Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Leveraging Features and Networks for Probabilistic Tensor Decomposition

Piyush Rai, Yingjian Wang, Lawrence Carin
{piyush.rai,yingjian.wang,lcarin}@duke.edu
‘ ECE Department, Duke University
Durham, NC 27708

Abstract

We present a probabilistic model for tensor decomposition where one or more tensor modes may have side-information about the mode entities in form of their features and/or their adjacency network. We consider a Bayesian approach based on the Canonical PARAFAC (CP) decomposition and enrich this single-layer decomposition approach with a *two-layer* decomposition. The second layer fits a factor model for each layer-one factor matrix and models the factor matrix via the mode entities' features and/or the network between the mode entities. The second-layer decomposition of each factor matrix also learns a *binary* latent representation for the entities of that mode, which can be useful in its own right. Our model can handle both continuous as well as binary tensor observations. Another appealing aspect of our model is the simplicity of the model inference, with easy-to-sample Gibbs updates. We demonstrate the results of our model on several benchmarks datasets, consisting of both real and binary tensors.

Introduction

Learning from multiway and multirelational data is becoming more and more ubiquitous in the era of big data. Multiway tensor data (Kolda and Bader 2009; Acar and Yener 2009), in particular, arises in diverse applications, such as recommender systems (Yin et al. 2013), multirelational networks (Nickel, Tresp, and Kriegel 2011), and brain-computer imaging (Cichocki 2013), among others. Tensor decomposition methods (Kolda and Bader 2009; Acar and Yener 2009) provide an effective way to extract latent factors from such data, and are now routinely used for tensor *completion* of sparse, incomplete tensors. Probabilistic tensor decomposition methods (Chu and Ghahramani 2009; Xiong et al. 2010; Xu, Yan, and Qi 2013; Rai et al. 2014) are especially appealing because they can deal with with diverse data types and missing data in a principled way via a proper generative model.

In many problems of practical interest, in addition to the tensor data, there is also additional information for one of the more tensor modes. For example, consider a tensor data containing a three-way user-location-activity relationship denoted by $\mathcal{Y}$ where $\mathcal{Y}_{ijk} = 1$ denotes that user i visited location j and performed activity k. In addition to the tensor data $\mathcal{Y}$, there may be additional information associated with one or more of the modes of the tensor. For examples, we may be given user attributes, location attributes, and a user-user social network (Fig 1 (a)). Leveraging such additional sources of information can lead to improved tensor decomposition and completion, especially when a significant amount of tensor data is missing, or data in an entire slice is missing, or in cold-start problems involving tensor data (Ermiş, Acar, and Cemgil 2013; Narita et al. 2012). There has been a recent interest in tensor decomposition methods that leverage side-information associated with one or more of the tensor modes (Ermiş, Acar, and Cemgil 2013; Narita et al. 2012).

Motivated by this problem, in this paper, we present a probabilistic, fully Bayesian approach for tensor decomposition and completion with side-information. Our approach has the following key advantages: (i) ability to incorporate side-information from modes that have side-information in form of a feature matrix and/or an adjacency network between the mode entities, (ii) a two-layer decomposition of the tensor, with the second layer decomposition also providing a binary vector representation for entities in each mode, which can be useful in its own right, (iii) a generative, fully Bayesian model which allows modeling both real-valued and binary-valued tensors and infers the appropriate tensor rank from the data, and (iv) conjugate Bayesian inference for both real and binary data, allowing easy to derive Gibbs sampling updates to facilitate a fully Bayesian analysis.

The Model

At the core of our proposed model is the Canonical PARAFAC (CP) decomposition (Kolda and Bader 2009; Xiong et al. 2010; Rai et al. 2014) model. Formally, a K-way tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_K}$ can be represented as:

$$\mathcal{X} = \sum_{r=1}^{R} \lambda_r \cdot \boldsymbol{u}_r^{(1)} \circ \boldsymbol{u}_r^{(2)} \circ \cdots \circ \boldsymbol{u}_r^{(K)}$$

where n_k denotes the dimension of tensor $\mathcal{X}$ along the k^{th} way (or *mode*) of the tensor, $\boldsymbol{u}_r^{(k)} \in \mathbb{R}^{n_k}$, '$\circ$' denotes the outer product, and R is the *rank* of the tensor $\mathcal{X}$. This construction essentially expresses the tensor $\mathcal{X}$ as a sum of

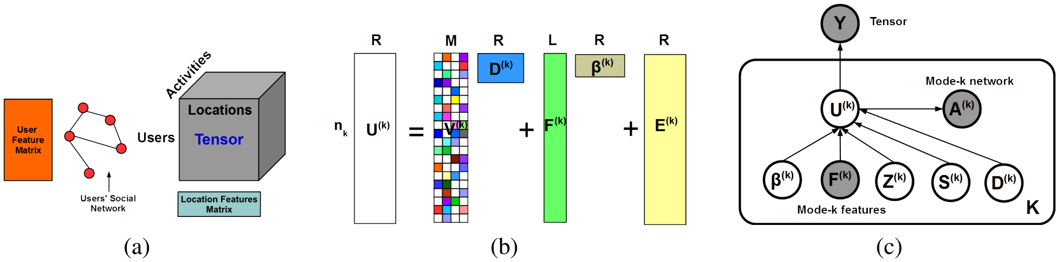

Figure 1: (a) Illustration of a tensor with side information available for two of its modes. (b) Second-layer factorization of the mode k factor matrix as in Equation 1. (c) the graphical model

R rank-1 tensors. Another concise representation can be via a diagonal tensor $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_R)$ and a set of *mode factor matrices* $\boldsymbol{U}^{(k)} = [\boldsymbol{u}_1^{(k)}, \boldsymbol{u}_2^{(k)}, \cdots, \boldsymbol{u}_R^{(k)}]$, $k = 1, 2, \cdots, K$, where $\boldsymbol{U}^{(k)} \in \mathbb{R}^{n_k \times R}$ is the factor matrix of mode k of the tensor: $\mathcal{X} = [[\Lambda, \boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \cdots, \boldsymbol{U}^{(K)}]]$.

