh, USA, 2006.
[42] G.J. Mysore, P. Smaragdis, Non-negative approach to semi-supervised separation of speech from noise with the use of temporal dynamics, in: Proceedings of IEEE International Conference on Audio, Speech and Signal Processing, Prague, Czech Republic, 2011.
[43] C. Joder, F. Weninger, F. Eyben, D. Virette, B. Schuller, Real-time speech separation by semi-supervised nonnegative matrix factorization, in: Proceedings of the 9th International Conference on Latent Variable Analysis and Signal Separation, Tel-Aviv, Israel, 2012.
[44] U. Simsekli, A.T. Cemgil, Score guided musical source separation using generalized coupled tensor factorization, in: European Signal Processing Conference, 2012.
[45] N. Yasuraoka, H. Kameoka, T. Yoshioka, H.G. Okuno, I-divergence-based dereverberation method with auxiliary function approach, in: Proceedings of IEEE International Conference on Audio, Speech and Signal Processing, Prague, Czech Republic, 2011.
[46] R. Singh, B. Raj, P. Smaragdis, Latent-variable decomposition based dereverberation of monaural and multi-channel signals, in: Proceedings of IEEE International Conference on Audio, Speech and Signal Processing, Dallas, USA, 2010.
[47] K. Mahkonen, A. Hurmalainen, T. Virtanen, J.F. Gemmeke, Mapping sparse representation to state likelihoods in noise-robust automatic speech recognition, in: Interspeech, 2011, pp. 465–468.
[48] O. Dikmen, A. Mesaros, Sound event detection using non-negative dictionaries learned from annotated overlapping events, in: Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, USA, 2013.
[49] A. Hurmalainen, T. Virtanen, Learning state labels for sparse classification of speech with matrix deconvolution, in: Proceedings of Automatic Speech Recognition and Understanding Workshop, ASRU, Olomouc, Czech Republic, 2013.
[50] J. Gemmeke, H. Van hamme, B. Cranen, L. Boves, Compressive sensing for missing data imputation in noise robust speech recognition, IEEE J. Sel. Top. Signal Process. 4 (2) (2010) 272–287.
[51] J. Le Roux, H. Kameoka, N. Ono, A. de Cheveigné, S. Sagayama, Computational auditory induction as a missing-data model-fitting problem with Bregman divergence, SIAM J. Sci. Comput. 54 (5) (2011) 658–676.
[52] T. Virtanen, Separation of sound sources by convolutive sparse coding, in: Proceedings of ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing, Jeju, Korea, 2004.
[53] U. Simsekli, Y.K. Yilmaz, A.T. Cemgil, Score guided audio restoration via generalised coupled tensor factorisation, in: IEEE International Conference on Audio, Speech and Signal Processing, 2012.
[54] J. Nikunen, T. Virtanen, Object-based audio coding using non-negative matrix factorization for the spectrogram representation, in: Proceedings of the 128th Audio Engineering Society Convention, London, UK, 2010.
[55] J. Nikunen, T. Virtanen, M. Vilermo, Multichannel audio upmixing by time-frequency filtering using non-negative tensor factorization, J. Audio Eng. Soc. 60 (10) (2012) 794–806.
[56] A. Ozerov, A. Liutkus, R. Badeau, G. Richard, Coding-based informed source separation: nonnegative tensor factorization approach, IEEE Trans. Audio Speech Lang. Process. 21 (8) (2013) 1699–1712.
[57] G.J. Mysore, P. Smaragdis, B. Raj, Non-negative hidden Markov modeling of audio with application to source separation, in: Proceedings of the 9th International Conference on Latent Variable Analysis and Signal Separation, St. Malo, France, 2010.
[58] N. Mohammadiha, P. Smaragdis, A. Leijon, Prediction based filtering and smoothing to exploit temporal dependencies in NMF, in: Proceedings of IEEE International Conference on Audio, Speech and Signal Processing, Vancouver, Canada, 2013.
[59] C. Fevotte, J. Le Roux, J.R. Hershey, Non-negative dynamical system with application to speech and audio, in: Proceedings of IEEE International Conference on Audio, Speech and Signal Processing, Vancouver, Canada, 2013.
[60] U. Simsekli, J. Le Roux, J. Hershey, Hierarchical and coupled non-negative dynamical systems with application to audio modeling, in: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2013.
[61] A. Ozerov, C. Fevotte, M. Charbit, Factorial scaled hidden Markov model for polyphonic audio representation and source separation, in: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2009, pp. 121–124.

