Scalable Bayesian Matrix and Tensor Factorization for Discrete Data

by

Changwei Hu

Department of Electrical and Computer Engineering
Duke University

Date: __________________________

Approved:

______________________________
Lawrence Carin, Supervisor

______________________________
Piyush Rai

______________________________
Robert Calderbank

______________________________
Galen Reeves

______________________________
Matthew Harding

______________________________
Katherine Heller

Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Electrical and Computer Engineering in the Graduate School of Duke University
2017
Abstract

Scalable Bayesian Matrix and Tensor Factorization for Discrete Data

by

Changwei Hu

Department of Electrical and Computer Engineering
Duke University

Date: __________________________

Approved:

______________________________
Lawrence Carin, Supervisor

______________________________
Piyush Rai

______________________________
Robert Calderbank

______________________________
Galen Reeves

______________________________
Matthew Harding

______________________________
Katherine Heller

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Electrical and Computer Engineering in the Graduate School of Duke University 2017
Matrix and tensor factorization methods decompose the observed matrix and tensor data into a set of factor matrices. They provide a useful way to extract latent factors or features from complex data, and also to predict missing data. Matrix and tensor factorization has drawn significant attention in a wide variety of applications, such as topic modeling, recommender systems, and learning from social network and knowledge bases. However, developing factorization methods for massive and sparse observations remains a challenge, especially when the data are binary or count-valued (which is true of most real-world data). In this thesis, we present a set of scalable Bayesian factorization models for low rank approximation of massive matrices or tensors with binary and count-valued observations. The proposed models enjoy the following properties: (1) The inference complexity scales linearly in the number of non-zeros in the data; (2) The side-information along a certain dimension, such as pairwise relationships (e.g., an adjacency network) between entities, can be easily leveraged to handle issues such as data sparsity, and the cold-start problem; (3) The proposed models have full local conjugacy, leading to simple, closed-form batch inference as well as online inference; (4) In contrast to many existing matrix and tensor factorization methods, in which factor matrices are usually assumed to be real-valued, we assume non-negativity on factor matrices. The non-negative factor matrices in our model provide easy interpretability; (5) For tensor factorization, the number of “topics”, or in other words, the rank of tensor, can be inferred from the
data. In this thesis, we evaluate the proposed models on a variety of real-world data sets, from diverse domains, such as analyzing scholarly text data, political science data, large-scale market transaction data and knowledge-graphs.
## Contents

Abstract iv  
List of Tables x  
List of Figures xii  
Acknowledgements xiv  

1 Introduction 1  
  1.1 Non-negative Matrix Factorization for Binary and Count-valued Data with Hierarchical Side-information 2  
  1.2 Scalable Bayesian Non-negative Tensor Factorization for Massive Count Data 3  
  1.3 Zero-Truncated Poisson Tensor Factorization for Massive Binary Data 4  
  1.4 Learning from Multi-relational Data using Binary Tensor Factorization 6  

2 Non-negative Matrix Factorization for Discrete Data with Hierarchical Side-Information 8  
  2.1 Introduction 8  
  2.2 The Model 10  
    2.2.1 Background 10  
    2.2.2 Leveraging Multi-level Side-Information 11  
    2.2.3 Learning Multi-level Embeddings 12  
    2.2.4 Modeling Binary $X$ 13  
  2.3 Inference via Gibbs Sampling 14
List of Tables

2.1 Loglikelihood comparison between PFA and PFA-SSI for State of the Union (STOU), 20 newsgroup (20 News), Scholar, and NIPS data sets. 10% data was held out as testing data, and 90% used as training. Results are averaged over 10 random splits of training and test. ... 21

2.2 Loglikelihood, AUC and AUC-PR comparison between PFA and PFA-SSI for Cora and CiteSeer datasets. 10% data was held out as testing data, and 90% used as training. Results are averaged over 10 random splits of training and test. ... 21

2.3 Most prominent topic for 12 groups in 20 newsgroup data ... 22

2.4 Most prominent topic for each supergroup in 20 newsgroup data ... 24

2.5 Two of the most prominent topics (for considered time-period of 1988-2003) for three of the authors in NIPS data ... 24

2.6 Five most similar authors (for considered time-period of 1988-2003) for three of the authors in NIPS data ... 24

3.1 Loglikelihood and MAE comparison for different methods (the two baselines, our model with batch inference, and our model with online inference) on four datasets. Note: lranTD gave out-of-memory error on publications and food transactions data sets so we are unable to report its results on these data sets. We also only report the MAE for lranTD, and not the log-likelihood, because it uses a Gaussian likelihood model for the data. ... 43

3.2 Most probable words in topics related to optics, genomics, machine learning/signal processing(ML/SP) and statistics (Stats), and top ranked venues in ML/SP community. ... 44

3.3 Three of the store factors inferred from the transaction data (top-5 stores shown for each) ... 49
4.1 Tensor completion accuracies in terms of AUC-ROC scores. Results are averaged over 10 splits of training and test data. Note: (1) Bayesian CP was infeasible to run on the Scholars and Facebook data; (2) Due to the lack of publicly available code for Quad-App and PQ-QuadApp, we only report its results on Kinship, UMLS, and MovieLens data (results taken from (Ermis and Bouchard, 2014)).

4.2 For the Scholars data, the most probable words in topics related to evolutionary biology (Evo Bio), medical imaging (Med Imag), machine learning/signal processing (ML/SP) and oncology, and top ranked venues in ML/SP.

4.3 Cold-start setting

5.1 Statistics of FB-15K, NELL-50K, and WN-100K

5.2 Most prominent entities in topics inferred from UMLS

5.3 Most prominent relations in topics inferred from UMLS

5.4 AUC comparison for multi-relational Data

5.5 Computational time comparison

5.6 Most prominent entities in topics inferred for FB-15K, WN-100K and NELL-50K

5.7 Most prominent relations in topics inferred for FB-15K

5.8 AUC-PR for knowledge bases

5.9 AUC and AUC-PR (A-PR below) comparison between BPBFM-1 and BPBFM-2
List of Figures

2.1 Two examples of the type of side-information that our proposed framework can leverage. Left: Side-information specified in form of a multi-layer hierarchy with bipartite connections between nodes in adjacent layers. Right: Side-information specified in form of an ontology over known labels. In this case, each document is associated with a single categorical label and these labels are the leaf nodes of a label ontology. 9

2.2 20 newsgroups data. Left Figure: Inferred similarities between the level-one nodes (i.e., between the 20 groups) in the side-information. Right Figure: Inferred similarities between the level-one and the level-two nodes (i.e., 20 groups and 7 supergroups) in the side-information. The numbers are indices for groups, and numbers with same color indicate that the corresponding groups are associated with the same supergroup. 23

2.3 State of the union data. Left Figure: Inferred similarities between the presidents in the embedding space. The numbers before each president are labels for parties. 1: Independent; 2: Federalist; 3: Democratic-Republican; 4: Democrat; 5: Whig; 6: Republican. In the legend, the names of all presidents from the same party are shown in the same color. Right Figure: Inferred similarities between presidents (level-one nodes) and parties (level-two nodes). 23

3.1 Distribution over inferred ranks for synthetic data (left), and λ inferred using 80% training data (right). 42

3.2 Histogram of affiliations for top 20 authors in factors related to machine learning/signal processing (top left) and statistics (top right), optics (bottom left), and genomics (bottom right). 45

3.3 Country factors (top row) and time factors (bottom row) for Julian Assange asylum in Ecuador (left column) and 2012 Benghazi attack (right column). 47
3.4 Distributions over items for three factors (each factor corresponds to a cluster). ........................................ 49
3.5 Time vs heldout log likelihoods with various methods on transactions data ........................................ 51
3.6 Timing comparison of various methods on Scholars data ................. 51
4.1 Binary tensor with an associated binary network between objects in mode-1 of the tensor (in general, network for other modes may also be available). In the “cold-start” setting as shown above, data along some of the tensor dimensions will be completely missing ............ 57
4.2 Running time (log-scale) comparison of various methods on Kinship (top left), UMLS (top right), Scholars (bottom left), and Facebook (bottom right) datasets. ....................................... 72
4.3 Histogram of the department-affiliations for the top 20 authors in factors related to evolutionary biology (top left), medical imaging (top right), machine learning/signal processing (bottom left) and oncology (bottom right). ..................................................... 74
4.4 Histogram of the department-affiliations of the top 15 held-out authors associated with the factors of medical imaging (top) and oncology (bottom). The left column is obtained using no co-authorship information, and the right column is obtained using co-authorship information. ..................................................... 76
5.1 Left: Model-1 with each relation \( r \) having its own independent parameter matrix \( \Lambda^r \). Right: Model-2 with parameter sharing across relations via a set of basis matrices .......................... 79
5.2 The basic setup of the bilinear latent factor model for multi-relational data ..................................................... 80
5.3 Parameter sharing across relations for Model-2 ....................... 83
5.4 Similarities between relations in Wordnet ............................. 95
Acknowledgements

First and foremost I would like to express my sincere gratitude to my advisor, Dr. Lawrence Carin, for his academic advising and support. During the past few years, Larry has offered significant help in my research, by allowing me the freedom to explore diversified machine learning topics, and providing me with the opportunities to collaborate with excellent researchers in both academia and industry. His rigorous attitude, persistent passion towards research, and diligent working will carry a lifelong influence on my future careers.

In addition, I want to especially acknowledge my co-advisor, Dr. Piyush Rai, with whom I have worked closely for long time. His expertise in machine learning and insightful opinions were extremely helpful in my PhD research. I would also like to express my deep thanks to the dissertation committee members, Dr. Robert Calderbank, Dr. Galen Reeves, and Dr. Katherine Heller, for their time and comments on this thesis. I want to thank Dr. Matthew Harding for offering me the opportunity to understand some interesting problems in economics, and develop machine learning methods for analyzing food transaction data.

Then I would like to thank my labmates at Duke, Xin Yuan, Ricardo Henao, Changyou Chen, Liming Wang, Jianbo Yang, Qinliang Su, Bo Chen, Mingyuan Zhou, Lingbo Li, Miao Liu, Yingjian Wang, Zhengming Xing, Shaobo Han, David Carlson, Kyle Ulrich, Wenzhao Lian, Jiaji Huang, Yunchen Pu, Zhe Gan, Zhao Song and Chunyuan Li, for their help in and outside research. Working and being friends...
with them make my life at Duke an enjoyable experience.

Finally, I want to thank my parents, Kesu Hu and Chunying Xu, and my wife, Ye Hu, for their support and love.
Matrix and tensor factorization offers a useful way to learn latent factors from massive and complex data. Although several matrix and tensor factorization methods have been proposed in recent years (Kolda and Bader, 2009; Chi and Kolda, 2012), and there has been a significant recent interest on developing scalable factorization methods (Kang et al., 2012; Inah et al., 2015; Papalexakis et al., 2012; Beutel et al., 2014), most of these methods treat data as real-valued while many real-word data are binary or count-valued. Motivated by the prevalence of binary and count-valued observations, this dissertation presents a scalable, fully Bayesian non-negative matrix and tensor factorization framework which can handle massive binary and count-valued data. In addition, the proposed framework can leverage side-information along a certain dimension easily, making it capable of handling issues such as data sparsity and the cold-start problem, etc.

In the following chapters, we first introduce a factorization method for binary and count-valued 2D matrix data, and then generalize it to model observations with multiple dimensions (tensors).
1.1 Non-negative Matrix Factorization for Binary and Count-valued Data with Hierarchical Side-information

Non-negative matrix factorization (NMF) is a fundamental problem in many machine learning applications, such as text modeling (Zhou et al., 2012b), social network modeling (Yang and Leskovec, 2013), recommender systems (Gopalan et al., 2015), and so on. By assuming non-negativity on factor matrices, NMF can provide easy-to-interpret results.

In Chapter 2 (Hu et al., 2016a), we present a probabilistic framework for efficient non-negative matrix factorization of discrete (count or binary) data. In addition to the matrix being factorized, we also consider the scenario when side-information is available along the rows and/or columns. The side-information is given as a multi-level structure, taxonomy or ontology, with nodes at each level being categorical-valued observations. For example, when modeling documents with a two-level side-information (documents being at level-zero), level-one may represent (one or more) authors associated with each document and level-two may represent affiliations of each author. The model easily generalizes to more than two levels (or taxonomy/ontology of arbitrary depth), and it can learn embeddings of entities present at each level in the data or side-information hierarchy (e.g., documents, authors, affiliations, in the previous example), with appropriate sharing of information across levels. The model also enjoys full local conjugacy, facilitating efficient Gibbs sampling for model inference. Inference cost scales in the number of non-zero entries in the data matrix, which is especially appealing for real-world massive but sparse matrices.
1.2 Scalable Bayesian Non-negative Tensor Factorization for Massive Count Data

In this dissertation, we further extend matrix factorization to higher order and discuss tensor factorization. We focus on CANDECOMP-PARAFAC (CP) tensor factorization, which approximates the observed tensor as a weighted summation of several rank-one tensors. CP tensor factorization methods (Kolda and Bader, 2009) are useful to learn latent factors from complex multiway data. These methods decompose the original tensor data into a set of factor matrices (one for each mode or way of the tensor), which can be used as a latent feature representation for the objects in each of the tensor mode, and can be used for other tasks, such as tensor completion.

Among tensor factorization methods, probabilistic approaches (Chu and Ghahramani, 2009; Xu et al., 2013; Rai et al., 2014, 2015b) are especially appealing because of a proper generative model of the data, which allows modeling different data types and handling missing data in a natural way.

Real-world tensor data are often count-valued (or binary) (Nickel et al., 2011; Chi and Kolda, 2012). For example, from a database of research publications, one may construct a three-way (AUTHORS × WORDS × VENUES) count-valued tensor, where the three dimensions could be authors, words, and publication venues and each entry of the tensor denotes the number of time an author used a specific word at a specific venue. Tensor factorization on this multiway data can be used for topic modeling on such a publications corpus (the latent factors would correspond to topics). Another application could be in recommender systems; having learned the latent factors of authors and venues, one can use these factors for author-author recommendation (for potential co-authors) or author-venue recommendation (recommending the most appropriate venues for a given author).

Although there has been a significant recent interest on developing scalable tensor
factorization methods (Kang et al., 2012; Inah et al., 2015; Papalexakis et al., 2012; Beutel et al., 2014), most of these methods treat data as real-valued, and are therefore inappropriate for handling count-valued data.

Motivated by the prevalence of such count-valued tensors, we propose a probabilistic tensor factorization framework in Chapter 3 (Hu et al., 2015a), which can handle count-valued tensors, while being scalable for massive tensor data. Our starting point will be a conjugate, fully Bayesian model for count-valued observations, for which we develop efficient inference methods (both batch and online inference). Our generative model can handle overdispersed counts as well as infer the rank of the decomposition. Moreover, leveraging a reparameterization of the Poisson distribution as a multinomial facilitates conjugacy in the model and enables simple Gibbs sampling and variational Bayes (VB) inference updates. We also develop a set of online inference algorithms that allow scaling up the model to massive tensors. We apply our framework on diverse real-world applications, such as analyzing a political science database, multiway topic modeling on a scientific publications database, and analyzing a massive household transactions data set.