The probabilistic CP decomposition models (Xiong et al. 2010; Rai et al. 2014) assume a Gaussian prior on the columns/rows of each factor matrix, e.g., $\boldsymbol{u}_r^{(k)} \sim \mathcal{N}or(0, \boldsymbol{I}_{n_k})$ or $\boldsymbol{u}_{i_k:}^{(k)} \sim \mathcal{N}or(0, \boldsymbol{I}_R)$. A key question is specifying/inferring the rank of the decomposition. In this paper, we will follow the multiplicative Gamma process (MGP) (Bhattacharya and Dunson 2011; Rai et al. 2014) construction: $\lambda_r \overset{\text{ind}}{\sim} \mathcal{N}or(0, \tau_r^{-1})$, for $1 \leq r \leq R$, $\tau_r = \prod_{l=1}^{r} \delta_l$, $\delta_1 \sim \mathrm{Ga}(a_1, 1)$, $\delta_r \sim \mathrm{Ga}(a_2, 1)$, with $a_2 > 1$, for $1 < r \leq R$, which provides an efficient way to infer the rank.

Two-Layer CP Decomposition with Side-Information

The existing probabilistic CP decomposition methods (Xiong et al. 2010; Rai et al. 2014) are based on a single-layer decomposition of the underlying tensor $\mathcal{X}$ which admits a decomposition in terms of the factor matrices $\{\boldsymbol{U}^{(k)}\}_{k=1}^{K}$, and each factor matrix is assumed drawn from a Gaussian prior. Moreover, these methods cannot leverage side-information (e.g., a feature matrix $\boldsymbol{F}^{(k)}$ and/or a network $\boldsymbol{A}^{(k)}$) that may be available for one or more of the tensor modes. We first show how the side-information in form of the mode k feature matrix $\boldsymbol{F}^{(k)}$ can be incorporated via our proposed *two-layer* CP decomposition. We will then show how the mode k network $\boldsymbol{A}^{(k)}$ can be incorporated in the model. We will refer to our model as BCPFN (abbreviated for **B**ayesian **CP** with **F**eatures and **N**etworks).

Incorporating Mode Features: The mode features $\boldsymbol{F}^{(k)}$ are introduced in the model via a two-layer approach, which models the mode k latent factor matrix $\boldsymbol{U}^{(k)}$ by another factor model:

$$\boldsymbol{U}^{(k)} = \boldsymbol{V}^{(k)}\boldsymbol{D}^{(k)} + \boldsymbol{F}^{(k)}\boldsymbol{\beta}^{(k)} + \boldsymbol{E}^{(k)} \qquad (1)$$

The second-layer factor model in Equation 1 represents each mode factor matrix $\boldsymbol{U}^{(k)} \in \mathbb{R}^{n_k \times R}$ in terms of a low-rank component $\boldsymbol{V}^{(k)}\boldsymbol{D}^{(k)}$ (where $\boldsymbol{V}^{(k)} \in \mathbb{R}^{n_k \times M}, \boldsymbol{D}^{(k)} \in \mathbb{R}^{M \times R}$, with $M \leq R$), plus a regression model on the *observed* mode feature/attribute matrix $\boldsymbol{F}^{(k)} \in \mathbb{R}^{n_k \times L}$ via a set of regression coefficients $\boldsymbol{\beta}^{(k)} \in \mathbb{R}^{L \times R}$. Note that it is possible to have more than one feature matrix for mode k, in which case we can have a separate regression model for each feature matrix. Also, in the absence of the mode feature matrix, the regression term will be ignored. The matrix $\boldsymbol{E}^{(k)} \in \mathbb{R}^{n_k \times R}$ captures the idiosyncratic noise. We further assume $\boldsymbol{V}^{(k)}$ to be a *sparse* matrix (see Figure 1 (b)) of size $n_k \times M$, modeled as an element-wise product of a binary matrix $\boldsymbol{Z}^{(k)} \in \{0, 1\}^{n_k \times M}$ and a real-valued matrix $\boldsymbol{S}^{(k)} \in \mathbb{R}^{n_k \times M}$.

This model essentially provides a two-layered representation for each of the n_k entities in mode k: one in terms of the mode factor matrix $\boldsymbol{U}^{(k)} \in \mathbb{R}^{n_k \times R}$ from the first layer of CP decomposition, and the other in terms of the sparse coefficients $\boldsymbol{V}^{(k)} \in \mathbb{R}^{n_k \times M}$ of a layer-two *dictionary* $\boldsymbol{D}^{(k)} \in \mathbb{R}^{M \times R}$ with the sparse matrix $\boldsymbol{V}^{(k)}$ represented as an element-wise product of a binary matrix and a real-valued matrix, $\boldsymbol{V}^{(k)} = \boldsymbol{Z}^{(k)} \odot \boldsymbol{S}^{(k)}$ (akin to the way in Beta-Bernoulli process based latent factor models (Paisley and Carin 2009)), obtained from the second layer of decomposition of $\boldsymbol{U}^{(k)}$ (Equation 1).

In particular, the sparse representation $\boldsymbol{V}^{(k)}$ (or the binary matrix $\boldsymbol{Z}^{(k)}$) can be useful in its own right. For example, it immediately yields a multiway generalization of overlapping stochastic blockmodels (Miller, Jordan, and Griffiths 2009; Latouche et al. 2011) by representing each mode k entity $i_k \in \{1, \ldots, n_k\}$ in terms of a sparse real-valued/binary vector. This representation also provides a way to simultaneously learn binary hash codes for multimodal data (each modality would correspond to one tensor mode), generalizing the existing methods for learning hash codes, which are limited to data having only two modalities (Zhen and Yeung 2012; Bronstein et al. 2010; Zhen and Yeung). In this particular setting for multimodal hashing, a binary-valued main tensor could correspond to the *multimodal* relational constraints between the objects in different modalities, and raw features of objects in each modality can be used as side-information. Besides, this two-layer representation can also be seen as a *nonlinear* decomposition of the main tensor $\mathcal{Y}$. This is in contrast with standard CP decomposition which performs a linear decomposition of the tensor.

The representation of mode k factor matrix $\boldsymbol{U}^{(k)}$ in terms of mode k observed features $\boldsymbol{F}^{(k)}$ is also appealing in *cold-start* problems in recommender systems. In these problems, a new entity i_* in some tensor mode may not have any information in the tensor data. To predict its latent factors $\boldsymbol{U}^{(k)}_{i_*}$, it observed features $\boldsymbol{F}^{(k)}_{i_*}$ can be leveraged using the factor model given in Equation 1. This approach to deal with cold-start problem is also reminiscent of the regression latent factor model (Agarwal and Chen 2009) for two-dimensional matrices, which are a special case of multiway tensors.