Please cite this article in press as: U. Şimşekli et al., Non-negative tensor factorization models for Bayesian audio processing, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.011

[62] R. Badeau, Gaussian modeling of mixtures of non-stationary signals in the time-frequency domain (hr-nmf), in: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2011, pp. 253–256.
[63] R. Badeau, M.D. Plumbley, Multichannel hr-nmf for modelling convolutive mixtures of non-stationary signals in the time-frequency domain, in: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2013, pp. 1–4.
[64] P. Smaragdis, Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs, in: Independent Component Analysis and Blind Signal Separation, 2004, pp. 494–499.
[65] D. FitzGerald, M. Cranitch, E. Coyle, Shifted nonnegative matrix factorisation for sound source separation, in: Proceedings of IEEE Workshop on Statistical Signal Processing, Bordeaux, France, 2005.
[66] M.N. Schmidt, M. Mørup, Nonnegative matrix factor 2-D deconvolution for blind single channel source separation, in: Proceedings of the 6th International Symposium on Independent Component Analysis and Blind Signal Separation, Charleston, USA, 2006.
[67] D. FitzGerald, M. Cranitch, E. Coyle, Extended nonnegative tensor factorisation models for musical source separation, Comput. Intell. Neurosci. 2008 (2008) 872425.
[68] A. Ozerov, C. Févotte, Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation, IEEE Trans. Audio Speech Lang. Process. 18 (3) (2010) 550–563.
[69] H. Sawada, H. Kameoka, S. Araki, N. Ueda, Formulations and algorithms for multichannel complex NMF, in: Proceedings of IEEE International Conference on Audio, Speech and Signal Processing, Prague, Czech Republic, 2011.
[70] J. Nikunen, T. Virtanen, Direction of arrival based spatial covariance model for blind sound source separation, IEEE Trans. Audio Speech Lang. Process. 22 (3) (2014) 727–739.
[71] T. Virtanen, A. Klapuri, Analysis of polyphonic audio using source-filter model and non-negative matrix factorization, in: Advances in Models for Acoustic Processing, Neural Information Processing Systems Workshop, Whistler, Canada, 2006, extended abstract.
[72] T. Heittola, A. Klapuri, T. Virtanen, Musical instrument recognition in polyphonic audio using source-filter model for sound separation, in: Proceedings of International Conference on Music Information Retrieval, Kobe, Japan, 2009.
[73] J.-L. Durrieu, G. Richard, B. David, C. Févotte, Source/filter model for unsupervised main melody extraction from polyphonic audio signals, IEEE Trans. Audio Speech Lang. Process. 18 (3) (2010) 564–575.
[74] J. Durrieu, B. David, G. Richard, A musically motivated mid-level representation for pitch estimation and musical audio source separation, IEEE J. Sel. Top. Signal Process. 5 (6) (2011) 1180–1191.
[75] A. Ozerov, E. Vincent, F. Bimbot, A general flexible framework for the handling of prior information in audio source separation, IEEE Trans. Audio Speech Lang. Process. 20 (4) (2012) 1118–1133.
[76] U. Simsekli, A.T. Cemgil, B. Ermis, Learning mixed divergences in coupled matrix and tensor factorization models, in: International Conference on Audio, Speech and Signal Processing, 2015.
[77] R. Gemulla, E. Nijkamp, P.J. Haas, Y. Sismanis, Large-scale matrix factorization with distributed stochastic gradient descent, in: ACM Special Interest Group on Knowledge Discovery and Data Mining, 2011.
[78] A. Beutel, A. Kumar, E.E. Papalexakis, P.P. Talukdar, C. Faloutsos, E.P. Xing, Flexifact: scalable flexible factorization of coupled tensors on Hadoop, in: SIAM International Conference on Data Mining, 2014.

**Umut Şimşekli** is a Ph.D. student in the Department of Computer Engineering at Bogazici University, Istanbul. His current interests are in machine learning, Bayesian statistics, audio and music processing; more specifically on non-negative matrix and tensor factorization models, divergence learning, and large-scale distributed inference. He received his B.Sc. and M.Sc. degrees in 2008 and 2010, respectively.

**Tuomas Virtanen** is an Academy Research Fellow and Associate Professor (tenure track) at Department of Signal Processing, Tampere University of Technology (TUT), Finland, where he is leading the Audio Research Group. He received the M.Sc. and Doctor of Science degrees in information technology from TUT in 2001 and 2006, respectively. He has also been working as a research associate at Cambridge University Engineering Department, UK. He has been developing methods for single-channel sound source separation using non-negative matrix factorization based techniques, and noise-robust speech recognition, music content analysis, and audio event detection. In addition to the above topics, his research interests include content analysis of audio signals in general and machine learning. He has authored more than 100 scientific publications on the above topics.

**Ali Taylan Cemgil** received his Ph.D. (2004) from SNN, Radboud University Nijmegen, the Netherlands. Between 2004 and 2008 he worked as a postdoctoral researcher at Amsterdam University and the Signal Processing and Communications Lab., University of Cambridge, UK. He is currently an associate professor of Computer Engineering at Bogazici University, Istanbul, Turkey. He is an associate editor of IEEE Signal Processing Letters. His research interests are in Bayesian statistical methods, approximate inference, machine learning and signal processing.

Please cite this article in press as: U. Şimşekli et al., Non-negative tensor factorization models for Bayesian audio processing, Digital Signal Process. (2015), http://dx.doi.org/10.1016/j.dsp.2015.03.011