1.3 Zero-Truncated Poisson Tensor Factorization for Massive Binary Data

Many machine learning applications, such as recommender systems and community detection of social networks, involve tensor data with binary values. For example, a multirelational social network (Nickel et al., 2011) can be described as a three-way binary tensor with two modes denoting people and the third mode denoting the types of relationships.

Usually, real-world binary tensors are massive (each dimension can be very large) but extremely sparse (very few ones in the tensor). For example, in a recommender system, each positive example (e.g., an item selected a set) implicitly creates several
negative examples (items \textit{not} chosen). Likewise, in a knowledge base, the validity of one relation automatically implies invalidity of several other relations. In all these settings, the number of negative examples greatly overwhelms the number of positive examples.

Unfortunately, binary tensor factorization methods (Nickel et al., 2011; Xu et al., 2013; Rai et al., 2014), based on probit or logistic likelihood, scale poorly for massive binary tensors because these require evaluating the likelihood/loss-function on \textit{both} ones as well as zeros in the tensor. One possibility is to use heuristics such as \textit{undersampling} the zeros, but such heuristics usually result in less accurate solutions. Another alternative is to use the \textit{squared loss} (Hidasi and Tikk, 2012; Nickel et al., 2012) as the model-fit criterion, which facilitates linear scalability in the number of ones in the tensor. However, such an approach can often lead to suboptimal results (Ermis and Bouchard, 2014) in practice.

In Chapter 4 (Hu et al., 2015b), we present a scalable Bayesian model for low-rank factorization of massive tensors with binary observations. The proposed model has the following key properties: (1) in contrast to the models based on the logistic or probit likelihood, using a zero-truncated Poisson likelihood for binary data allows our model to scale up in the number of \textit{ones} in the tensor, which is especially appealing for massive but sparse binary tensors; (2) side-information in form of binary pairwise relationships (e.g., an adjacency network) between objects in any tensor mode can also be leveraged, which can be especially useful in “cold-start” settings (Ermiş et al., 2013; Narita et al., 2012); and (3) the model admits simple Bayesian inference via batch, as well as \textit{online} MCMC; the latter allows scaling up even for \textit{dense} binary data (i.e., when the number of ones in the tensor/network is also massive). In addition, non-negative factor matrices in our model provide easy interpretability, and the tensor rank can be inferred from the data. We evaluate our model on several large-scale real-world binary tensors, achieving excellent computational scalability,
and also demonstrate its usefulness in leveraging side-information provided in form of mode-network(s).

1.4 Learning from Multi-relational Data using Binary Tensor Factorization

Learning from multi-relational data is ubiquitous in problems in areas such as social/biological network analysis (Goldenberg et al., 2010; Jenatton et al., 2012) and, more recently, in the modeling of large knowledge bases such as YAGO (Suchanek et al., 2007), NELL (Carlson et al., 2010), Freebase (Bollacker et al., 2008), Google Knowledge Vault project (Dong et al., 2014), etc. Data in these problems usually consist of a sparsely observed set of triplets of the form entity-relation-entity and can be represented as a three-way binary sparse tensor $\mathbf{Y}$.

Given such data, we may be interested in predicting the existence of the unknown links between entities (e.g., in social/biological networks) or predicting the validity of previously unseen facts (e.g., for knowledge base completion). Other examples of learning tasks include clustering of entities, or ranking of entities for a given entity and a relation (e.g., for answering queries from a database).

Commonly used methods for learning from such data include methods based on tensor decomposition (Nickel et al., 2011; Sutskever et al., 2009; Jenatton et al., 2012), and more generally, methods that learn embeddings of the entities and relations (Socher et al., 2013; Bordes et al., 2013; Wang et al., 2014b; Yang et al., 2014; Dong et al., 2014). These embeddings are learned by optimizing some objective that assigns a higher score to an observed (positive) triplet as compared to unobserved (assumed negative) triplets, where the score is a function of the embeddings; also see this comprehensive recent review article (Nickel et al., 2015). In Section 5.4 on Related Work, we discuss these and other methods in more detail.

In Chapter 5 (Hu et al., 2016b), we present a Bayesian binary tensor factorization
model for multi-relational data where the observations are of the form entity-relation-entity triplets (e.g., in multi-relational social/biological networks, knowledge bases, etc.). Our model defines each triplet as generated via a relation-specific bilinear function of the embeddings (which correspond to “topics”) of entities associated with that triplet. In contrast to the existing methods, a distinguishing aspect of the proposed model is its dependence solely on the observed (i.e., positive) triplets in the data. Consequently, learning scales in the number of observed triplets, which is usually much smaller than the number of all triplets (consisting of both positives and negatives), and training the model does not require artificially generated invalid (i.e., negative) triplets. The topic-based entity embeddings also facilitate interpretability (e.g., entities can easily be clustered based on topics). We further extend the model to allow sharing of statistical strength across multiple relations, which is particularly beneficial if the number of relations is very large and/or if the amount of data per relation is very small. We also develop simple-to-implement batch as well as online Gibbs sampling algorithms, based on a data-augmentation scheme. We apply our framework on tasks such as multi-relational link-prediction, and learning from large knowledge bases.
2.1 Introduction

Non-negative matrix factorization for discrete data is a fundamental problem in many applications, such as text modeling (Zhou et al., 2012b; Hu et al., 2014), social network modeling (Yang and Leskovec, 2013), recommender systems (Gopalan et al., 2015), and so on. Often, in addition to the matrix being factorized, there is side-information available along the rows and/or columns, that can be leveraged to handle issues such as data sparsity, the cold-start problem, etc. Several attempts have been made in the recent past (Agarwal and Chen, 2009; Kim et al., 2012; Gopalan et al., 2014; Chaney et al., 2015) to incorporate such side-information when it is given in flat-structured feature vectors/covariates along the rows and/or the columns of the data matrix. In many problems, however, the side-information can more naturally be specified in terms of a hierarchy, with each node in the hierarchy being a categorical-valued observation. See Fig. 2.1 for some examples where the side-information is in form of a hierarchy or ontology of categorical-valued observations.
Although data exhibiting such structure are prevalent in many applications, existing matrix factorization models cannot properly leverage such forms of side-information arranged in form of multiple layers.

**Figure 2.1**: Two examples of the type of side-information that our proposed framework can leverage, Left: Side-information specified in form of a multi-layer hierarchy with bipartite connections between nodes in adjacent layers. Right: Side-information specified in form of an ontology over known labels. In this case, each document is associated with a single categorical label and these labels are the leaf nodes of a label ontology.

We present a generative Bayesian framework that allows us to leverage such structural (e.g., specified hierarchically or via a taxonomy) side-information in the context of non-negative matrix factorization of discrete data. Moreover, the proposed framework can handle count as well as binary matrices in a unified manner. In addition to being useful for standard tasks such as matrix completion for count/binary data, our framework can also be used for topic modeling, while leveraging the available side-information. Another appealing aspect of our framework is that, in addition to learning the embeddings for the rows and columns of the data matrix, it can also learn embeddings of the nodes present in the structure that forms the side-information; e.g., for the two examples shown in Fig. 2.1, our model can learn the embeddings of documents and words, as well as learn the embeddings for the entities that constitute the side-information - authors and affiliations in Fig 2.1 (left) and each of the nodes in the label taxonomy in Fig 2.1 (right). These interpretable embeddings can
be useful in other tasks, such as clustering and classification, or for topic modeling at *multiple resolutions*, which significantly enhances the versatility and usefulness of our framework for applications beyond matrix factorization and completion.

Our framework also enjoys full local conjugacy which facilitates closed-form Gibbs sampling for all the model parameters. Moreover, inference in our model (for both count as well as binary matrix case) scales in the number of nonzeros in the data matrix, which makes it scale easily to massive but sparse matrices.

2.2 The Model

Here, we will present the model description assuming that the side-information is given with a hierarchy or ontology with two levels; the model can be easily modified to work with arbitrary number of levels.

We assume that we are given a data matrix $X$ of size $M \times N$, where each column of $X$ represents an object (e.g., a document). Fig. 2.1 shows two examples where the observations in $X$ are count-valued (e.g., word counts for documents). The case when the observations in $X$ are binary will be discussed subsequently in Sec. 2.2.4. The side-information for the objects is provided in form of a multi-level structures, such as a hierarchy (Fig. 2.1-left) or an ontology (Fig. 2.1-right).

2.2.1 Background

In the absence of any side-information, the counts matrix $X \in \mathbb{Z}^{M \times N}$ can be modeled via a Poisson factor analysis (PFA) model as $X \sim \text{Pois}(UV^T)$ where $U$ and $V$ are positive-valued matrices of size $M \times R$ and $N \times R$, respectively, and $R$ denotes the number of latent factors. This construction is also equivalent to assuming that each entry $x_{mn}$ in $X$ can be written as a sum of $R$ *latent counts* (Dunson and Herring, 2005):
\[ x_{mn} = \sum_{r=1}^{R} x_{mnr}, \quad x_{mnr} \sim \text{Pois}(u_{mr}v_{nr}) \] (2.1)

\[ \mathbf{u}_r \sim \text{Dir}(\alpha, \ldots, \alpha) \] (2.2)

\[ v_{nr} \sim \text{Ga}\left( g_r, \frac{q_r}{1 - q_r} \right) \] (2.3)

\[ g_r \sim \text{Ga}(c_0, 1/h_0) \] (2.4)

\[ q_r \sim \text{Beta}(c\epsilon, c(1 - \epsilon)) \] (2.5)

Note that each Dirichlet drawn column \( \mathbf{u}_r \) of \( \mathbf{U} \) represents a distribution (i.e., a “topic”) over the \( M \) objects (e.g., words) along the rows of \( \mathbf{X} \). Also note that the Poisson-gamma construction (Eq. 2.1–2.3) is equivalent to a gamma-negative binomial model (Zhou et al., 2012b) for each entry \( x_{mn} \) of \( \mathbf{X} \).

### 2.2.2 Leveraging Multi-level Side-Information

We would like to leverage the multi-level side-information available for the columns of \( \mathbf{X} \) (as shown in Fig. 2.1). To accomplish this, we augment the PFA generative model using a multi-level conditioning structure imposed on the \( N \times R \) factor score matrix \( \mathbf{V} \), whose each row \( \mathbf{v}_n = [v_{n1}, \ldots, v_{nR}] \) denotes the factor scores (or embedding) of a level-zero object \( n \).

In particular, to leverage the side-information (i.e., from level-one and above), we first model the \( r^{th} \) factor score of object \( n \) as a sum of contributions from each of the level-one nodes associated with this object

\[ v_{nr} = \sum_{l \in \mathcal{L}^{(1)}_n} v_{nrl} \] (2.6)

\[ v_{nrl} \sim \text{Ga}(g_{rl}, q_r/(1 - q_r)) \] (2.7)

where \( \mathcal{L}^{(1)} \) denotes the set of all nodes in level-one in the hierarchy and \( \mathcal{L}^{(1)}_n \) denotes the subset of these nodes associated with object \( n \) from level-zero.
Using gamma-additivity, Eq. 2.6-2.7 can be combined as

\[
v_{nr} \sim \text{Ga} \left( \sum_{l \in \mathcal{L}_n^{(1)}} g_{lr}, \frac{q_r}{1 - q_r} \right)
\]  

(2.8)

In Eq. 2.8, \(g_{lr}\) denotes the \(r^{th}\) factor score of node \(l\) at level-one (first level of side-information).

To leverage the level-two side-information, we likewise assume that the factor scores of this level-one node \(l\) can, in turn, be written as a sum of contributions from each of the level-two nodes it is associated with:

\[
g_{lr} = \sum_{p \in \mathcal{L}_l^{(2)}} g_{l rp}
\]

(2.9)

\[
g_{l rp} \sim \text{Ga}(h_{pr}, 1/\beta_0)
\]

(2.10)

\[
h_{pr} \sim \text{Ga}(s, 1/\beta_1)
\]

(2.11)

where \(\mathcal{L}^{(2)}\) denotes the set of all nodes in level-two of the side-information hierarchy and \(\mathcal{L}_l^{(2)}\) denotes the subset of these nodes associated with node \(l\) in level-one. Note that Eq. 2.9-2.10 can also be combined as

\[
g_{lr} \sim \text{Ga} \left( \sum_{p \in \mathcal{L}_l^{(2)}} h_{pr}, 1/\beta_0 \right)
\]

(2.12)

In Eq. 2.12, \(h_{pr}\) denotes the \(r^{th}\) factor score of node \(p\) at level-two (second level of side-information). Subsequently, we will refer to our model as **PFA-SSI**, as an abbreviation for **P**oisson **F**actor **A**nalysis with **S**tructural **S**ide-**I**nformation.

### 2.2.3 Learning Multi-level Embeddings

Our generative model provides a natural and effective way of learning embeddings of the objects being modeled (e.g., the documents) as well as the embeddings of the nodes that together constitute the multi-level side-information (e.g., the authors...
and affiliations or the label ontology as shown in Fig. 2.1). To see this, note that $v_n = [v_{n1}, \ldots, v_{nR}]$, $g_l = [g_{l1}, \ldots, g_{lR}]$, and $h_p = [h_{p1}, \ldots, h_{pR}]$ can be interpreted as embeddings of the $n^{th}$ level-zero object, and the $l^{th}$ level-one node and the $p^{th}$ level-two node in the multi-level side-information, respectively. Note that all these embeddings are in the same $R$-dimensional space and hence are “comparable”. Since in our model the embeddings correspond to topics, the embeddings allow us to discover the topics associated with each object as well as the topics associated with each constituent node of the side-information. For example, if the side-information is given in form of a label ontology then our model can infer the embedding of each label in the ontology and the topics associated with each label. Such a property makes our framework readily applicable for tasks such as: (1) supervised topic modeling (Ramage et al., 2009; Rabinovich and Blei, 2014) with multi-level supervision, which most of existing methods are unable to leverage in a proper way (also see Sec. 4.4 on Related Work); and (2) assigning labels to unlabeled (i.e., test) objects by inferring the embeddings of these objects, using the dictionary $U$ learned from the labeled training data, applying a standard PFA with dictionary fixed as $U$, and finding the most similar labels by comparing these inferred embeddings with the embeddings of the set of labels in the training data. Note that, if the side-information is given as a tree/ontology over labels, such an approach would even allow labeling a test object with a non-leaf label, even though the training set objects may only have leaf node labels (somewhat mimicking a zero-shot learning setting).