Incorporating Mode Network: Often, a network between the entities of some tensor mode(s) may be given. We denote the mode k network by a binary matrix $\boldsymbol{A}^{(k)} \in \{0,1\}^{n_k \times n_k}$, with its elements denoted as $A^{(k)}_{i_k j_k}, 1 \leq i_k, j_k \leq n_k$, where $A^{(k)}_{i_k j_k} = 1$ denotes that entity i_k and j_k are similar (have an edge between them). We assume that the edge probability $p(A^{(k)}_{i_k j_k})$ is a logistic function of the inner product of factors $\boldsymbol{u}_{i_k} \in \mathbb{R}^{1\times R}$ and $\boldsymbol{u}_{j_k} \in \mathbb{R}^{1\times R}$ of entires i_k and j_k, respectively:

$$p(A^{(k)}_{i_k j_k} = 1) = \frac{1}{1 + \exp(-\boldsymbol{u}^{(k)}_{i_k} \boldsymbol{u}^{(k)\top}_{j_k})} \qquad (2)$$

Note that the network $\boldsymbol{A}^{(k)}$ need not be fully observed; we may only be given the network information for a subset of mode k entity-pairs, in which case we only model the observed edges.

Modeling Layer-Two Latent Variables: We now describe the prior distributions over the remaining latent variables of the proposed model. For the variables in layer-two factor model $\boldsymbol{U}^{(k)} = (\boldsymbol{Z}^{(k)} \odot \boldsymbol{S}^{(k)})\boldsymbol{D}^{(k)} + \boldsymbol{F}^{(k)}\boldsymbol{\beta}^{(k)} + \boldsymbol{E}^{(k)}$, we assume the binary matrix $\boldsymbol{Z}^{(k)} \in \{0,1\}^{n_k \times M}$ is drawn from on a Beta-Bernoulli process (Paisley and Carin 2009):

$$\begin{aligned} \boldsymbol{z}^{(k)}_{i_k m} &\sim \text{Bernoulli}(\pi^{(k)}_m) \\ \pi^{(k)}_m &\sim \text{Beta}(\alpha, \beta) \end{aligned}$$

where $\boldsymbol{z}^{(k)}_{i_k m}, i_k = 1, 2, \cdots, n_k, m = 1, \ldots, M$ is the (i_k, m)-th element of $\boldsymbol{Z}^{(k)}$, and α and β are the hyper-parameters for the Beta prior.

The priors on $\boldsymbol{D}^{(k)} \in \mathbb{R}^{M\times R}$ and $\boldsymbol{S}^{(k)} \in \mathbb{R}^{n_k \times M}$ are given by $\boldsymbol{d}^{(k)}_m \sim \mathcal{N}or(0, R^{-1}\boldsymbol{I}_R)$, $\boldsymbol{s}^{(k)}_{i_k} \sim \mathcal{N}or(0, \gamma_s^{-1}\boldsymbol{I}_M)$, where $\boldsymbol{d}^{(k)}_m \in \mathbb{R}^{1\times R}, m = 1, 2, \cdots, M$, $\boldsymbol{s}^{(k)}_{i_k} \in \mathbb{R}^{1\times M}, i_k = 1, 2, \cdots, n_k$, and γ_s is the precision of s_{i_k} with a diffuse Ga$(10^{-6}, 10^{-6})$ prior.

The mode-specific regression coefficients matrix $\boldsymbol{\beta} \in \mathbb{R}^{L\times R}$ is assumed drawn as $\boldsymbol{\beta}^{(k)}_l \sim \mathcal{N}or(0, \rho^2\boldsymbol{I}_R), l = 1, 2, \cdots, L$, where $\boldsymbol{\beta}^{(k)}_l \in \mathbb{R}^{1\times R}$. Finally, $\boldsymbol{E}^{(k)} \in \mathbb{R}^{n_k\times R}$ is the matrix of residuals with rows $\boldsymbol{e}^{(k)}_{i_k} \sim \mathcal{N}or(0, \sigma^2\boldsymbol{I}_R), i_k = 1, 2, \cdots, n_k$.

Inference

The goal of inference in our model is to infer the latent variables $\{\boldsymbol{U}^{(k)}, \boldsymbol{Z}^{(k)}, \boldsymbol{S}^{(k)}, \boldsymbol{D}^{(k)}, \boldsymbol{\beta}^{(k)}\}_{k=1}^K$, and the latent variables $\{\delta_r\}_{r=1}^R$, and $\{\lambda_r\}_{r=1}^R$ associated with the multiplicative Gamma process construction, given real or binary tensor observations which we will denote by $\mathcal{Y} = \{y_{\boldsymbol{i}}\}_{\boldsymbol{i}\in I}$, where I is the index set of all the tensor observations and an index $\boldsymbol{i}$ is of the form $i_1 i_2 \ldots i_K$. When the tensor observations are real-valued, we assume the Gaussian noise model: $p(\mathcal{Y}|\mathcal{X}) = \prod_{\boldsymbol{i}} \mathcal{N}or(y_{\boldsymbol{i}}|x_{\boldsymbol{i}}, \tau_\epsilon^{-1})$, where $x_{\boldsymbol{i}} = \sum_{r=1}^R \lambda_r \prod_{k=1}^K u^{(k)}_{i_k r}$ and τ_ϵ is the precision of noise. When the tensor observations are binary-valued (e.g., for multirelational social network data), we assume the logistic link function: $p(\mathcal{Y}|\mathcal{X}) = \prod_{\boldsymbol{i}} (\frac{1}{1+\mathrm{e}^{-x_{\boldsymbol{i}}}})^{y_{\boldsymbol{i}}} (\frac{\mathrm{e}^{-x_{\boldsymbol{i}}}}{1+\mathrm{e}^{-x_{\boldsymbol{i}}}})^{1-y_{\boldsymbol{i}}}$. Apart from the tensor observations $\mathcal{Y}$, we may also have side-information in form of the mode k feature matrix $\boldsymbol{F}^{(k)}$ and/or the mode k network $\boldsymbol{A}^{(k)}$.

Since exact inference is intractable in the model, we use Markov Chain Monte Carlo (MCMC) to perform approximate inference. One appealing aspect of the inference in our model, as we will show, is that inference can be performed using closed-form Gibbs sampling updates, even when the tensor observations are binary, or when the likelihood term involves the binary mode network $\boldsymbol{A}^{(k)}$. This is accomplished via using the Pólya Gamma sampling strategy (Polson, Scott, and Windle 2012), which leads to locally conjugate Gibbs updates in our model, enabling efficient inference. Recently, (Rai et al. 2014) also employed Pólya-Gamma sampling for Bayesian CP decomposition of binary tensors but only considered a single-layer decomposition and did not consider side-information in form of features and networks associated with the tensor modes. We compare with their method in our experiments.

For brevity, we provide all the sampling update equations in the appendix.