2.2.4 Modeling Binary $X$

If the matrix $X$ is binary, we can replace the Poisson likelihood for the counts with a Bernoulli-Poisson likelihood for binary data. The Bernoulli-Poisson model (Zhou, 2015) is based on first drawing a count-valued latent variable from a Poisson and thresholding it at one to generate the binary observation. In our model, this amounts
to the following generative model for each binary entry $x_{mn}$ in $X$

$$x_{mn} = 1(z_{mn} \geq 1), \quad z_{mn} \sim \text{Pois}(\sum_{r=1}^{R} u_{mr}v_{nr})$$

(2.13)

The rest of the generative model is the same as when $X$ is count-valued (as described in earlier sections). Marginalizing out $z_{mn}$ leads to the following

$$x_{mn} \sim \text{Ber}\left(1 - \exp(-\sum_{r=1}^{R} u_{mr}v_{nr})\right)$$

(2.14)

In contrast to the logistic/probit likelihood for binary data, the Bernoulli-Poisson construction used here is appealing due to two reasons. The first is that the computations scale in the number of nonzeros in $X$ rather than the number of observations in $X$. This is possible because, in the conditional posterior of $z_{mn}$

$$z_{mn}|x_{mn}, u_m, v_n \sim x_{mn} \cdot \text{Pois}_+(\sum_{r=1}^{R} u_{mr}v_{nr})$$

(2.15)

which means $z_{mn} = 0$ with probability one if $x_{mn} = 0$, and therefore need not be sampled if $x_{mn} = 0$. The second reason is that this link function is skewed (towards having very few nonzeros), resembling the complementary loglog function (Piegorsch, 1992; Collett, 2002), unlike the logistic/probit link, and therefore can better model highly sparse binary matrices.

2.3 Inference via Gibbs Sampling

Exact inference in our model is intractable and therefore we resort to approximate inference. Leveraging the Poisson-multinomial equivalence, which allows re-expressing a Poisson random draw as a draw from a multinomial (Dunson and Herring, 2005;
Zhou et al., 2012b), we obtain a model with full local conjugacy. This allows closed-form Gibbs sampling for all the model parameters. A key aspect of our model is that inference based on Gibbs sampling scales in the number of nonzero entries in \( X \) (for both count as well as binary data which we model as thresholded counts as discussed in Sec. 2.2.4), which makes it especially attractive for massive but sparse matrices. Although, here we only consider batch Gibbs sampling, our inference method can be easily extended to perform online Gibbs sampling (Guhaniyogi et al., 2014; Hu et al., 2015b), which will allow scaling up to even more massive data sets. We leave this extension to future work.

**Sampling the latent counts** \( x_{mnr} \) and \( x_{mnrl} \): Using the Poisson-multinomial equivalence, if the matrix \( X \) is count-valued, then the latent counts \( x_{mnr} \) and \( x_{mnrl} \) can be sampled as

\[
\{x_{mnr}\} \sim \text{Mult}(x_{mn}; \frac{u_{mr}v_{nr}}{\sum_{r=1}^{R} u_{mr}v_{nr}}) 
\]

\[
\{x_{mnrl}\} \sim \text{Mult}(x_{mn}; v_{nrl}/v_{nr})
\]

If \( X \) is binary-valued, we need to first sample the latent count \( z_{mn} \) for each nonzero \( x_{mn} \) from a truncated Poisson. Then \( x_{mnr} \) can be sampled as

\[
\{x_{mnr}\} \sim \text{Mult}(z_{mn}; \{u_{mr}v_{nr}\}/\sum_{r=1}^{R} u_{mr}v_{nr})
\]

and \( x_{mnrl} \) is sampled the same way as in equation 2.17.

**Sampling** \( u_r \): Using the multinomial-Dirichlet conjugacy, \( u_r \) is sampled as

\[
u_r \sim \text{Dir}(\alpha + x_{1.}, \ldots, \alpha + x_{M.})\]

**Sampling** \( v_{nr} \): \( v_{nr} \) can be updated by \( v_{nr} = \sum_{l \in L_n^{(i)}} v_{nrl} \), where \( v_{nrl} \) is sampled as

\[
v_{nrl} \sim \text{Ga}(q_{lr} + x_{nrl}, q_r)
\]
Sampling $q_r$: Using the additive property of Poisson distribution, we have $x_{nr} \sim \text{Pois}(v_{nr})$. Integrating out $v_{nr}$, $x_{nr}$ can be expressed as a draw from the following negative-binomial distribution

$$x_{nr} \sim \text{NB}(\sum_{l \in L_1} g_{lr}, q_r)$$  \hspace{1cm} (2.21)

Then $q_r$ can be sampled by using negative-binomial-beta conjugacy as

$$q_r \sim \text{Beta}(c \epsilon + x_{..r}, c(1 - \epsilon) + \sum_{n=1}^{N} \sum_{l \in L_1} g_{lr})$$  \hspace{1cm} (2.22)

Sampling $g_{lr}$: Using the additive property of Poisson distribution, $x_{nrl} \sim \text{Pois}(u_r v_{nrl})$, which can be further rewritten as $x_{nrl} \sim \text{Pois}(v_{nrl})$ since $u_r = 1$. Let $D_l$ be the set of all objects (i.e. documents) associated with node $l$ (i.e. an author) on the first layer, we further obtain the following equation by applying the additive property of Poisson distribution once more

$$\sum_{n \in D_l} x_{nrl} \sim \text{Pois}(\sum_{n \in D_l} v_{nrl})$$  \hspace{1cm} (2.23)

Since the gamma distribution is infinitely divisible, $\sum_{n \in D_l} v_{nrl}$ can be expressed as

$$\sum_{n \in D_l} v_{nrl} \sim \text{Ga}(|D_l| g_{lr}, q_r/(1 - q_r))$$  \hspace{1cm} (2.24)

Integrating out $\sum_{n \in D_l} v_{nrl}$ in equation 2.23 and 2.24, $\sum_{n \in D_l} x_{nrl}$ can be expressed as a negative-binomial distribution by

$$\sum_{n \in D_l} x_{nrl} \sim \text{NB}(|D_l| g_{lr}, q_r)$$  \hspace{1cm} (2.25)

Using scaling property of the gamma distribution,

$$|D_l| g_{lr} \sim \text{Ga}(\sum_{p \in A_l} h_{pr}, \frac{|D_l| g_{lr}}{\beta_0})$$  \hspace{1cm} (2.26)
Using the data augmentation method for negative-binomial distribution (Zhou and Carin, 2015), $g_{lr}$ can be sampled as

$$f_{lr} \sim \text{CRT}(\sum_{n \in D_l} x_{nrl}, |D_l|g_{lr})$$

(2.27)

$$g_{lr} \sim \text{Ga}(\sum_{p \in A_l} h_{pr} + f_{lr}, \frac{1}{\beta_0 - |D_l| \ln(1 - q_r)})$$

(2.28)

where CRT denotes the Chinese restaurant table (Zhou and Carin, 2015) distribution.

**Sampling $h_{pr}$:** According to corollary 2 in (Zhou and Carin, 2015), $f_{lr} \sim \text{Pois}(-|D_l|g_{lr} \ln(1 - q_r))$. As $g_{lr} = \sum_{p \in A_l} g_{l_{lpr}}$, $f_{lr}$ can be augmented as $f_{lr} = \sum_{p \in A_l} f_{lrp}$, where

$$f_{lrp} \sim \text{Pois}(-|D_l|g_{l_{lpr}} \ln(1 - q_r))$$

(2.29)

Let $\mathcal{M}_p$ be the set containing all level-one nodes associated with a level-two node $p$. We then have

$$\sum_{f \in \mathcal{M}_p} \ell_{fra} \sim \text{Pois}(-\ln(1 - q_r) \sum_{f \in \mathcal{M}_p} (|D_f|g_{l_{lpr}}))$$

(2.30)

$\sum_{f \in \mathcal{M}_p} f_{lrp}$ can be expressed as a negative-binomial distribution by integrating out $\sum_{f \in \mathcal{M}_p} g_{l_{lpr}}$.

$$\sum_{f \in \mathcal{M}_p} f_{lrp} \sim \text{NB}(|\mathcal{M}_p|h_{pr}, \frac{-\ln(1 - q_r)}{\beta_0 - \ln(1 - q_r)})$$

(2.31)

Using scaling property of the gamma distribution,

$$|\mathcal{M}_p|h_{pr} \sim \text{Ga}(s, |\mathcal{M}_p|/\beta_1)$$

(2.32)

Applying the data augmentation method for negative-binomial distribution once more, $h_{pr}$ is sampled as

$$f'_{pr} \sim \text{CRT}(\sum_{l \in \mathcal{M}_p} f_{lrp}, s)$$

(2.33)

$$h_{pr} \sim \text{Ga}(s + f'_{pr}, \frac{1}{\beta_1 - |\mathcal{M}_p| \ln(1 - Q_r)})$$

(2.34)
where $Q_r = \frac{-\ln(1-q_r)}{\beta_0 - \ln(1-q_r)}$.

2.4 Related Work

Our work has interesting parallels with some existing methods that attempt to leveraging side-information when modeling discrete data; for example, methods that can incorporate supervision in Latent Dirichlet Allocation (LDA) based topic models (Rosen-Zvi et al., 2004; Mcauliffe and Blei, 2008; Lacoste-Julien et al., 2009; Wang and Blei, 2011; Zhu et al., 2012), and recent work on utilizing side-information in matrix factorization models for count/binary data (Gopalan et al., 2014; Acharya et al., 2015). These class of methods are, however, limited in the type of side-information that can be leveraged, as they usually do not assume any structure within the associated side-information (which is usually given in form of a flat feature vector or a single binary/multi-class label associated with each object).

Among other related work, our framework is somewhat similar in spirit to hierarchically supervised LDA model (Perotte et al., 2011) and the topic model for taxonomies (Bakalov et al., 2012). However, these methods are designed strictly for leveraging very specific, taxonomy-based side-information, whereas our framework can handle more general forms of structural side-information, and a taxonomy is just one of the examples of such side-information.

Structural side-information can also be utilized for specific cases using specialized hierarchical generative models such as the hierarchical Dirichlet Process (Teh et al., 2006). However, in order to properly utilize the type of multi-level side-information our framework can easily utilize, such models would require significant modeling sophistication. Moreover, inference can be considerably more challenging in such models.

In addition to being richer in terms of the types of structural side-information that can be leveraged, our fully Bayesian framework is also conceptually simpler.
in construction as compared to the aforementioned class of methods. At the same
time, our model is easily amenable to efficient inference, and has several interesting
properties that the existing methods lack (e.g., learning embeddings the objects be-
ing modeled as well as embeddings of the nodes at all levels in the side-information),
and is applicable in a wide variety of applications, such as topic modeling, recom-
mender systems, network modeling, while leveraging structural side-information in a
principled way.

2.5 Experiments

We evaluate our model, both quantitatively (in its ability to predict missing data
in the matrix $X$) and quantitatively (interpretability of the topics and embeddings
learned by the model), by performing experiments on six real-world data sets. For
four of the data sets, the matrix $X$ has count-valued observations, whereas for the re-
maining two (Cora and CiteSeer $^1$), the observations in $X$ are binary. The description
of each data set and the associated side-information is given below:

- **20 Newsgroup:** This data $^2$ consists of 18,774 documents (vocabulary size
5638) organized into 20 groups where each of the groups can be further classi-
fied into a super-group (there are a total of seven super-groups). The indices
for groups are as follows. 1: alt.atheism; 2: comp.graphics; 3: comp.os.ms-
windows.misc; 4: comp.sys.ibm.pc.hardware; 5: comp.sys.mac.hardware; 6:
comp.windows.x; 7: misc.forsale; 8: rec.autos; 9: rec.motorcycles; 10: rec.sport.baseball;
16: soc.religion.christian; 17: talk.politics.guns; 18: talk.politics.mideast; 19:
talk.politics.misc; 20: talk.religion.misc. Thus the side-information can be

\footnote{http://preview.tinyurl.com/jq4sag6}
\footnote{http://qwone.com/~jason/20Newsgroups/}
thought of as a two-level taxonomy. For this data, $X$ is $5638 \times 18774$ word-count matrix.

- **State of the Union:** This dataset includes 225 state of the union messages (vocabulary size 7518) delivered annually by 41 presidents of the US from 1790 to 2014 (Wang and McCallum, 2006). Party affiliation for each president is also available (Independent, Federalist, Democratic-Republican, Democrat, Whig, Republican). Thus the side-information is a two-level taxonomy. For this data, $X$ is $7518 \times 225$ word-count matrix.

- **Scholars:** This dataset includes abstracts of 20,149 papers (vocabulary size 8663 words) written by 2,425 researchers associated with 200 affiliations at a US university’(Hu et al., 2015b). The side-information is a two-level hierarchy. For this data, $X$ is $8663 \times 20149$ word-count matrix.

- **NIPS:** 2484 articles (vocabulary size 14036) of the NIPS conferences from 1988 to 2003. The corpus consists of 2865 authors. For this data $^3$, the side-information only consists of a single level (author identities). For this data, $X$ is $14036 \times 2484$ word-count matrix.

- **Cora:** The data contains 2708 research papers from 7 sub-areas of machine learning: case-based reasoning, genetic algorithms, neural networks, probabilistic methods, reinforcement learning, rule learning, and theory. There are overall 5429 citations (links) between the papers.

- **CiteSeer:** The Citeseer data set contains 3312 papers which can be classified into 6 categories: Agents, AI, DB, IR, ML, and HCI. There are overall 4591 citations (links) between the papers.

$^3$ http://ai.stanford.edu/~gal/Data/NIPS/
Table 2.1: Loglikelihood comparison between PFA and PFA-SSI for State of the Union (STOU), 20 newsgroup (20 News), Scholar, and NIPS data sets. 10% data was held out as testing data, and 90% used as training. Results are averaged over 10 random splits of training and test.

<table>
<thead>
<tr>
<th>Methods</th>
<th>STOU</th>
<th>20 Newsgroup</th>
<th>Scholar</th>
<th>NIPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PFA (Zhou et al., 2012b)</td>
<td>-23232</td>
<td>-522876</td>
<td>-506905</td>
<td>-345853</td>
</tr>
<tr>
<td>PFA-SSI</td>
<td>-22168</td>
<td>-397969</td>
<td>-389060</td>
<td>-293404</td>
</tr>
</tbody>
</table>

Table 2.2: Loglikelihood, AUC and AUC-PR comparison between PFA and PFA-SSI for Cora and CiteSeer datasets. 10% data was held out as testing data, and 90% used as training. Results are averaged over 10 random splits of training and test.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Cora Loglike</th>
<th>Cora AUC</th>
<th>Cora AUC-PR</th>
<th>CiteSeer Loglike</th>
<th>CiteSeer AUC</th>
<th>CiteSeer AUC-PR</th>
</tr>
</thead>
<tbody>
<tr>
<td>PFA (Zhou et al., 2012b)</td>
<td>-9057</td>
<td>0.699</td>
<td>0.740</td>
<td>-9973</td>
<td>0.545</td>
<td>0.670</td>
</tr>
<tr>
<td>PFA-SSI</td>
<td>-3682</td>
<td>0.808</td>
<td>0.841</td>
<td>-4042</td>
<td>0.788</td>
<td>0.814</td>
</tr>
</tbody>
</table>

All of our experiments were performed on a standard desktop with 12 GB RAM. For each data set, we set the number of topics (\(R\)) to be 200, which serves as an upper bound on the number of topics and the model can prune away the unnecessary topics due to the beta-negative binomial construction (Zhou et al., 2012b) of our model. In all our experiments, we fix the hyperparameters \(\beta_0\) and \(\beta_1\) to 1, \(\epsilon = 1/R\), and the Dirichlet hyperparameter \(\alpha\) was fixed at 0.1. These hyperparameter settings worked well for all the data sets.