In this section, we provide the sampling update equations for $\{\boldsymbol{U}^{(k)}, \boldsymbol{Z}^{(k)}, \boldsymbol{S}^{(k)}, \boldsymbol{D}^{(k)}$, and $\boldsymbol{\beta}^{(k)}\}_{k=1}^K$. Update equations for the variables $\{\delta_r\}_{r=1}^R$, and $\{\lambda_r\}_{r=1}^R$ associated with the multiplicative Gamma process are provided in the appendix.

Updating $\boldsymbol{U}^{(k)}$: Note that in our CP decomposition model with side information, the mode k factor matrix $\boldsymbol{U}^{(k)}$ generates two types of observations: the main tensor $\mathcal{Y}$ and the mode k network $\boldsymbol{A}^{(k)}$. When the main tensor $\mathcal{Y}$ is real-valued, i.e., $p(\mathcal{Y}|\mathcal{X}) = \prod_{\boldsymbol{i}} \mathcal{N}or(y_{\boldsymbol{i}}|x_{\boldsymbol{i}}, \tau_\epsilon^{-1})$, we have:

$$x_{\boldsymbol{i}} = c^{(k)}_{\boldsymbol{i}_k r} u^{(k)}_{\boldsymbol{i}_k r} + d^{(k)}_{\boldsymbol{i}_k r} \qquad (3)$$

where $c^{(k)}_{\boldsymbol{i}_k r} = \lambda_r \prod_{k' \neq k} u^{(k)}_{\boldsymbol{i}_{k'} r}$ and $d^{(k)}_{\boldsymbol{i}_k r} = \sum_{r' \neq r} \lambda_{r'} \prod_{k=1}^K u^{(k)}_{\boldsymbol{i}_k r'}$. Since $\boldsymbol{U}^{(k)}$ also depends on the network observations $\boldsymbol{A}^{(k)}$ which are binary-valued, due to the logistic link function, the likelihood is not conjugate. We use the Pólya-Gamma sampling strategy(Polson, Scott, and Windle 2012) in this case to facilitate deriving closed-form Gibbs sampling update. In order to do so, we first denoting $B^{(k)}_{i_k j_k} = \boldsymbol{u}^{(k)}_{i_k} \boldsymbol{u}^{(k)\top}_{j_k} = \sum_{r=1}^R u^{(k)}_{i_k r} u^{(k)}_{j_k r}$, which leads to the following:

$$u_{i_k r}^{(k)} = \frac{1}{u_{j_k r}^{(k)}} B_{i_k j_k}^{(k)} - \frac{\sum_{r' \neq r} u_{i_k r'}^{(k)} u_{j_k r'}^{(k)}}{u_{j_k r}^{(k)}} \tag{4}$$

Then with the $\psi_{i_k j_k}^{(k)} \sim \mathrm{PG}(1, B_{i_k j_k}^{(k)})$, where PG denotes the Pólya-Gamma distribution (Polson, Scott, and Windle 2012), $\boldsymbol{u}_{i_k}^{(k)}$ is drawn from the Gaussian:

$$\boldsymbol{u}_{i_k}^{(k)} \sim \mathcal{N}or(\hat{\boldsymbol{\mu}}_{i_k}^{(k)}, \hat{\Sigma}_{i_k}^{(k)})$$

where the posterior covariance is given by $\hat{\Sigma}_{i_k}^{(k)} = ({\Sigma_{i_k}^{(k)}}^{-1} + \Omega_{i_k}^{(k)})^{-1}$, $\Omega_{i_k}^{(k)} = \mathrm{diag}(\tau_{i_k 1}^{(k)}, \tau_{i_k 2}^{(k)}, \cdots, \tau_{i_k R}^{(k)})$, and using Equation 3, 4, and following the Pólya Gamma sampling scheme (Polson, Scott, and Windle 2012), we have

$$\tau_{nr}^{(k)} = \tau_e \sum_{\mathcal{Y}, i_k = n} {c_{i_k r}^{(k)}}^2 + \sum_{A, i_k = n} {u_{j_k r}^{(k)}}^2 \psi_{i_k j_k}^{(k)}, \quad 1 \leq r \leq R$$

The posterior mean is defined as

$$\hat{\boldsymbol{\mu}}_{i_k}^{(k)} = \hat{\Sigma}_{i_k}^{(k)} ({\Sigma_{i_k}^{(k)}}^{-1} \boldsymbol{\mu}_{i_k}^{(k)} + \Omega_{i_k}^{(k)} \boldsymbol{\alpha}_{i_k}^{(k)})$$

with $\boldsymbol{\mu}_{i_k}^{(k)} = (\boldsymbol{s}_{i_k}^{(k)} \odot \boldsymbol{z}_{i_k}^{(k)}) D^{(k)} + T_{i_k}^{(k)} F^{(k)}$, $\Sigma_{i_k}^{(k)} = \gamma_\epsilon^{-1} \boldsymbol{I}_R$, $\boldsymbol{\alpha}_{i_k}^{(k)} = [\alpha_{i_k 1}^{(k)}, \alpha_{i_k 2}^{(k)}, \cdots, \alpha_{i_k R}^{(k)}]^\top$ and again, following the Pólya Gamma sampling scheme (Polson, Scott, and Windle 2012), we have

$$\alpha_{nr}^{(k)} = (\tau_{nr}^{(k)})^{-1} [\tau_e \sum_{\mathcal{Y}, i_k = n} c_{i_k r}^{(k)} (y_{\boldsymbol{i}} - d_{i_k r}^{(k)})$$
$$+ \sum_{A, i_k = n} u_{j_k r}^{(k)} (A_{i_k j_k}^{(k)} - 0.5 - \psi_{i_k j_k}^{(k)} \sum_{r' \neq r} u_{i_k r'}^{(k)} u_{j_k r'}^{(k)})], 1 \leq r \leq R$$

When the tensor observations $\mathcal{Y}$ are binary, i.e.,

$$p(\mathcal{Y} | \mathcal{X}) = \prod_{\boldsymbol{i}} (\frac{1}{1 + \mathrm{e}^{-x_{\boldsymbol{i}}}})^{y_{\boldsymbol{i}}} (\frac{\mathrm{e}^{-x_{\boldsymbol{i}}}}{1 + \mathrm{e}^{-x_{\boldsymbol{i}}}})^{1 - y_{\boldsymbol{i}}}$$

we need to introduce another set of Pólya Gamma variables(Rai et al. 2014) for each tensor observation: $\phi_i \sim \mathrm{PG}(1, x_i)$. With the inclusion of these variables, the updates of $\boldsymbol{u}_{i_k}^{(k)}, 1 \leq i_k \leq n_k, 1 \leq k \leq K$ are similar to the real $\mathcal{Y}$ case, and are given by

$$\tau_{nr}^{(k)} = \sum_{\mathcal{Y}, i_k = n} {c_{i_k r}^{(k)}}^2 \phi_{\boldsymbol{i}} + \sum_{A, i_k = n} {u_{j_k r}^{(k)}}^2 \psi_{i_k j_k}^{(k)}, \quad 1 \leq r \leq R$$

$$\alpha_{nr}^{(k)} = (\tau_{nr}^{(k)})^{-1} [\sum_{\mathcal{Y}, i_k = n} c_{i_k r}^{(k)} (y_{\boldsymbol{i}} - 0.5 - \phi_{\boldsymbol{i}} d_{i_k r}^{(k)})$$
$$+ \sum_{A, i_k = n} u_{j_k r}^{(k)} (A_{i_k j_k}^{(k)} - 0.5 - \psi_{i_k j_k}^{(k)} \sum_{r' \neq r} u_{i_k r'}^{(k)} u_{j_k r'}^{(k)})], 1 \leq r \leq R$$