2.5.1 Predicting Held-out Data

We evaluate our model on predicting missing data in the matrix \(X\) by holding out 10% of the observations and predicting them via our non-negative matrix factorization approach, using the remaining 90% data as training data. Each experiment was repeated 10 times and the average accuracies are reported.

**Baseline:** We compare our model with Poisson Factor Analysis (PFA) (Zhou et al., 2012b), which is a state-of-the-art non-negative matrix factorization method and also subsumes many other discrete matrix factorization methods (including
gamma-Poisson count matrix factorization, LDA, etc.) as special cases. Also note that the PFA model of (Zhou et al., 2012b) can only handle count data. Therefore, to apply this baseline for the two binary data sets, we modified the PFA implementation ourselves by replacing the Poisson likelihood model by the Bernoulli-Poisson model. Also, we are unable to provide here comparison with other baselines because, to the best of our knowledge, none of the existing methods can incorporate the type of multi-level side-information available for the count/binary matrices we use in our experiments.

Table 2.1 shows the results for the cases when $X$ is count-valued and Table 2.2 shows the results for the cases when $X$ is binary-valued. For the count-valued data sets, we report the heldout log-likelihood. For the binary-valued data sets, we report the heldout log-likelihood as well as area under the ROC curve (AUC) and area under the precision-recall curve (AUC-PR). As shown in Table 2.1 and Table 2.2, our model significantly outperforms PFA on all the data sets, which shows our model’s ability in leveraging structural side-information in an effective way.

Table 2.3: Most prominent topic for 12 groups in 20 newsgroup data

<table>
<thead>
<tr>
<th>atheism</th>
<th>graphics</th>
<th>win.misc</th>
<th>pc.hardware</th>
<th>win.x</th>
<th>forsale</th>
</tr>
</thead>
<tbody>
<tr>
<td>religion</td>
<td>image</td>
<td>windows</td>
<td>dos</td>
<td>file</td>
<td>sale</td>
</tr>
<tr>
<td>real</td>
<td>graphics</td>
<td>pc</td>
<td>system</td>
<td>windows</td>
<td>offer</td>
</tr>
<tr>
<td>god</td>
<td>bit</td>
<td>mail</td>
<td>drive</td>
<td>program</td>
<td>st</td>
</tr>
<tr>
<td>book</td>
<td>data</td>
<td>ac</td>
<td>scsi</td>
<td>output</td>
<td>shipping</td>
</tr>
<tr>
<td>true</td>
<td>computer</td>
<td>os</td>
<td>card</td>
<td>server</td>
<td>condition</td>
</tr>
<tr>
<td>question</td>
<td>software</td>
<td>dos</td>
<td>mb</td>
<td>entry</td>
<td>price</td>
</tr>
<tr>
<td>liar</td>
<td>processing</td>
<td></td>
<td></td>
<td>mit</td>
<td>email</td>
</tr>
<tr>
<td>hockey</td>
<td>crypt</td>
<td>electronics</td>
<td>med</td>
<td>space</td>
<td>christian</td>
</tr>
<tr>
<td>game</td>
<td>db</td>
<td>data</td>
<td>doctor</td>
<td>space</td>
<td>mary</td>
</tr>
<tr>
<td>ca</td>
<td>key</td>
<td>circuit</td>
<td>patients</td>
<td>nasa</td>
<td>entry</td>
</tr>
<tr>
<td>hockey</td>
<td>encryption</td>
<td>signal</td>
<td>msg</td>
<td>billion</td>
<td>church</td>
</tr>
<tr>
<td>espn</td>
<td>chip</td>
<td>input</td>
<td>disease</td>
<td>cost</td>
<td>win</td>
</tr>
<tr>
<td>team</td>
<td>government</td>
<td>output</td>
<td>day</td>
<td>extra</td>
<td>rules</td>
</tr>
<tr>
<td>nhl</td>
<td>clipper</td>
<td>pin</td>
<td>chronic</td>
<td>based</td>
<td>sin</td>
</tr>
<tr>
<td>year</td>
<td>keys</td>
<td>loop</td>
<td>medical</td>
<td>station</td>
<td>scripture</td>
</tr>
</tbody>
</table>
Figure 2.2: 20 newsgroups data. Left Figure: Inferred similarities between the level-one nodes (i.e., between the 20 groups) in the side-information. Right Figure: Inferred similarities between the level-one and the level-two nodes (i.e., 20 groups and 7 supergroups) in the side-information. The numbers are indices for groups, and numbers with same color indicate that the corresponding groups are associated with the same supergroup.

Figure 2.3: State of the union data. Left Figure: Inferred similarities between the presidents in the embedding space. The numbers before each president are labels for parties. 1: Independent; 2: Federalist; 3: Democratic-Republican; 4: Democrat; 5: Whig; 6: Republican. In the legend, the names of all presidents from the same party are shown in the same color. Right Figure: Inferred similarities between presidents (level-one nodes) and parties (level-two nodes).
Table 2.4: Most prominent topic for each supergroup in 20 newsgroup data

<table>
<thead>
<tr>
<th>religion</th>
<th>comp</th>
<th>auto</th>
<th>sport</th>
<th>sci</th>
<th>politics</th>
<th>forsale</th>
</tr>
</thead>
<tbody>
<tr>
<td>god</td>
<td>image</td>
<td>bike</td>
<td>game</td>
<td>space</td>
<td>gun</td>
<td>sale</td>
</tr>
<tr>
<td>bible</td>
<td>graphics</td>
<td>dod</td>
<td>hockey</td>
<td>launch</td>
<td>government</td>
<td>offer</td>
</tr>
<tr>
<td>jesus</td>
<td>bit</td>
<td>ca</td>
<td>period</td>
<td>satellite</td>
<td>crime</td>
<td>st</td>
</tr>
<tr>
<td>christian</td>
<td>data</td>
<td>back</td>
<td>april</td>
<td>satellites</td>
<td>control</td>
<td>shipment</td>
</tr>
<tr>
<td>christians</td>
<td>computer</td>
<td>car</td>
<td>espn</td>
<td>technology</td>
<td>firearms</td>
<td>condition</td>
</tr>
<tr>
<td>christ</td>
<td>software</td>
<td>bmw</td>
<td>play</td>
<td>commercial</td>
<td>news</td>
<td>price</td>
</tr>
<tr>
<td>word</td>
<td>processing</td>
<td>front</td>
<td>team</td>
<td>system</td>
<td>criminal</td>
<td>email</td>
</tr>
</tbody>
</table>

Table 2.5: Two of the most prominent topics (for considered time-period of 1988-2003) for three of the authors in NIPS data

<table>
<thead>
<tr>
<th>Geoff Hinton</th>
<th>Michael Jordan</th>
<th>Peter Bartlett</th>
</tr>
</thead>
<tbody>
<tr>
<td>units</td>
<td>probability</td>
<td>theorem</td>
</tr>
<tr>
<td>hidden</td>
<td>space</td>
<td>tree</td>
</tr>
<tr>
<td>weights</td>
<td>parameters</td>
<td>bound</td>
</tr>
<tr>
<td>hinton</td>
<td>likelihood</td>
<td>case</td>
</tr>
<tr>
<td>information</td>
<td>dimensional</td>
<td>training</td>
</tr>
<tr>
<td>inputs</td>
<td>bayesian</td>
<td>proof</td>
</tr>
<tr>
<td>net</td>
<td>cluster</td>
<td>decision</td>
</tr>
<tr>
<td>set</td>
<td>prior</td>
<td>dimension</td>
</tr>
<tr>
<td>internal</td>
<td>structure</td>
<td>test</td>
</tr>
<tr>
<td>bias</td>
<td>maximum</td>
<td>theory</td>
</tr>
<tr>
<td></td>
<td>information</td>
<td>boosting</td>
</tr>
</tbody>
</table>

Table 2.6: Five most similar authors (for considered time-period of 1988-2003) for three of the authors in NIPS data

<table>
<thead>
<tr>
<th>Geoff Hinton</th>
<th>Michael Jordan</th>
<th>Peter Bartlett</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE Rumelhart</td>
<td>Nir Friedman</td>
<td>Robert Williamson</td>
</tr>
<tr>
<td>J McClelland</td>
<td>Sathiya Keerthi</td>
<td>D Helmbold</td>
</tr>
<tr>
<td>J Elman</td>
<td>Miguel Carreira-Perpinan</td>
<td>John Shawe-Taylor</td>
</tr>
<tr>
<td>Antony Bloesch</td>
<td>Tony Jebara</td>
<td>E Sontag</td>
</tr>
<tr>
<td>Ryotaro Kamimura</td>
<td>David MacKay</td>
<td>V Maiorov</td>
</tr>
</tbody>
</table>

2.5.2 Qualitative Analyses

We perform qualitative analyses of our results on various data sets using the topics and the embeddings learned by our model.

20 Newsgroup Data: For this data, Table 2.3 shows the most prominent topic
associated with 12 groups of the level-one side-information. Note that our model learns embeddings of each of these groups and the non-negative embeddings of each group can be used to identify the most active topic associated with that group. Likewise, Table 2.4 shows the most prominent topic associated with each of the 7 super-groups of the level-two side-information. As Table 2.3 and Table 2.4 show, the topics inferred are closely related to the corresponding groups/super-groups. Using the inferred group/super-group embeddings, we also compute cosine similarities between groups and between groups and supergroups. Fig. 5.4 shows the plots of the estimated similarities. As the plots show, similarities between groups that belong to the same super-group are high, as reflected by the block-diagonal pattern in Fig. 5.4 (left). Likewise, each group has a higher inferred similarity with its own super-group as compared to other super-groups, as shown in Fig. 5.4 (right). These results show that the embeddings learned by our model are meaningful and are consistent with the ground-truth.

**State of the Union Data:** For the State of the Union data, we use the inferred embeddings of presidents and parties to compute president-president similarity and president-party similarity. The resulting plots are shown in Fig. 2.3. It is interesting to note that the president-president inferred similarity plot shows a block-diagonal structure (for better visualization, the president indices are ordered based on the party indices), with presidents from the same party inferred to be highly similar with each-other. This suggests that the side-information from level-two nodes (parties) is effectively transferred to level-one nodes (presidents).

**NIPS Data:** We next look at the topics inferred from the NIPS data. Using the inferred embeddings for each author, we rank the most prominent topics for each author (based on the embedding scores). Table 2.5 shows two most active topics for each of three of the authors in NIPS data. As Table 2.5 shows, the inferred most prominent topics for each of these authors are consistent with what these authors
were best known for the time-period (1988-2003) covered by this data collection. We also perform an experiment to find the most similar authors for a given author. For this, we use the author embeddings to compute author-author similarity and, in Table 2.6, show the five most similar authors for each authors from a set of three authors. The results in Table 2.5 and Table 2.6 show that the inferred embeddings can provide a good explanation of the data.

2.5.3 Classification via Inferred Embeddings

The embeddings learned by our model can also be useful for classification tasks. To demonstrate this, we perform an experiment on multiclass classification. For this experiment, we use the 20 newsgroup data, which is divided into a training set consisting of 11269 documents and a test set consisting of 7507 documents. We use the training set to train our PFA-SSI model and use the word-topic matrix $U$ and the label embedding matrix $G$ learned from the training data to predict the labels for test set documents as follows: we infer the document embedding $V_{test}$ of each test document by sampling from the posterior conditioning on $U$ and then find the most similar label for each test document by comparing the inferred test document’s embedding with embedding of each label. As a baseline, we fit an LDA model on training and test documents, train a multiclass SVM on the topic proportions and labels of the training data, and use the learned classifier to predict the labels for test documents. Our model gave a classification accuracy of 63.7% whereas the LDA+SVM baseline gave a classification accuracy of 61.5%. This experiment shows that our model, although not originally designed for classification tasks, can nevertheless achieve reasonable classification accuracies because the supervision enhances the discriminative power of the embeddings learned by our model.
3

Scalable Bayesian Non-Negative Tensor Factorization for Massive Count Data

3.1 Introduction

Discovering interpretable latent structures in complex multiway (tensor) data is an important problem when learning from polyadic relationships among multiple sets of objects. Tensor factorization (Kolda and Bader, 2009; Cichocki et al., 2009) is a promising way of extracting such latent structures, by factorizing the original tensor. The inferred factors can be used to analyze objects in each mode of the tensor, or to perform other tasks such as tensor completion.

Of particular interest in the context of such multiway data are sparsely-observed count-valued tensors. Tensor data of this form are routinely encountered in many applications. For example, in analyzing a database of scientific publications, the data may be in form of a sparse four-way count-valued tensor (authors × words × journals × years). Another application where multiway count data is routinely encountered is the analysis of contingency tables (Johndrow et al., 2014) which represent the co-occurrence statistics of multiple sets of objects.
We present a scalable model for analyzing such sparsely-observed data. Our framework is based on a beta-negative binomial construction, which provides a principled generative model for tensors with sparse and potentially overdispersed count data, and produces a non-negative tensor factorization. In addition to performing non-negative tensor factorization and tensor completion for count-valued tensors, our model has the property that each latent factor inferred for a tensor mode also represents a distribution (or “topic”, as in topic models) over the objects of that tensor mode; our model naturally accomplishes this by placing a Dirichlet prior over the columns of the factor matrix of each tensor mode. In addition to providing an expressive and interpretable model for analyzing sparse count-valued tensors, the model automatically infers the rank of the decomposition, which side-steps the crucial issue of pre-specifying the rank of the decomposition (Kolda and Bader, 2009; Rai et al., 2014; Zhao et al., 2015).

Our framework consists of a set of batch and scalable online inference methods. Using a reparameterization of the Poisson distribution as a multinomial allows us to achieve conjugacy, which facilitates closed-form Gibbs sampling as well as variational Bayes (VB) inference. Moreover, we provide three online inference algorithms, based on conditional density filtering (Guhaniyogi et al., 2014), based on stochastic variational inference (Hoffman et al., 2013), and stochastic gradient MCMC which additionally leverages the simplicial structure (Patterson and Teh, 2013) of the model parameters and the intrinsic geometry of the target posterior. These inference algorithms enable scaling up the model to massive-sized tensor data.

One of the motivations behind our work is analyzing massive multiway data for tasks such as understanding thematic structures in scholarly databases (e.g., to design better recommender systems for scholars), understanding consumer behavior from shopping patterns of large demographies (e.g., to design better marketing and supply strategies), and understanding international relations in political science studies. We
provide qualitative analyses for such applications on large-scale real-world data sets, and the scalability behavior of our model.

3.2 Canonical PARAFAC Decomposition

Given a tensor $\mathbf{Y}$ of size $n_1 \times n_2 \times \cdots \times n_K$, with $n_k$ denoting the size of $\mathbf{Y}$ along the $k^{th}$ mode (or “way”) of the tensor, the goal in a Canonical PARAFAC (CP) decomposition (Kolda and Bader, 2009) is to decompose $\mathbf{Y}$ into a set of $K$ factor matrices $\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(K)}$ where $\mathbf{U}^{(k)} = [\mathbf{u}_1^{(k)}, \ldots, \mathbf{u}_R^{(k)}]$, $k = \{1, \ldots, K\}$, denotes the $n_k \times R$ factor matrix associated with mode $k$. More precisely, in its most general form, CP decomposition expresses the tensor $\mathbf{Y}$ via a weighted sum of $R$ rank-1 tensors as $\mathbf{Y} \sim f(\sum_{r=1}^{R} \lambda_r, \mathbf{u}_r^{(1)} \odot \cdots \odot \mathbf{u}_r^{(K)})$. The form of $f$ depends on the type of data being modeled (e.g., $f$ can be Gaussian for real-valued, Bernoulli-logistic for binary-valued, Poisson for count-valued tensors). Here $\lambda_r$ is the weight associated with the $r^{th}$ rank-1 component, the $n_k \times 1$ column vector $\mathbf{u}_r^{(k)}$ represents the $r^{th}$ latent factor of mode $k$, and $\odot$ denotes vector outer product.

3.3 Beta-negative Binomial CP Decomposition

We focus on modeling count-valued tensor data (Chi and Kolda, 2012) and assume the following generative model for the tensor $\mathbf{Y}$

$$
\mathbf{Y} \sim \text{Pois}\left(\sum_{r=1}^{R} \lambda_r, \mathbf{u}_r^{(1)} \odot \cdots \odot \mathbf{u}_r^{(K)}\right) \tag{3.1}
$$

$$
\mathbf{u}_r^{(k)} \sim \text{Dir}(a^{(k)}, \ldots, a^{(k)}) \tag{3.2}
$$

$$
\lambda_r \sim \text{Gamma}(g, \frac{p_r}{1-p_r}) \tag{3.3}
$$

$$
p_r \sim \text{Beta}(c \epsilon, c(1-\epsilon)) \tag{3.4}
$$
We use subscript \( i = \{i_1, \ldots, i_K\} \) to denote the \( K \)-dimensional index of the \( i \)-th entry in the tensor \( Y \). Using this notation, the \( i \)-th entry of the tensor can be written as \( y_i \sim \text{Pois}(\sum_{r=1}^{R} \lambda_r \prod_{k=1}^{K} d_{ikr}(k)) \). The data consist of \( N \) observations \( \{y_i\}_{i=1}^{N} \) from the tensor \( Y \).

Since the gamma-Poisson mixture distribution is equivalent to a negative binomial distribution (Kozubowski and Podgórski, 2008), (4.3) and (4.5), coupled with the beta prior (Eq 4.6) on \( p_r \), lead to what we will call the beta-negative binomial CP (BNBCP) decomposition model. A few things about our model worth noting are:

- The Dirichlet prior on the factors \( u_{r}^{(k)} \) naturally imposes non-negativity constraints (Chi and Kolda, 2012) on the factor matrices \( U(1), \ldots, U(K) \). Moreover, since the columns \( u_{r}^{(k)} \) of these factor matrices sums to 1, each \( u_{r}^{(k)} \) represents a distribution (e.g., a “topic”) over the \( n_k \) entities in mode \( k \).

- The gamma-beta hierarchical construction of \( \lambda_r \) (Eq 4.5 and 4.6) allows inferring the rank of the tensor by setting an upper bound \( R \) on the number of factors and letting the inference procedure infer the appropriate number of factors by shrinking the coefficients \( \lambda_r \)'s to close to zero for the irrelevant factors.

- The resulting negative binomial model is useful for modeling overdispersed count data in cases where the Poisson likelihood may not be suitable.

- Using alternate parameterizations (Section 3.3.1) of the Poisson distribution in (4.3) lead to a fully conjugate model and facilitate efficient Gibbs sampling and variational Bayes (VB) inference, in both batch as well as online settings.

### 3.3.1 Reparametrizing the Poisson Distribution

The generative model described in Eq (4.3)-(4.6) is not conjugate. We now describe two equivalent parametrizations (Dunson and Herring, 2005; Zhou et al., 2012b) of
Eq (4.3), which transform Eq (4.3)-(4.6) into a fully conjugate model and facilitate easy-to-derive and scalable inference procedures. These parameterizations are based on a data augmentation scheme described below.

The first parametrization expresses the \( i \)-th count-valued entry \( y_i \) of the tensor \( \mathbf{Y} \) as a sum of \( R \) latent counts \( \{ \tilde{y}_{ir} \}_{r=1}^R \)

\[
y_i = \sum_{r=1}^R \tilde{y}_{ir}, \quad \tilde{y}_{ir} \sim \text{Pois}(\lambda_r \prod_{k=1}^K u_{ikr}^{(k)})
\]  

(3.5)

The second parametrization assumes the vector \( \{ \tilde{y}_{ir} \}_{r=1}^R \) of latent counts is drawn from a multinomial as

\[
\tilde{y}_{i1}, \ldots, \tilde{y}_{iR} \sim \text{Mult}(y_i; \zeta_{i1}, \ldots, \zeta_{iR})
\]

\[
\zeta_{ir} = \frac{\lambda_r \prod_{k=1}^K u_{ikr}^{(k)}}{\sum_{r=1}^R \lambda_r \prod_{k=1}^K u_{ikr}^{(k)}}
\]  

(3.6)

The above parameterizations follows from the following lemma (Dunson and Herring, 2005; Zhou et al., 2012b):

**Lemma 1.** Suppose that \( x_1, \ldots, x_R \) are independent random variables with \( x_r \sim \text{Pois}(\theta_r) \) and \( x = \sum_{r=1}^R x_r \). Set \( \theta = \sum_{r=1}^R \theta_r \); let \( (z, z_1, \ldots, z_R) \) be another set of random variables such that \( z \sim \text{Pois}(\theta) \), and \( (z_1, \ldots, z_R) \mid z \sim \text{Mult}(z; \frac{\theta_1}{\theta}, \ldots, \frac{\theta_R}{\theta}) \). Then the distribution of \( x = (x, x_1, \ldots, x_R) \) is the same as the distribution of \( z = (z, z_1, \ldots, z_R) \).

These parameterizations, along with the fact that the columns \( u_{ikr}^{(k)} \) of each factor matrix are drawn from a Dirichlet, allows us to leverage the Dirichlet-multinomial conjugacy and derive simple Gibbs sampling and variational Bayes (VB) inference update equations, as described in Section 5.3.
3.4 Inference

We first present the update equations for batch Gibbs sampling (Section 3.4.1) and batch VB inference (Section 3.4.2). We then present two online inference algorithms, based on: (i) conditional density filtering (Guhaniyogi et al., 2014), which provides an efficient way to perform online MCMC sampling using conditional sufficient statistics of the model parameters; and (ii) stochastic variational inference (Hoffman et al., 2013), which will allow scaling up VB inference by processing data in small minibatches.

We also define two quantities $s_{j,r}^{(k)} = \sum_{i:i_k=j} \tilde{y}_{ir}$ and $s_r = \sum_i \tilde{y}_{i,r}$ which denote aggregates (sufficient statistics) computed using the latent counts $\tilde{y}_{ir}$. These quantities appear at various places in the description of the inference algorithms we develop.

3.4.1 Gibbs Sampling

Our Gibbs sampling procedure alternates between the following steps:

- **Sampling $\tilde{y}_{ir}$**: The latent counts $\{\tilde{y}_{ir}\}_{r=1}^{R}$ are sampled from a multinomial (3.6).

- **Sampling $u_r^{(k)}$**: Due to the Dirichlet-multinomial conjugacy, the columns of each factor matrix have Dirichlet posterior and are sampled as

  $$u_r^{(k)} \sim \text{Dir}(a^{(k)} + s_{1,r}^{(k)}, a^{(k)} + s_{2,r}^{(k)}, \ldots, a^{(k)} + s_{n_k,r}^{(k)})$$

  (3.7)

- **Sampling $p_r$**: Using the fact that $s_r = \sum_i \tilde{y}_{i,r}$ and marginalizing over the $u_{i_k,i}^{(k)}$’s in (4.13), we have $s_r \sim \text{Pois}(\lambda_r)$. Using this, along with (4.5), we can express $s_r$ using a negative binomial distribution, i.e., $s_r \sim \text{NB}(g_r, p_r)$. Then,
due to the conjugacy between negative binomial and beta, we can sample $p_r$ as

$$p_r \sim \text{Beta}(c \epsilon + s_r, c(1 - \epsilon) + g_r)$$  \hspace{1cm} (3.8)

- **Sampling** $\lambda_r$: Again using the fact that $s_r \sim \text{Pois}(\lambda_r)$, and due to the gamma-Poisson conjugacy, we have

$$\lambda_r \sim \text{Gamma}(g_r + s_r, p_r)$$  \hspace{1cm} (3.9)

**Computational Complexity:** Sampling the latent counts $\{\tilde{y}_{ir}\}_{r=1}^R$ for each nonzero observation $y_{i}$ (note that for $y_i = 0$, the latent counts are trivially zero) requires computing $\{\zeta_{ir}\}_{r=1}^R$, and computing each $\zeta_{ir}$ requires $O(K)$ time (Eq 3.6). Therefore, sampling all the latent counts $\{\tilde{y}_{ir}\}_{r=1}^R$ requires $O(NRK)$ time. Sampling the latent factors $\{u^{(k)}_r\}_{r=1}^R$ for the $K$ tensor modes requires $O(RK)$ time. Sampling $\{p_r\}_{r=1}^R$ and $\{\lambda_r\}_{r=1}^R$ requires $O(R)$ time each. Of all these steps, sampling the latent counts $\{\tilde{y}_{ir}\}_{r=1}^R$ (which are also used to compute the sufficient statistics $s_{j,r}^{(k)}$ and $s_r$) is the most dominant step, leading to an overall time-complexity of $O(NRK)$ for the Gibbs sampling procedure.

The linear dependence on $N$ (number of nonzeros) is especially appealing because most real-world count-valued tensors are extremely sparse (have much less than even 1% nonzeros). In contrast to the standard negative-binomial models for count data, for which the inference complexity also depends on the zeros whose number may be massive (and therefore heuristics, such as subsampling the zeros, are needed), the reparametrizations (Section 3.3.1) used by our model allow us to ignore the zeros in the multinomial sampling step (the sufficient statistics do not depend on the zero entries in the tensor), thereby significantly speeding up the inference.
3.4.2 Variational Bayes Inference

Using the mean-field assumption (Jordan et al., 1999), we approximate the target posterior distribution by

$$Q = \prod_{i,r} q(y_{ir}) \prod_{k,r} q(u_{kr}^{(k)}) \prod_r q(\lambda_r) \prod_r q(p_r).$$

Our fully conjugate model enables closed-form variational Bayes (VB) inference updates, with the distribution $q(y_{ir})$, $q(u_{kr}^{(k)})$, $q(\lambda_r)$, and $q(p_r)$ being multinomial, Dirichlet, beta, and gamma, respectively. We summarize the update equations for the variational parameters of each of these distributions, below:

- **Updating $\tilde{y}_{ir}$:** Using (3.6), the updates for $y_{ir}$ are given by $\mathbb{E}[y_{ir}] = y_i \tilde{\zeta}_{ir}$ where $\tilde{\zeta}_{ir} = \frac{\zeta_{ir}}{\sum_{r=1}^{R} \zeta_{ir}}$ and $\zeta_{ir}$ can be computed as

$$\tilde{\zeta}_{ir} = \exp\{\Psi(s_r + g_r) + \ln(p_r) + \sum_{k=1}^{K} \Psi(s_{ik,r}^{(k)} + a^{(k)}) - \Psi\left[\sum_{k=1}^{K} (s_{ik,r}^{(k)} + a^{(k)})\right]\} \ (3.10)$$

where $\Psi(.)$ is the digamma function, which is the first derivative of the logarithm of the gamma function.

- **Updating $u_{kr}^{(k)}$:** The mean-field posterior $q(u_{kr}^{(k)})$ is Dirichlet with each of the component means given by $\mathbb{E}[u_{kr}^{(k)}] = \frac{\rho_{kr}^{(k)}}{\sum_{k=1}^{K} \rho_{kr}^{(k)}}$ where $\rho_{kr}^{(k)} = a^{(k)} + s_{kr}^{(k)}$.

- **Updating $p_r$:** The mean-field posterior $q(p_r)$ is beta with mean given by $\mathbb{E}[p_r] = \frac{p_{ra}}{p_{ra} + p_{rb}}$ where $p_{ra} = c\epsilon + s_r$, $p_{rb} = c(1-\epsilon) + g_r$.

- **Updating $\lambda_r$:** The mean-field posterior $q(\lambda_r)$ is gamma with mean given by $\mathbb{E}[\lambda_r] = \lambda_{ra}\lambda_{rb}$, where $\lambda_{ra} = (g_r + s_r)$ and $\lambda_{rb} = p_r$.  

34
A note on Gibbs vs VB: The per-iteration time-complexity of the VB inference procedure is also $O(NRK)$. It is to be noted however that, in practice, one iteration of VB in this model is a bit more expensive than one iteration of Gibbs, due to the digamma function evaluation for the $\tilde{\zeta}_{ir}$ which is needed in VB when updating the $\tilde{y}_{ir}$'s. Prior works on Bayesian inference for topic models (Heinrich and Goesele, 2009) also support this observation.

3.4.3 Online Inference

Batch Gibbs (Section 3.4.1) and VB (Section 3.4.2) inference algorithms are simple to implement and efficient to run on moderately large-sized problems. These algorithms can however be slow to run for massive data sets (e.g., where the number of tensor entries $N$ and/or the dimension of the tensor is massive). The Gibbs sampler may exhibit slow mixing and the batch VB may be slow to converge. To handle such massive tensor data, we develop two online inference algorithms. The first is online MCMC based conditional density filtering (Guhaniyogi et al., 2014), while the second is based on stochastic variational inference (Hoffman et al., 2013). Both these inference algorithms allow processing data in small minibatches and enable our model to analyze massive and/or streaming tensor data.