Updating $\boldsymbol{Z}^{(k)}$: For the update of $z_{i_k m}^{(k)}$, $i_k = 1, 2, \cdots, n_k,\ m = 1, 2, \cdots, M,\ k = 1, 2, \cdots, K$,

$$p(z_{i_k m}^{(k)} | -) \propto \mathcal{N}or(\boldsymbol{u}_{i_k}^{(k)} | (\boldsymbol{s}_{i_k}^{(k)} \odot \boldsymbol{z}_{i_k}^{(k)}) D^{(k)} + \boldsymbol{F}_{i_k}^{(k)} \boldsymbol{\beta}^{(k)}, \gamma_\epsilon^{-1} \boldsymbol{I}_R) \times \mathrm{Bern}(z_{i_k m}^{(k)} | \pi_m^{(k)}) \tag{5}$$

Therefore $z_{i_k m}^{(k)} \sim \mathrm{Bern}(\frac{c_1}{c_1 + c_0})$, where $c_1 = \pi_m^{(k)} \exp[-\frac{\gamma_\epsilon}{2} ({s_{i_k m}^{(k)}}^2 \boldsymbol{d}_m^{(k)} {\boldsymbol{d}_m^{(k)}}^\top - 2 s_{i_k m}^{(k)} \Delta_{i_k, -m}^{(k)} {\boldsymbol{d}_m^{(k)}}^\top)]$, $c_0 = 1 - \pi_m^{(k)}$, and $\Delta_{i_k, -m}^{(k)} = \boldsymbol{u}_{i_k}^{(k)} - \boldsymbol{F}_{i_k}^{(k)} \boldsymbol{\beta}^{(k)} - (\boldsymbol{s}_{i_k}^{(k)} \odot \boldsymbol{z}_{i_k}^{(k)}) D^{(k)} + (s_{i_k m}^{(k)} \odot z_{i_k m}^{(k)}) \boldsymbol{d}_m^{(k)}$.

Updating $\boldsymbol{S}^{(k)}$: For the update of $s_{i_k m}^{(k)}$, $i_k = 1, 2, \cdots, n_k,\ m = 1, 2, \cdots, M,\ k = 1, 2, \cdots, K$,

$$p(s_{i_k m}^{(k)} | -) \propto \mathcal{N}or(\boldsymbol{u}_{i_k}^{(k)} | (\boldsymbol{s}_{i_k}^{(k)} \odot \boldsymbol{z}_{i_k}^{(k)}) D^{(k)} + \boldsymbol{F}_{i_k}^{(k)} \boldsymbol{\beta}^{(k)}, \gamma_\epsilon^{-1} \boldsymbol{I}_R) \times \mathcal{N}or(\boldsymbol{s}_{i_k}^{(k)} | \boldsymbol{0}, \gamma_s^{-1} \boldsymbol{I}_M) \tag{6}$$

Therefore $s_{i_k m}^{(k)} \sim \mathcal{N}or(\mu_{s_{i_k m}^{(k)}}, \Sigma_{s_{i_k m}^{(k)}})$, where $\Sigma_{s_{i_k m}^{(k)}} = (\gamma_s + \gamma_\epsilon {z_{i_k m}^{(k)}}^2 \boldsymbol{d}_m \boldsymbol{d}_m^\top)^{-1}$, and $\mu_{s_{i_k m}^{(k)}} = \gamma_\epsilon \Sigma_{s_{i_k m}^{(k)}} z_{i_k m}^{(k)} \Delta_{i_k, -m}^{(k)} \boldsymbol{d}_m^\top$. Note that when $z_{i_k m}^{(k)} = 0$, $\Sigma_{s_{i_k m}^{(k)}} = \gamma_s^{-1}$, and $\mu_{s_{i_k m}^{(k)}} = 0$.

Updating $\pi^{(k)}$: For the update of $\pi_m^{(k)}$, $m = 1, 2, \cdots, M,\ k = 1, 2, \cdots, K,\ p(\pi_m^{(k)} |) \propto \mathrm{Beta}(\pi_m^{(k)} | a_0, b_0) \prod_{i_k = 1}^{n_k} \mathrm{Bern}(z_{i_k m}^{(k)} | \pi_m^{(k)})$, which leads to the simple update:

$$\pi_m^{(k)} \sim \mathrm{Beta}(a_0 + \sum_{i_k = 1}^{n_k} z_{i_k m}^{(k)}, b_0 + n_k - \sum_{i_k = 1}^{n_k} z_{i_k m}^{(k)})$$

Updating $\boldsymbol{D}^{(k)}$: For the update of layer-two dictionary matrix $\boldsymbol{D}^{(k)}$, we have for $\boldsymbol{d}_m^{(k)}$, $m = 1, 2, \cdots, M,\ k = 1, 2, \cdots, K$,

$$p(\boldsymbol{d}_m^{(k)} | -) \propto \prod_{i_k = 1}^{n_k} \mathcal{N}or(\boldsymbol{u}_{i_k}^{(k)} | (\boldsymbol{s}_{i_k}^{(k)} \odot \boldsymbol{z}_{i_k}^{(k)}) D^{(k)} + \boldsymbol{F}_{i_k}^{(k)} \boldsymbol{\beta}^{(k)}, \gamma_\epsilon^{-1} \boldsymbol{I}_R) \times \mathcal{N}or(\boldsymbol{d}_m^{(k)} | \boldsymbol{0}, R^{-1} \boldsymbol{I}_R) \tag{7}$$