Conditional Density Filtering

The conditional density filtering (CDF) algorithm (Guhaniyogi et al., 2014) for our model selects a minibatch of tensor entries at each iteration, samples the latent counts $\{\tilde{y}_{ir}\}_{r=1}^R$ for these entries conditiond on the previous estimates of the model parameters, updates the sufficient statistics $s^{(k)}_{j,r}$ and $s_r$ using these latent counts (as described below), and resamples the model parameters conditioned on these sufficient statistics. Denoting $I_t$ as data indices in minibatch at round $t$, the algorithm proceeds as
• **Sampling** \( \tilde{y}_i \): For all \( i \in I_t \), sample the latent counts \( \tilde{y}_{ir(i \in I_t)} \) using (3.6).

• **Updating the conditional sufficient statistics:** Using data from the current minibatch, update the conditional sufficient statistics as:

\[
\begin{align*}
    s_{j,r}^{(k,t)} &= (1 - \gamma_t)s_{j,r}^{(k,t-1)} + \gamma_t \frac{N}{B} \sum_{i \in I_t, i_k = j} \tilde{y}_{ir} \\
    s_{r}^{(t)} &= (1 - \gamma_t)s_{r}^{(t-1)} + \gamma_t \frac{N}{B} \sum_{i \in I_t} \tilde{y}_{ir} 
\end{align*}
\] (3.11) (3.12)

Note that the updated conditional sufficient statistics (CSS), indexed by superscript \( t \), is a weighted average of the old CSS, indexed by superscript \( t - 1 \), and of that computing only using the current minibatch (of size \( B \)). In addition, the latter term is further weighted by \( N/B \) so as to represent the average CSS over the entire data. In the above, \( \gamma_t \) is defined as \( \gamma_t = (t_0 + t)^{-\kappa}, t_0 \geq 0 \), and \( \kappa \in (0.5, 1] \) is needed to guarantee convergence (Cappé and Moulines, 2009).

• **Updating** \( u_r^{(k)}, p_r, \lambda_r \): Using the updated CSS, draw \( M \) samples for each of the model parameters \( \{u_r^{(k,m)}, p_r^{(m)}, \lambda_r^{(m)}\}_{m=1}^M \), from the following conditionals:

\[
\begin{align*}
    u_r^{(k)} &\sim \text{Dir}(a_r^{(k)} + s_{1,r}^{(k,t)}, \ldots, a_r^{(k)} + s_{n_k,r}^{(k,t)}) \\
p_r &\sim \text{Beta}(c \epsilon + s_r^{(t)}, c(1 - \epsilon) + g_r) \\
\lambda_r &\sim \text{Gamma}(g_r + s_r^{(t)}, p_r) 
\end{align*}
\] (3.13) (3.14) (3.15)

and either store the sample averages of \( u_r^{(k)}, p_r, \) and \( \lambda_r \), or their analytic means to use for the next CDF iteration (Guhaniyogi et al., 2014). Since the analytic means of the model parameters are available in closed-form in this case, we use the latter option, which obviates the need to draw \( M \) samples, thereby also speeding up the inference significantly.

We next describe the stochastic (online) VB inference for our model.
Stochastic Variational Inference

The batch VB inference (Section 3.4.2) requires using the entire data for the parameter updates in each iteration, which can be computationally expensive and can also result in slow convergence. Stochastic variational inference (SVI), on the other hand, leverages ideas from stochastic optimization (Hoffman et al., 2013) and, in each iteration, uses a small randomly chosen minibatch of the data to updates the parameters. Data from the current minibatch is used to compute stochastic gradients of the variational objective w.r.t. each of the parameters and these gradients are subsequently used in the parameter updates. For our model, the stochastic gradients depend on the sufficient statistics computed using the current minibatch $I_t$: $s^{(k,t)}_{j,r} = \sum_{i\in I_t : z_{ik} = j} \tilde{y}_{ir}$ and $s^{(t)}_r = \sum_{i\in I_t} \bar{y}_{i,r}$, where $\tilde{y}_{ir}$ is computed using Eq 3.10. Denoting $B$ as the minibatch size, we reweight these statistics by $\frac{N}{B}$ to compute the average sufficient statistics over the entire data (Hoffman et al., 2013) and update the other variational parameters as follows:

$$
\rho^{(k,t)}_{ikr} = (1 - \gamma_t)\rho^{(k,t-1)}_{ikr} + \gamma_t (a^{(k)} + (N/B)s^{(k,t)}_{ikr}) 
$$

$$
\rho^{(t)}_{ra} = (1 - \gamma_t)\rho^{(t-1)}_{ra} + \gamma_t (c_\epsilon + (N/B)s^{(t)}_r) 
$$

$$
\rho^{(t)}_{rb} = (1 - \gamma_t)\rho^{(t-1)}_{rb} + \gamma_t (c(1 - \epsilon) + g_r) 
$$

$$
\lambda^{(t)}_{ra} = (1 - \gamma_t)\lambda^{(t-1)}_{ra} + \gamma_t (g_r + (N/B)s^{(t)}_r) 
$$

$$
\lambda^{(t)}_{rb} = (1 - \gamma_t)\lambda^{(t-1)}_{ra} + \gamma_t p_r 
$$

where $\gamma_t$ is defined as $\gamma_t = (t_0 + t)^{-\kappa}$, $t_0 \geq 0$, and $\kappa \in (0.5, 1]$ is needed to guarantee convergence (Hoffman et al., 2013).

Computational Complexity:

In contrast to the batch Gibbs and batch VB, both of which have $O(NRK)$ cost per-iteration, the per-iteration cost of the online inference algorithms (CDF and SVI) is
$O(|I_t|RK)$ where $|I_t|$ is the minibatch size at round $t$. We use a fixed minibatch size $B$ for each minibatch, so the per-iteration cost is $O(BRK)$.

3.5 Related Work

Although tensor factorization methods have received considerable attention recently, relatively little work exists on scalable analysis of massive count-valued tensor data. Most of the recently proposed methods for scalable tensor decomposition (Kang et al., 2012; Beutel et al., 2014; Inah et al., 2015; Papalexakis et al., 2015) are based on minimizing the Frobenious norm of the tensor reconstruction error, which may not be suitable for count or overdispersed count data. The rank of decomposition also needs to be pre-specified, or chosen via cross-validation. Moreover, these methods assume the tensor to be fully observed and thus cannot be used for tensor completion tasks. Another key difference between these methods and ours is that scaling up these methods requires parallel or distributed computing infrastructure, whereas our fully Bayesian method exhibits excellent scalability on a single machine. At the same time, the simplicity of the inference update equations would allow our model to be easily parallelized or distributed. We leave this possibility to future work.

One of the first attempts to explicitly handle count data in the context of non-negative tensor factorization includes the work of (Chi and Kolda, 2012), which is now part of the Tensor Toolbox \(^1\). This method optimizes the Poisson likelihood, using an alternating Poission regression sub-routine, with non-negative constraints on the factor matrices. However, this method requires the rank of the decomposition to be specified, and cannot handle missing data. Due to its inability in handling missing data, for our experiments (Section 4.5), as a baseline, we implement and use a Bayesian version of this model which can handle missing data.

Among other works of tensor factorization for count data, the method in (Baz-

erque et al., 2013) can deal with missing values, though the rank still needs to be specified, and moreover the factor matrices are assumed to be real-valued, which makes it unsuitable for interpretability of the inferred factor matrices.

In addition to the Poisson non-negative tensor factorization method of (Chi and Kolda, 2012), some other non-negative tensor factorization methods (Shashua and Hazan, 2005; Cichocki et al., 2009; Schmidt and Mohamed, 2009) also provide interpretability for the factor matrices. However, these methods usually have one or more of the following limitations: (1) there is no explicit generative model for the count data, (2) the rank needs to be specified, and (3) the methods do not scale to the massive tensor data sets of scales considered in this work.

Methods that facilitate a full Bayesian analysis for massive count-valued tensors, which are becoming increasingly prevalent nowadays, are even fewer. A recent attempt on Bayesian analysis of count data using Poisson likelihood is considered in (Schein et al., 2014); however, unlike our model, their method cannot infer the rank and relies on batch VB inference, limiting its scaling behavior. Moreover, the Poisson likelihood may not be suitable for overdispersed counts.

Finally, inferring the rank of the tensor, which is NP-complete in general (Kolda and Bader, 2009), is another problem for which relatively little work exists. Recent attempts at inferring the rank of the tensor in the context of CP decomposition include (Rai et al., 2014; Zhao et al., 2015); however (1) these methods are not applicable for count data, and (2) the inferred factor matrices are real-valued, lacking the type of interpretability needed in many applications.

Our framework is similar in spirit to the matrix factorization setting proposed in (Zhou et al., 2012b) which turns out to be a special case of our framework. In addition, while (Zhou et al., 2012b) only developed (batch) Gibbs sampling based inference, we present both Gibbs sampling as well as variational Bayesian inference, and design efficient online Bayesian inference methods to scale up our framework for
handling massive real-world tensor data.

To summarize, in contrast to the existing methods for analyzing tensors, our fully Bayesian framework, based on a proper generative model, provides a flexible method for analyzing massive count-valued tensors, side-stepping crucial issues such as rank-specification, providing good interpretability of the latent factors, while still being scalable for analyzing massive real-world tensors via online Bayesian inference.

3.6 Experiments

We apply the proposed model on a synthetic and three real-world data sets that range in their sizes from moderate to medium to massive. The real-world tensor data sets we use in our experiments are from diverse application domains, such as analyzing country-country interaction data in political science, topic modeling on multiway publications data (with entities being authors, words, and publication venues), and analysis of massive household transactions data. These data sets include:

- **Synthetic Data**: This is a tensor of size $300 \times 300 \times 300$ generated using our model by setting an upper bound $R = 50$ over the number of factors; only 20 factors were significant (based on the values of $\lambda_r$), resulting in an effective rank 20.

- **Political Science Data (GDELT)**: This is a real-world four-way tensor data of country-country interactions. The data consists of 220 countries, 20 action types, and the interactions date back to 1979 (Leetaru and Schrodt, 2013). We focus on a subset of this data collected during the year 2011, resulting in a tensor of size $220 \times 220 \times 20 \times 52$. Section 3.6.4 provides further details.
• **Publications (Pub) Data:** This is a $2425 \times 9088 \times 4068$ count-valued tensor, constructed from a database of research papers published by researchers at Duke University\(^2\); the three tensor modes correspond to authors, words, and venues. Section 3.6.3 provides further details.

• **Transactions (Food) Data:** This is a $117054 \times 438 \times 67095$ count-valued tensor, constructed from a database of transactions data of food item purchases at various stores in the US\(^3\); the three tensor modes correspond to households, stores, and items. Section 3.6.5 provides further details.

### 3.6.1 Inferring the Rank

To begin with, as a sanity check for our model, we first perform an experiment on the synthetic data described above to see how well the model can recover the true rank (tensor completion results are presented separately in Section 3.6.2). For this experiment, we run the batch Gibbs sampler (the other inference methods also yield similar results) with 1000 burn-ins, and 1000 collection samples. We experiment with three settings: using 20%, 50% and 80% data for training. The empirical distribution (estimated using the collected MCMC samples) of the effective inferred rank for each of these settings is shown in Figure 3.1 (left). In each collection iteration, the effective rank is computed after a simple thresholding on the $\lambda_r$'s where components with very small $\lambda_r$ are not counted (also see Figure 3.1 (right)). With 80% training data, the distribution shows a distinct peak at 20 and even with smaller amounts of training data (20% and 50%), the inferred rank is fairly close to the ground truth of 20. In Figure 3.1 (right), we show the spectrum of all the $\lambda_r$'s comparing the ground truth

\(^2\) Obtained from [https://scholars.duke.edu/](https://scholars.duke.edu/)

\(^3\) Data provided by United States Department of Agriculture (USDA) under a Third Party Agreement with Information Resources, Inc. (IRI)
vs the inferred values.

![Figure 3.1](image_url): Distribution over inferred ranks for synthetic data (left), and $\lambda$ inferred using 80% training data (right).

### 3.6.2 Tensor Completion Results

We next experiment on the task of tensor completion, where for each method 95% of the data are used for training and the remaining 5% data is used as the heldout set (note that the data sets we use are extremely sparse in nature, with considerably less than 1% entries of the tensor being actually observed). The results are reported in Table 3.1 where we show the log likelihood and the mean-absolute error (MAE) in predicting the heldout data. Timing-comparison for the various batch and online inference methods is presented separately in Section 3.6.6.

For this experiment, we compare our BNBCP model (using the various inference methods) with (1) BAYESPTF - a fully Bayesian variant (we implemented it ourselves) of a state-of-the-art Poisson Tensor Factorization model originally proposed in (Chi and Kolda, 2012) (which cannot however handle missing data), and (2) LRANTD (Zhou et al., 2012a) which is an optimization based non-negative tensor decomposition method. As Table 3.1 shows, our methods achieve better log-likelihood and MAE as compared to these baselines. Moreover, among our batch and online
Bayesian inference methods, the online inference methods give competitive or better results as compared to their batch counterparts. In particular, the online MCMC method based on conditional density filtering (BNBCP-CDF) works the best across all the methods (please see Section 3.6.6 for a timing comparison).

Table 3.1: Loglikelihood and MAE comparison for different methods (the two base-lines, our model with batch inference, and our model with online inference) on four datasets. Note: lraNTD gave out-of-memory error on publications and food transactions data sets so we are unable to report its results on these data sets. We also only report the MAE for lraNTD, and not the log-likelihood, because it uses a Gaussian likelihood model for the data.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Toy</th>
<th>GDELT</th>
<th>Pub</th>
<th>Food</th>
<th>Toy</th>
<th>GDELT</th>
<th>Pub</th>
<th>Food</th>
</tr>
</thead>
<tbody>
<tr>
<td>BayesPTF</td>
<td>-107563</td>
<td>-4425695</td>
<td>-860808</td>
<td>-2425433</td>
<td>1.012</td>
<td>55.478</td>
<td>1.636</td>
<td>1.468</td>
</tr>
<tr>
<td>lraNTD</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>1.019</td>
<td>65.049</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BNBCP-Gibbs</td>
<td>-97580</td>
<td>-3079883</td>
<td>-619258</td>
<td>-2512112</td>
<td>0.989</td>
<td>45.436</td>
<td>1.565</td>
<td>1.459</td>
</tr>
<tr>
<td>BNBCP-VB</td>
<td>-99381</td>
<td>-2971769</td>
<td>-632224</td>
<td>-2533086</td>
<td>0.993</td>
<td>43.485</td>
<td>1.574</td>
<td>1.472</td>
</tr>
<tr>
<td>BNBCP-CDF</td>
<td>-95472</td>
<td>-2947309</td>
<td>-597817</td>
<td>-2403094</td>
<td>0.985</td>
<td>44.243</td>
<td>1.551</td>
<td>1.423</td>
</tr>
<tr>
<td>BNBCP-OnlineVB</td>
<td>-98446</td>
<td>-3169335</td>
<td>-660068</td>
<td>-2518996</td>
<td>0.989</td>
<td>46.188</td>
<td>1.601</td>
<td>1.461</td>
</tr>
</tbody>
</table>

3.6.3 Analyzing Publications Database

The next experiment is on a three-way tensor constructed from a scientific publications database. The data consist of abstracts from papers published by various researchers at Duke University. In addition to the paper abstract, the venue information for each paper is also available. The data collection contains 2425 authors, 9088 words (after removing stop-words), and 4068 venues which results in a $2425 \times 9088 \times 4068$ word-counts tensor, on which we run our model. As the output of the tensor decomposition, we get three factor matrices. Since the latent factors in our model are non-negative and sum to one, each latent factor can also be interpreted as a distribution over authors/words/venues, and consequently represents a “topic”. Therefore the three factor matrices inferred by our model for this data correspond

4 Data crawled from https://scholars.duke.edu/
to authors × topics, words × topics, and venue × topics, which we use to further analyze the data.