Therefore $\boldsymbol{d}_m^{(k)} \sim \mathcal{N}or(\boldsymbol{d}_m^{(k)} | \mu_{\boldsymbol{d}_m^{(k)}}, \Sigma_{\boldsymbol{d}_m^{(k)}})$, where $\boldsymbol{d}_m^{(k)} \sim \mathcal{N}or(\boldsymbol{d}_m^{(k)} | \mu_{\boldsymbol{d}_m^{(k)}}, \Sigma_{\boldsymbol{d}_m^{(k)}})$, $\Sigma_{\boldsymbol{d}_m^{(k)}} = (R + \gamma_\epsilon \sum_{i_k = 1}^{n_k} {s_{i_k m}^{(k)}}^2 {z_{i_k m}^{(k)}}^2)^{-1} \boldsymbol{I}$, $\mu_{\boldsymbol{d}_m^{(k)}} = \gamma_\epsilon (\sum_{i_k = 1}^{n_k} s_{i_k m}^{(k)} z_{i_k m}^{(k)} \Delta_{i_k, -m}^{(k)}) \Sigma_{\boldsymbol{d}_m^{(k)}}$, and $\Delta_{i_k, -m}^{(k)} = \boldsymbol{u}_{i_k}^{(k)} - T_{i_k}^{(k)} F^{(k)} - (\boldsymbol{s}_{i_k}^{(k)} \odot \boldsymbol{z}_{i_k}^{(k)}) D^{(k)} + (s_{i_k m}^{(k)} \odot z_{i_k m}^{(k)}) \boldsymbol{d}_m^{(k)}$.

Updating $\boldsymbol{\beta}^{(k)}$: For the update of mode k regression coefficient matrix $\boldsymbol{\beta}^{(k)} \in \mathbb{R}^{L \times R}$, we have for $\boldsymbol{\beta}_l^{(k)}$, $l = 1, 2, \cdots, L,\ k = 1, 2, \cdots, K$,

$$p(\boldsymbol{\beta}_l^{(k)} | -) \propto \prod_{i_k = 1}^{n_k} \mathcal{N}or(\boldsymbol{u}_{i_k}^{(k)} | (\boldsymbol{s}_{i_k}^{(k)} \odot \boldsymbol{z}_{i_k}^{(k)}) D^{(k)} + \boldsymbol{F}_{i_k}^{(k)} \boldsymbol{\beta}^{(k)}, \gamma_\epsilon^{-1} \boldsymbol{I}_R) \times \mathcal{N}or(\boldsymbol{\beta}_l^{(k)} | 0, \rho^2 \boldsymbol{I}_R) \tag{8}$$

Therefore $\boldsymbol{\beta}_l^{(k)} \sim \mathcal{N}or(\boldsymbol{\beta}_l^{(k)} | \mu_{\boldsymbol{\beta}_l^{(k)}}, \Sigma_{\boldsymbol{\beta}_l^{(k)}})$, where $\Sigma_{\boldsymbol{\beta}_l^{(k)}} = (R + \gamma_\epsilon \sum_{i_k = 1}^{n_k} {\boldsymbol{F}_{i_k l}^{(k)}}^2)^{-1} \boldsymbol{I}_R$, $\mu_{\boldsymbol{\beta}_l^{(k)}} = \gamma_\epsilon (\sum_{i_k = 1}^{n_k} \boldsymbol{F}_{i_k l}^{(k)} \Lambda_{i_k, -l}^{(k)}) \Sigma_{\boldsymbol{\beta}_l^{(k)}}$, and $\Lambda_{i_k, -l}^{(k)} = \boldsymbol{u}_{i_k}^{(k)} - (\boldsymbol{s}_{i_k}^{(k)} \odot \boldsymbol{z}_{i_k}^{(k)}) D^{(k)} - \boldsymbol{F}_{i_k}^{(k)} \boldsymbol{\beta}^{(k)} + \boldsymbol{F}_{i_k l}^{(k)} \boldsymbol{\beta}_l^{(k)}$.

Table 1: AUC Scores: Binary Tensor Completion with Side-Information

	Lazega-Lawyers		**UCLAF**	
	95% missing	**50% missing**	**95% missing**	**50% missing**
Bayesian CP	0.6037 (±0.0081)	0.8207 (±0.0059)	0.8412 (±0.0412)	0.9113 (±0.0134)
BCPFN	**0.6414 (±0.0203)**	**0.8336 (±0.0101)**	**0.8855 (±0.0312)**	**0.9407 (±0.0156)**

Related Work

The performance of tensor decomposition methods tends to deteriorate as the amount of missing data in the tensor becomes large (Ermiş, Acar, and Cemgil 2013; Acar, Kolda, and Dunlavy 2011). To improve the performance of tensor decomposition and completion methods in high missing data regimes, methods based on coupled matrix-tensor factorization (CMTF) (Ermiş, Acar, and Cemgil 2013; Acar, Kolda, and Dunlavy 2011; Simsekli et al. 2013) have been proposed. The CMTF, originally inspired by collective matrix factorization (Singh and Gordon 2008), assumes that the some of the tensor modes have associated feature matrices, and the tensor and the mode k feature matrix both share the mode k factor matrix. These methods however lack a rigorous generative model for the data and therefore cannot model more general types of data such as binary-valued tensors, and do not have a Bayesian formulation unlike our framework. Also, these methods cannot leverage networks assocaited with the tensor modes. Due to its inability in dealing with binary tensors, the CMTF framework cannot be applied for problems such as modeling multirelational data with side-information.

The work in (Narita et al. 2012) uses network information via a graph Laplacian approach. However there are several key differences from our work: (i) it is however limited to real-valued tensors; (ii) the the graph is assumed to be *fully observed*; (iii) the computational complexity such a model based on graph regularization scales quadratically in the number of entities in each mode.

In contrast to the above methods, our framework is general enough to allow modeling both real-valued as well as binary tensors, and at the same time, can incorporate one or more feature matrices and/or adjacency network between the entites of tensor modes. Moreover, unlike these methods, our method does not require the rank of tensor decomposition to be specified which our model infers in a manner similar to (Rai et al. 2014).

Experiments

We perform experiments on both real-valued and binary-valued tensor data, each having features and/or network associated with one or more tensor modes. We are especially interested in regimes when the main tensor has a significantly large amount of missing data. For our experiments, we use the following datasets:

(i) **Lazega-Lawyers Data:** This is a multirelational dataset consisting of a 71 × 71 × 3 binary tensor consisting of 3 type of relationships (work, advice, friendship) between 71 lawyers in some New England law firms (Lazega 2001). The dataset also consists of 7 real-valued features for each lawyer, such as gender, location, age, years employed, etc).

(ii) **UCLAF Data:** This is sparse binary 164 × 168 × 5 tensor (Zheng et al. 2010) containing data from 164 users, visiting a subset of 168 locations, and performing a subset of 5 activities. For this data, we also have two feature matrices containing information about location features and user-location preferences, respectively.