We apply the model BNBCP-CDF on this data (with $R = 200$) and using the inferred words × topics matrix, in Table 4.2 (left) we show the list of 10 most probable words in four factors/topics that seem to correspond to optics, genomics, machine learning & signal processing, and statistics. To show the topic representation across different departments, we present a histogram of departmental affiliations for 20 authors with highest probabilities in these four factors. We find that, for the genomics factor, the top authors (based on their topic scores) have affiliations related to biology which makes intuitive sense. Likewise, for the statistics factor, most of the top authors are from statistics and biostatistics departments. The top 20 authors in factors that correspond to optics and machine learning & signal processing, on the other hand, are from departments of electrical and computer engineering and/or computer science, etc.

Table 3.2: Most probable words in topics related to optics, genomics, machine learning/signal processing(ML/SP) and statistics (Stats), and top ranked venues in ML/SP community.

<table>
<thead>
<tr>
<th>Optics</th>
<th>Genomics</th>
<th>ML/SP</th>
<th>Stats</th>
<th>Top Venues in ML/SP</th>
</tr>
</thead>
<tbody>
<tr>
<td>gigapixel</td>
<td>gene</td>
<td>dictionary</td>
<td>model</td>
<td>ICASSP</td>
</tr>
<tr>
<td>microcamera</td>
<td>chromatin</td>
<td>sparsity</td>
<td>priors</td>
<td>IEEE trans. sig. proc.</td>
</tr>
<tr>
<td>cameras</td>
<td>occupancy</td>
<td>model</td>
<td>bayesian</td>
<td>ICML</td>
</tr>
<tr>
<td>aperture</td>
<td>centromere</td>
<td>bayesian</td>
<td>lasso</td>
<td>Siam j. img. sci.</td>
</tr>
<tr>
<td>lens</td>
<td>transcription</td>
<td>compressed</td>
<td>latent</td>
<td>IEEE t. img. proc.</td>
</tr>
<tr>
<td>multiscale</td>
<td>genome</td>
<td>compressive</td>
<td>inference</td>
<td>IEEE int. sym. biom. img.</td>
</tr>
<tr>
<td>optical system</td>
<td>sites</td>
<td>matrix</td>
<td>regression</td>
<td>NIPS</td>
</tr>
<tr>
<td>system</td>
<td>expression</td>
<td>denoising</td>
<td>sampler</td>
<td>IEEE t. wireless comm.</td>
</tr>
<tr>
<td>nanopores</td>
<td>sequence</td>
<td>gibbs</td>
<td>semiparamet.</td>
<td>IEEE w. sta. sig. proc.</td>
</tr>
<tr>
<td>metamaterial</td>
<td>vegfa</td>
<td>noise</td>
<td>nonparamet.</td>
<td>IEEE t. inf. theory</td>
</tr>
</tbody>
</table>

Similarly, using the inferred venues × topics matrix, we list the most likely venues for each topic. Due to space-limitations, here we only present the most likely venues
Figure 3.2: Histogram of affiliations for top 20 authors in factors related to machine learning/signal processing (top left) and statistics (top right), optics (bottom left), and genomics (bottom right).

in machine learning & signal processing factor/topic; the result is shown in Table 4.2 (right-most column). The result shows that venues like ICASSP, IEEE Trans. Signal Proc., ICML, and NIPS all rank at the top in the machine learning & signal processing factor, which again makes intuitive sense.

3.6.4 Analyzing Political Science Data

We use the model to analyze a real-world political science data set consisting of country-country interactions. Such analyses are typically done by political scientists to study, analyze and understand complex international multilateral relations among
countries. The data set is from the Global Database of Events, Location, and Tone (GDELT) (Leetaru and Schrodt, 2013). GDELT records the dyadic interactions between countries in the form of “Country A did something to Country B”. In our experiments, we consider 220 countries (“actors”) and 20 unique high-level action types in 52 weeks of year 2012. After preprocessing, we have a four-way (country-country-action-time) action counts tensor of size $220 \times 220 \times 20 \times 52$. Note that both first and second tensor mode represents countries; first mode as “sender” and the second mode as “receiver” of a particular action. In this analysis, we set $R$ to be large enough (200) and the model discovered roughly about 120 active components (i.e., components with significant value of $\lambda_r$).

![Graphs showing country factors and time factors for Julian Assange asylum in Ecuador and 2012 Benghazi attack](image)

**Figure 3.3:** Country factors (top row) and time factors (bottom row) for Julian Assange asylum in Ecuador (left column) and 2012 Benghazi attack (right column).

We apply the model (BNBCP-CDF; other methods yield similar results) and examine each of the time dimension factors, specifically looking for the significant components (based on the magnitude of $\lambda_r$) in which the time dimension factor also peaks during certain time(s) of the year. We show results with two such factors in Figure 3.3. In Figure 3.3 (column 1), the time and country (actor) factors seems
to suggest that this factor/topic corresponds to the event “Julian Assange”. The actor subplot shows spikes at Ecuador, United Kingdom, United States, and Sweden whereas the time factor in the bottom left subplot shows spikes between June and August. The time and countries involved are consistent with the public knowledge of the event of Julian Assange seeking refuge in Ecuador.

Likewise, in Figure 3.3 (column 2), the time and country (actor) factors seem to suggest that this factor corresponds to the event “Benghazi Attack” which took place on Sept. 12 (week 37) of 2012, in which Islamic militants attacked American diplomatic compound in Benghazi, Libya. The attack killed an US Ambassador. As the Figure shows, the top actors identified are US, Libya and Egypt, and spikes are found at around week 37 and 38, which are consistent with the public knowledge of this event.

The results of these analyses demonstrate that the interpretability of our model can be useful for identifying events or topics in such multiway interaction data.

3.6.5 Analyzing Transactions Data

We next apply our model (BNBCP-CDF; other methods yield similar results) for analyzing transactions data for food item purchases made at stores. Our data is collected for a demographically representative sample of US consumers who reside in large urban and suburban areas and purchase food in supermarkets and grocery stores. The data were provided by the USDA under a Third Party Agreement with IRI. Each transaction is identified by a unique Universal Product Code (UPC) barcode and the store where the transaction occurred. Some items such as fresh produce do not have UPCs and are identified separately. The households are observed over a four year period, during which they are provided with a technology that allows them to scan each purchase and record additional information such as the store where the purchase was made (and other economic data). Participating households are provided
with incentives designed to encourage compliance. For each household-product-store combination we record the number of unique purchases over the sampling period. The database has a total of 117,054 unique households, 438 stores, and 67,095 unique items and we construct a 3-way count tensor of size $117054 \times 438 \times 67095$ with about 6.2 million nonzero entries.

We apply the proposed model on this data by setting $R = 100$ (out of which about 60 components were inferred to have a significant value of $\lambda_r$) and looked at the stores factor matrix. Since each column (which sums to 1) of the store factor matrix can be thought of as a distribution over the stores, we look at three of the factors from the store factor matrix and tried to identify the stores that rank at the top in that factor. In Table 3.3, we show results from each of these factors. Factor 1 seems to suggest that it is about the most popular stores (included Walmart, for example), Factor 2 has stores that primarily deal in wholesale (e.g., Costco, Sam’s Wholesale Club), and Factor 3 contains stores that sell none or very few food items (e.g., Mobil, Petco). Note that the Walmart Super Center figures prominently in both Factor 1 and Factor 2.

Table 3.3: Three of the store factors inferred from the transaction data (top-5 stores shown for each)

<table>
<thead>
<tr>
<th>Factor 1</th>
<th>Factor 2</th>
<th>Factor 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walmart Sup. Center</td>
<td>Sam’s Club</td>
<td>Dick’s Sporting</td>
</tr>
<tr>
<td>Walmart Traders</td>
<td>Meijer</td>
<td>Mobil</td>
</tr>
<tr>
<td>Walmart Neighb.</td>
<td>Costco</td>
<td>Petco</td>
</tr>
<tr>
<td>Walmart</td>
<td>B J’S Wholesale</td>
<td>Sally Beauty</td>
</tr>
<tr>
<td>Kroger</td>
<td>Walmart Sup. Center</td>
<td>GNC All</td>
</tr>
</tbody>
</table>

We next look at the items factor matrix. In Figure 4.3, we plot the inferred distribution over items in each of the three clusters described above. For factors 1 and 2 (which correspond to the most popular stores and wholesale stores respectively), the distribution over the items (top and bottom panel in Figure 4.3) have a reasonably
significant mass over a certain range of items (for the items indexed towards the left side in the plots of factors 1 and 2). On the other hand, for factor 3 which corresponds to stores that sell no or very few types of food items, the distribution over the items is rather flat and diffuse with very weak intensities (looking at the scale on the y axis). From the Figure 4.3, it is also interesting to observe that the set of active items in factors (1 & 2) vs factor 3 seem to be mostly disjoint.

This analysis provides a first attempt to analyze food shopping patterns for American consumers on a large scale. As the world, at large, struggles with a combination of increasing obesity rates and food insecurity, this analysis shows that consumer preferences are densely clustered across both stores and items. This indicates that household tend to have fairly rigid preferences over the stores where they shop. Furthermore, they tend to consume a relatively small number of products from the
universe of available products. The concentration in both stores and products is indicative of limited search behavior and substantial behavioral rigidity which may be associated with suboptimal outcomes in terms of nutrition and health.

3.6.6 Scalability

We now perform an experiment comparing the proposed inference methods (batch and online) to assess their scalability (Figure 3.5). We first use the Transactions data (117054 \times 438 \times 67095) for this experiment. We would like to note that the state-of-the-art methods for count-valued tensor, such as the Poisson Tensor Factorization (PTF) method from the Tensor Toolbox (Chi and Kolda, 2012), are simply infeasible to run on this data because of storage explosion issue (the method requires expensive flattening operations of the tensor). The other baseline lrANTD (Zhou et al., 2012a) we used in our experiments was also infeasible to run on this data. We set \( R = 100 \) for each method (about 60 factors were found to be significant, based on the inferred values of the \( \lambda_r \)'s) and use a minibatch size of 100000 for all the online inference methods. For the conditional density filtering as well as stochastic variational inference, we set the learning rate as \( t_0 = 0 \) and \( \kappa = 0.5 \). Figure 3.5 shows that online inference methods (conditional density filtering and stochastic variational inference) converge much faster to a good solution than batch methods. This experiment shows that our online inference methods can be computationally viable alternatives if their batch counterparts are slow/infeasible to run on such data. We then perform another experiment on the Scholars data, on which the PTF method of (Chi and Kolda, 2012) was feasible to run and compare its per-iteration running time with our model (using both batch as well as online inference). Since PTF cannot handle missing data, for this experiment, each method was run with all the data. As Fig 3.6 shows, our methods have running times that are considerably smaller than that of PTF.
Figure 3.5: Time vs heldout log likelihoods with various methods on transactions data

Figure 3.6: Timing comparison of various methods on Scholars data
4.1 Introduction

With the recent surge in multiway, multirelational, or “tensor” data sets (Nickel et al., 2011; Kang et al., 2012), learning algorithms that can extract useful knowledge from such data are becoming increasingly important. Tensor decomposition methods (Kolda and Bader, 2009) offer an attractive way to accomplish this task. Among tensor data, of particular interest are real-world binary tensors, which are now ubiquitous in problems involving social networks, recommender systems, and knowledge bases, etc. For instance, in a knowledge base, predicate relations defined over the tuples (subjects, objects, verbs) can be represented in form of a binary three-way tensor (Kang et al., 2012).

Usually the real-world binary tensors are massive but extremely sparse. Many existing binary tensor factorization methods (Nickel et al., 2011; Xu et al., 2013; Rai et al., 2014) are based on probit or logistic likelihood. These methods scale poorly for massive binary tensors because these require evaluating the likelihood/loss-function
on both ones as well as zeros in the tensor.

It is therefore desirable to have methods that can perform efficient tensor decomposition for such data, ideally with a computational-complexity that depends only on the number of nonzeros (i.e., the ones) in the tensor, rather than the “volume” of the tensor. Motivated by this problem, we present a scalable Bayesian model for the Canonical PARAFAC (CP) tensor decomposition (Kolda and Bader, 2009), with an inference-complexity that scales linearly in the number of ones in the tensor. Our model uses a zero-truncated Poisson likelihood for each binary observation in the tensor; this obviates the evaluation of the likelihoods for the zero entries. At the same time, the significant speed-up is not at the cost of sacrificing on the quality of the solution. As our experimental results show, the proposed likelihood model yields comparable or better results to logistic likelihood based models, while being an order of magnitude faster in its running-time on real-world binary tensors. Note that replacing the zero-truncated Poisson by the standard Poisson makes our model also readily applicable for count-valued tensors (Chi and Kolda, 2012); although, in this exposition, we will focus exclusively on binary tensors.

Often, side-information (Acar et al., 2011; Beutel et al., 2014), e.g., pairwise relationships (partially/fully observed), may also be available for objects in some of the tensor dimensions. For example, in addition to a binary tensor representing authors × words × venues associations, the author × author co-authorship network may be available (at least for some pairs of authors). Such a network may be especially useful in “cold-start” settings where there is no data for some of the entities of a mode in the tensor (e.g., for some authors, there is no data in the tensor), but a network between entities in that mode may be available (See Fig 4.1 for an illustration). Our model allows leveraging such network(s), without a significant computational overhead, using the zero-truncated Poisson likelihood also to model these binary pairwise relationships.
To facilitate efficient fully Bayesian inference, we develop easy-to-implement batch as well as online MCMC inference; the latter is especially appealing for handling dense binary data, i.e., when the number of ones in the tensor and/or the network is also massive. Another appealing aspect about the model is its interpretability; a Dirichlet prior on the columns of each factor matrix naturally imposes non-negativity. In addition, the rank of decomposition can be inferred from the data.