(iii) **EEG data:** This data consists of a real-valued tensor of size 15 × 16 × 560, along with the network between the 560 entities in mode 3 constructed using their binary valued labels (an edge exists if two entites share the same label).

We compare our method with Bayesian CP decomposition (Rai et al. 2014), a recently proposed state-of-the-art method for Bayesian CP decomposition, except that it cannot leverage side-information, and with coupled matrix-tensor factorization (CMTF) (Ermiş, Acar, and Cemgil 2013; Acar, Kolda, and Dunlavy 2011) which can leverage side-information. Note that CMTF cannot deal with binary tensors and/or binary networks, so we could not compare with these on binary tensors, but we provide a comparison with this method on the real-valued EEG data.

The goal of our experiments is to demonstrate how our model leverages the side-information and leads to improved tensor decomposition especially in the cases where the tensor has a significantly high fraction of missing data, where Bayesian CP methods such as (Xiong et al. 2010; Rai et al. 2014), which even though being flexible model for handling both binary and real tensors, could break down when the amount of missing data becomes too high.

Each experiment is run 10 times with different splits of observed and missing data. We report both mean and the standard deviations. We run MCMC for 1000 iterations, with 600 burnin iterations, and collect samples every five samples, after the burnin phase. We compute the posterior averages using the samples collected after burnin. For Bayesian CP and our method, the rank R of the tensor need not be set (inferred by the MGP prior). For our method, the parameter M was simply set to R (though it can be inferred using priors such as the Indian Buffet Process). For CMTF, we tried a range of values for the rank and report the best results.

Binary Tensor Completion with Side-Information

In our first experiment, we compare our model with Bayesian CP decomposition (Rai et al. 2014) on the task of binary tensor completion. For this task, we evaluate both methods on the Lazega-Lawyers data and the UCALF data. To simulate the setting of significantly high fraction of missing data, we tried two settings: 95% missing and 50% missing. We report the Area under the Receiver Operating Characteristic (AUC) curve for the task of predicting the hidden entries in the tensor. As shown in Table 1, our method

Table 2: EEG Data: Comparision with Coupled Matrix-Tensor Factorization

	% Mean-Squared Error (lower better)		**% Variance Explained (higher better)**	
	50% missing	**75% missing**	**50% missing**	**75% missing**
CMTF	0.1515 (±0.0212)	0.8267 (±0.0682)	84.58% (±0.96)	17.49% (±3.12)
Bayesian CP	0.1662 (±0.0332)	0.2657 (±0.0241)	85.12% (±1.06)	73.19% (±2.64)
BCPFN	**0.1420 (±0.0102)**	**0.2289 (±0.0211)**	**85.79% (±1.01)**	**77.19% (±2.35)**

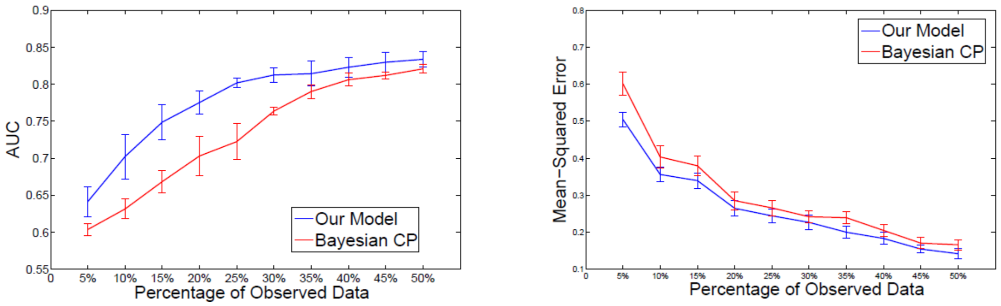

Figure 2: Tensor completion for varying amount of missing data: Left: Lazega-Lawyers Data (AUC Scores). Right: EEG Data (MSE)

significantly outperforms Bayesian CP especially when the amount of missing data is very high (95%).

Real-Valued Tensor Completion with Side-Information

For the EEG data, which is a real-valued tensor, we compare our model BCPCN with Coupled Matrix Tensor Factorization (Ermiş, Acar, and Cemgil 2013; Acar, Kolda, and Dunlavy 2011) (CMTF) and with Bayesian CP. For this data, the third mode has binary labels which we provide as feature for CMTF. We experiment with two missing data settings: 50% missing and 75% missing. The results are shown in Table 2 where we show both mean-squared-error and the percentage variance explained. In addition to performing better than Bayesian CP, our model does better than CMTF (which does take into account the side-information) for both these settings. Moreover, the difference between our model and CMTF becomes more pronounced for the 75% missing case, which suggests that our Bayesian framework is more robust to missing data even in the regimes when a significantly high fraction of data is missing.

Finally, we experiment with randomly held-out fractions of data in the main tensor and predict it using the observed data. For the Lazega Lawyers data and the EEG data, we vary the amount of observed data in the main tensor from 5% to 50% with increments of 5%. We run each method for each of these setting (each experiment is further repeated 10 times) and compare our model with Bayesian CP which cannot leverage side-information. As our experiment in Figure 2 shows, in the cases when the amount of missing data is very high, our model performs considerably better than Bayesian CP, which shows the benefit of our model for being able to leverage side-information in an effective manner.

Conclusion and Future Work

We have presented a probabilistic, fully Bayesian tensor decomposition method for sparse tensors, leveraging side-information in form of the features and/or network of the entities in each tensor mode. Our method is fairly general and can be applied for both real-valued as well as binary tensor data. Moreover, diverse types of side-information can be naturally incorporated in our framework.

The two-layer tensor decomposition approach presented here can in fact be further generalized by introducing nonlinearities on the first layer factors, akin to deep learning methods (Bengio 2009; Bengio, Courville, and Vincent 2013), e.g, using a sigmoid operation, prior to the second layer decomposition. To the best of our knowledge, deep learning methods have not yet been developed for tensor decomposition and we leave this generalization to future work.

Our framework is not limited to only tensor decomposition but can also be used for other problems such as *multiway* generalizations of overlapping clustering (Latouche et al. 2011) and multimodal hashing (Zhen and Yeung 2012). Currently we use MCMC based on closed-form Gibbs sampling updates to perform inference in this model. In terms of computation, in spite of being considerably more general, our method is only slightly more expensive than Bayesian CP (Rai et al. 2014), this extra cost is due to the additional layer of variables (second-layer factorization) to be sampled. Scaling up the inference for our model would be another future avenue of work.

Appendix

Updating MGP variables: We sample $\delta_r, 1 \leq r \leq R$ as $\delta_r \sim \text{Ga}(a_r + \frac{1}{2}(R - r + 1), 1 + \frac{1}{2}\sum_{h=r}^{R} \lambda_h^2 \prod_{l=1,l\neq r}^{h} \delta_l)$. For sampling $\lambda_r, 1 \leq r \leq R$, when tensor $\mathcal{Y}$ is real-valued, we have $\lambda_r \sim \mathcal{N}(\hat{\mu}_r, \hat{\tau}_r^{-1})$, where $\hat{\tau}_r = \tau_r + \tau_e \sum_{\boldsymbol{i}} a_{\boldsymbol{i}}^{r\,2}$

and $\hat{\mu}_r = \hat{\tau}_r^{-1} \tau_e \sum_{\boldsymbol{i}} a_{\boldsymbol{i}}^r (y_{\boldsymbol{i}} - b_{\boldsymbol{i}}^r)$. For the case when $\mathcal{Y}$ is binary, we have $\lambda_r = \frac{1}{a_{\boldsymbol{i}}^r} x_{\boldsymbol{i}} - \frac{b_{\boldsymbol{i}}^r}{a_{\boldsymbol{i}}^r}$. First the augment random variable $\phi_{\boldsymbol{i}}$ is drawn independently from the Pólya-Gamma distribution: $\phi_{\boldsymbol{i}} \overset{\text{ind}}{\sim} \text{PG}(1, x_{\boldsymbol{i}})$ where $\text{PG}(\cdot, \cdot)$ represents the Pólya-Gamma distribution. So $x_{\boldsymbol{i}} \sim \mathcal{N}(\frac{y_{\boldsymbol{i}} - 0.5}{\phi_{\boldsymbol{i}}}, {\phi_{\boldsymbol{i}}}^{-1})$.Then $\lambda_r, 1 \leq r \leq R$ is drawn from Gaussian. We use the Pólya Gamma sampling: $\lambda_r \sim \mathcal{N}(\hat{\mu}_r, \hat{\tau}_r^{-1})$ where $\hat{\tau}_r = \tau_r + \sum_{\boldsymbol{i}} {a_{\boldsymbol{i}}^r}^2 \phi_{\boldsymbol{i}}$ and $\hat{\mu}_r = \hat{\tau}_r^{-1} \sum_{\boldsymbol{i}} a_{\boldsymbol{i}}^r (y_{\boldsymbol{i}} - 0.5 - \phi_{\boldsymbol{i}} b_{\boldsymbol{i}}^r)$.

Acknowledgements

The research reported here was funded in part by ARO, DARPA, DOE, NGA and ONR.

References

Acar, E., and Yener, B. 2009. Unsupervised multiway data analysis: A literature survey. *Knowledge and Data Engineering, IEEE Transactions on*.

Acar, E.; Kolda, T. G.; and Dunlavy, D. M. 2011. All-at-once optimization for coupled matrix and tensor factorizations. *arXiv preprint arXiv:1105.3422*.

Agarwal, D., and Chen, B.-C. 2009. Regression-based Latent Factor Models. In *KDD*.

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35(8):1798–1828.

Bengio, Y. 2009. Learning deep architectures for ai. *Foundations and trends® in Machine Learning* 2(1):1–127.

Bhattacharya, A., and Dunson, D. 2011. Sparse bayesian infinite factor models. *Biometrika*.

Bronstein, M. M.; Bronstein, A. M.; Michel, F.; and Paragios, N. 2010. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *CVPR*.

Chu, W., and Ghahramani, Z. 2009. Probabilistic models for incomplete multi-dimensional arrays. In *AISTATS*.

Cichocki, A. 2013. Tensor decompositions: A new concept in brain data analysis? *arXiv preprint arXiv:1305.0395*.

Ermiş, B.; Acar, E.; and Cemgil, A. T. 2013. Link prediction in heterogeneous data via generalized coupled tensor factorization. *Data Mining and Knowledge Discovery*.

Kolda, T. G., and Bader, B. W. 2009. Tensor decompositions and applications. *SIAM review*.

Latouche, P.; Birmelé, E.; Ambroise, C.; et al. 2011. Overlapping stochastic block models with application to the french political blogosphere. *The Annals of Applied Statistics* 5.

Lazega, E. 2001. *The collegial phenomenon: The social mechanisms of cooperation among peers in a corporate law partnership*. Oxford University Press on Demand.

Miller, K.; Jordan, M. I.; and Griffiths, T. L. 2009. Nonparametric latent feature models for link prediction. In *NIPS*.

Narita, A.; Hayashi, K.; Tomioka, R.; and Kashima, H. 2012. Tensor factorization using auxiliary information. *Data Mining and Knowledge Discovery* 25.

Nickel, M.; Tresp, V.; and Kriegel, H. 2011. A three-way model for collective learning on multi-relational data. In *ICML*.

Paisley, J., and Carin, L. 2009. Nonparametric Factor Analysis with Beta Process Priors. In *ICML*.

Polson, N.; Scott, J.; and Windle, J. 2012. Bayesian inference for logistic models using Polya-Gamma latent variables, http://arxiv.org/abs/1205.0310.

Rai, P.; Wang, Y.; Guo, S.; Chan, G.; Dunson, D.; and Carin, L. 2014. Scalable bayesian low-rank decomposition of incomplete multiway tensors. In *ICML*.

Simsekli, U.; Ermis, B.; Cemgil, A. T.; and Acar, E. 2013. Optimal weight learning for coupled tensor factorization with mixed divergences. In *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European*, 1–5. IEEE.

Singh, A. P., and Gordon, G. J. 2008. Relational learning via collective matrix factorization. In *KDD*.

Xiong, L.; Chen, X.; Huang, T.; Schneider, J. G.; and Carbonell, J. G. 2010. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SDM*.

Xu, Z.; Yan, F.; and Qi, Y. 2013. Bayesian nonparametric models for multiway data analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yin, D.; Guo, S.; Chidlovskii, B.; Davison, B. D.; Archambeau, C.; and Bouchard, G. 2013. Connecting comments and tags: improved modeling of social tagging systems. In *WSDM*.

Zhen, Y., and Yeung, D.-Y. Co-regularized hashing for multimodal data. In *NIPS*.

Zhen, Y., and Yeung, D.-Y. 2012. A probabilistic model for multimodal hash function learning. In *KDD*.

Zheng, V. W.; Cao, B.; Zheng, Y.; Xie, X.; and Yang, Q. 2010. Collaborative filtering meets mobile recommendation: A user-centered approach. In *AAAI*.