4.2 Truncated Poisson Tensor Decomposition for Binary Data

The Canonical PARAFAC (CP) decomposition (Kolda and Bader, 2009) offers a way to express a tensor as a sum of rank-1 tensors. Each rank-1 tensor corresponds to a specific “factor” in the data. More specifically, the goal in CP decomposition is to decompose a tensor \( Y \) of size \( n_1 \times n_2 \times \cdots \times n_K \), with \( n_k \) denoting the size of \( Y \) along the \( k^{th} \) mode (or “way”) of the tensor, into a set of \( K \) factor matrices \( U^{(1)}, \ldots, U^{(K)} \) where \( U^{(k)} = [u^{(k)}_1, \ldots, u^{(k)}_R] \), \( k = \{1, \ldots, K\} \), denotes the \( n_k \times R \) factor matrix associated with mode \( k \).

In its most general form, CP decomposition expresses the tensor \( Y \) via a weighted sum of \( R \) rank-1 tensors as

\[
Y \sim f(\sum_{r=1}^{R} \lambda_r u_r^{(1)} \odot \cdots \odot u_r^{(K)})
\]

(4.1)

In the above, the form of the link-function \( f \) depends on the type of data being modeled (e.g., \( f \) can be Gaussian for real-valued, Bernoulli-logistic for binary-valued, Poisson for count-valued tensors). Here \( \lambda_r \) is the weight associated with the \( r^{th} \) rank-1 component, the \( n_k \times 1 \) column vector \( u_r^{(k)} \) represents the \( r^{th} \) latent factor of mode \( k \), and \( \odot \) denotes vector outer product.

We use subscript \( i = \{i_1, \ldots, i_K\} \) to denote the \( K \)-dimensional index of the \( i \)-th entry in the tensor \( Y \). Using this notation, the \( i \)-th entry of the tensor \( Y \) can be
Our focus in this section is on developing a probabilistic, fully Bayesian method for scalable low-rank decomposition of massive binary tensors. As opposed to tensor decomposition models based on the logistic likelihood for binary data (Xu et al., 2013; Rai et al., 2014), which require evaluation of the likelihood for both ones as well as zeros in the tensor, and thus can be computationally infeasible to run on massive binary tensors, our proposed model only requires the likelihood evaluations on the nonzero (i.e., the ones) entries in the tensor, and can therefore easily scale to massive binary tensors. Our model is applicable to tensors of any order $K \geq 2$ (the case $K = 2$ being a binary matrix).

Our model is based on a decomposition of the form given in Eq. 4.1; however, instead of using a Bernoulli-logistic link $f$ to generate each binary observation $y_i$ in $Y$, we assume an additional layer (Eq. 4.2) which takes a latent count-valued $y_i$ in $Y$ and thresholds this latent count at one to generate the actual binary-valued entry $b_i$ in the observed binary tensor, which we will denote by $B$:

$$b_i = 1(y_i \geq 1)$$  \hspace{1cm} (4.2)

$$Y \sim \text{Pois}(\sum_{r=1}^{R} \lambda_r \cdot u^{(1)}_r \odot \ldots \odot u^{(K)}_r)$$  \hspace{1cm} (4.3)

$$u^{(k)}_r \sim \text{Dir}(a^{(k)}, \ldots, a^{(k)})$$  \hspace{1cm} (4.4)

$$\lambda_r \sim \text{Gamma}(g_r, \frac{p_r}{1 - p_r})$$  \hspace{1cm} (4.5)

$$p_r \sim \text{Beta}(\epsilon, c(1 - \epsilon))$$  \hspace{1cm} (4.6)

Marginalizing out $y_i$ from Eq. 4.2 leads to the following (equivalent) likelihood model

$$b_i \sim \text{Bernoulli}(1 - \exp(-\sum_{r=1}^{R} \lambda_r \prod_{k=1}^{K} u^{(k)}_{i,r}))$$  \hspace{1cm} (4.7)

Note that the thresholding in (4.2) looks similar to a probit model for binary data.
(which however thresholds a normal at zero); however, the probit model (just like the logistic model) also needs to evaluate the likelihood at the zeros, and can therefore be slow on massive binary data with lots of zeros. Likelihood models of the form (Eq. 4.7) have previously also been considered in work on statistical models of undirected networks (Morup et al., 2011; Zhou, 2015). Interestingly, the form of the likelihood in 4.7 also resembles the complementary log-log function (Collett, 2002; Piegorsch, 1992), which is known to be a better model for imbalanced binary data than the logistic or probit likelihood, making it ideal for handling sparse binary tensors.

The conditional posterior of the latent count $y_i$ is given by

$$y_i | b_i, \lambda, \{u^{(k)}_{ik}\}_{k=1}^K \sim b_i \cdot \text{Pois}_+ \left( \sum_{r=1}^R \lambda_r \prod_{k=1}^K u^{(k)}_{ik} \right)$$

(4.8)

where $\text{Pois}_+ (\cdot)$ is zero truncated Poisson distribution. Eq. (4.8) suggests that if $b_i = 0$, then $y_i = 0$ almost surely with probability one, which can lead to significant computational savings, if the tensor has a large number of zeros. In addition, our model also enables leveraging a reparameterization (Section 4.2.2) of the Poisson distribution in terms of a multinomial, which allows us to obtain very simple Gibbs-sampling updates for the model parameters.

Note that the Dirichlet prior on the latent factors $u^{(k)}_r$ naturally imposes non-negativity constraints (Chi and Kolda, 2012) on the factor matrices $U^{(1)}, \ldots, U^{(K)}$. Moreover, since the columns $u^{(k)}_r$ of these factor matrices sums to 1, each $u^{(k)}_r$ can also be interpreted as a distribution (e.g., a “topic”) over the $n_k$ entities in mode $k$. Furthermore, the gamma-beta hierarchical construction (Zhou et al., 2012b) of $\lambda_r$ (Eq 4.5 and 4.6) allows inferring the rank of the tensor by setting an upper bound $R$ on the number of factors and inferring the appropriate number of factors by shrinking the coefficients $\lambda_r$’s to close to zero for the irrelevant factors. These
aspects make our model interpretable as well as provide it the ability to do model selection (i.e., inferring the rank), in addition to being computationally efficient by focusing the computations only on the nonzero entries in the tensor $B$.

### 4.2.1 Leverage Mode Networks

Often, in addition to the binary tensor $B$, pairwise relationships between entities in one or more tensor modes may be available in form of a symmetric binary network or an undirected graph. Leveraging such forms of side-information can be beneficial for tensor decomposition, especially if the amount of missing data in the main tensor $B$ is very high (Acar et al., 2011; Beutel et al., 2014; Rai et al., 2015a), and, even more importantly, in “cold-start” settings, where there is no data in the tensor for entities along some of the tensor mode(s), as shown in Fig 4.1. In the absence of any side-information, the posterior distribution of the latent factors $u_{i(k)}$ of such entities in that tensor mode would be the same as the prior (i.e., just a random draw). Leveraging the side-information (e.g., a network) helps avoid this.

**Figure 4.1:** Binary tensor with an associated binary network between objects in mode-1 of the tensor (in general, network for other modes may also be available). In the “cold-start” setting as shown above, data along some of the tensor dimensions will be completely missing.

For entities of the $k$-th mode of tensor $B$, we assume a symmetric binary network $A^{(k)} \in \{0, 1\}^{n_k \times n_k}$, where $A^{(k)}_{i_k j_k}$ denotes the relationship between mode-$k$ entities $i_k$
and \(j_k\).

Just like our tensor decomposition model, we model the mode-\(k\) network \(A^{(k)}\) as a weighted sum of rank-1 symmetric matrices, with a similar likelihood model as we use for the tensor observations. In particular, we assume a latent count \(X_{i_kj_k}^{(k)}\) for each binary entry \(A_{i_kj_k}^{(k)}\), and threshold it at one to generate \(A_{i_kj_k}^{(k)}\)

\[
A_{i_kj_k}^{(k)} = 1(X_{i_kj_k}^{(k)} \geq 1) \quad (4.9)
\]

\[
X^{(k)} \sim \text{Pois}(\sum_{r=1}^{R} \beta_r \cdot u_r^{(k)} \otimes u_r^{(k)}) \quad (4.10)
\]

\[
\beta_r \sim \text{Gamma}(f_r, \frac{h_r}{1-h_r}) \quad (4.11)
\]

\[
h_r \sim \text{Beta}(d, d(1-d)) \quad (4.12)
\]

Note that since \(A^{(k)}\) is symmetric, only the upper (or lower) triangular portion needs to be considered, and moreover, just like in the case of the tensor \(B\), due to the truncated Poisson construction, the likelihood at only the nonzero entries needs to be evaluated for this part as well.

4.2.2 Reparameterized Poisson Draws

To simplify posterior inference (Section 5.3), we make use of two re-parameterizations of a Poisson draw (Zhou et al., 2012b). The first parameterization is to express each latent count variable \(y_i^{(k)}\) and \(X_{i_kj_k}^{(k)}\) as a sum of another set of \(R\) latent counts \(\{\tilde{y}_{ir}^{(k)}\}_{r=1}^{R}\) and \(\{\tilde{X}_{i_kj_kr}^{(k)}\}_{r=1}^{R}\), respectively

\[
y_i^{(k)} = \sum_{r=1}^{R} \tilde{y}_{ir}^{(k)}, \quad \tilde{y}_{ir}^{(k)} \sim \text{Pois}(\lambda_r \prod_{k=1}^{K} u_{ikr}^{(k)}) \quad (4.13)
\]

\[
X_{i_kj_k}^{(k)} = \sum_{r=1}^{R} \tilde{X}_{i_kj_kr}^{(k)}, \quad \tilde{X}_{i_kj_kr}^{(k)} \sim \text{Pois}(\beta_r u_{i_kr}^{(k)} u_{jkr}^{(k)}) \quad (4.14)
\]
The second parameterization assumes that the latent counts \( \tilde{y}_i \) and \( \tilde{X}^{(k)}_{ikjr} \) are drawn from a multinomial

\[
\tilde{y}_i, \ldots, \tilde{y}_R \sim \text{Mult}(y_i; \zeta_1, \ldots, \zeta_R)
\]

\[
\zeta_{ir} = \frac{\lambda_r \prod_{k=1}^{K} \gamma_{kr}^{(k)}}{\sum_{r=1}^{R} \lambda_r \prod_{k=1}^{K} \gamma_{kr}^{(k)}}
\]

\[
\tilde{X}^{(k)}_{ikjr1}, \ldots, \tilde{X}^{(k)}_{ikjR} \sim \text{Mult}(X_{ikjr}^{(k)}; \kappa_{ikjr1}^{(k)}, \ldots, \kappa_{ikjR}^{(k)})
\]

\[
\kappa_{ikjr}^{(k)} = \frac{\beta_r \gamma_{ikr}^{(k)} \gamma_{jr}^{(k)}}{\sum_{r=1}^{R} \beta_r \gamma_{ikr}^{(k)} \gamma_{jr}^{(k)}}
\]

As we show in Section 5.3, these parameterizations enable us to exploit the gamma-Poisson as well as the Dirichlet-multinomial conjugacy to derive simple, closed-form Gibbs sampling updates for the model parameters.

### 4.3 Inference

Exact inference in the model is intractable and we resort to Markov Chain Monte Carlo (MCMC) (Andrieu et al., 2003) inference. In particular, the reparameterization discussed in Section 4.2.2 allows us to derive simple Gibbs sampling updates for all the latent variables, except for the latent counts \( y_i \), which are drawn from a truncated Poisson distribution via rejection sampling. As discussed earlier, the computational cost for our inference method scales linearly w.r.t. the number of ones in the tensor (plus the number of nonzeros in the network, if side-information is used). This makes our method an order of magnitude faster than models based on logistic or probit likelihood for binary data (Rai et al., 2014; Xu et al., 2013), without sacrificing on the quality of the results. The relative speed-up depends on the ratio of total volume of the tensor to the number of ones, which is given by \( \prod_{k=1}^{K} n_k / \text{nnz}(B) \); here \( \text{nnz}(B) \) denotes the number of nonzeros in the tensor.
In this section, we present both batch MCMC (Section 4.3.1) as well as an online MCMC (Section 4.3.2) method for inference in our model. The online MCMC algorithm is based on the idea of Bayesian Conditional Density Filtering (CDF) (Guhaniyogi et al., 2014), and can lead to further speed-ups over the batch MCMC if the number of nonzeros in the tensor is also massive. The CDF algorithm provides an efficient way to perform online MCMC sampling using surrogate conditional sufficient statistics (Guhaniyogi et al., 2014).

For both batch MCMC and CDF based online MCMC, we provide the update equations, with and without the side-information, i.e., the mode network(s). For what follows, we define four quantities: \( s_{j,r}^{(k)} \), \( s_r \), \( v_{k,r} \), \( v_r \), which denote aggregates computed using the latent counts \( \tilde{y}_{ir} \) and \( \tilde{X}_{ik,kr}^{(k)} \). These quantities will be used at various places in the description of the inference algorithms that we present here.

4.3.1 Batch MCMC Inference

Tensor without Mode Network(s)

**Sampling \( y_i \):** For each observation \( b_i \) in the tensor, the latent count \( y_i \) is sampled as

\[
y_i \sim b_i \cdot \text{Pois}_+(\sum_{r=1}^{R} \lambda_r \prod_{k=1}^{K} u_{ikr}^{(k)})
\]  

(4.17)

where \( \text{Pois}_+(-) \) is zero truncated Poisson distribution. Eq. (4.17) suggests that if \( b_i = 0 \), then \( y_i = 0 \) almost surely; and if \( b_i = 1 \), then \( y_i \sim \text{Pois}_+ (\sum_{r=1}^{R} \lambda_r \prod_{k=1}^{K} u_{ikr}^{(k)}) \). Therefore the \( y_i \)'s only need to be sampled for the nonzero \( b_i \)'s.

**Sampling \( \tilde{y}_{ir} \):** The latent counts \( \{\tilde{y}_{ir}\} \) are sampled from a multinomial as Eq. (4.15). Note that this also needs to be done only for the nonzero \( b_i \)'s.

**Sampling \( u_{ir}^{(k)} \):** The columns of each factor matrix have a Dirichlet posterior, and are sampled as
\[ \mathbf{u}^{(k)}_r \sim \operatorname{Dir}(a^{(k)} + s^{(k)}_{1,r}, a^{(k)} + s^{(k)}_{2,r}, \ldots, a^{(k)} + s^{(k)}_{n^{(k)}_{i_k}}) \]  

**Sampling** \( p_r \): Using the fact that \( s_r = \sum_i \tilde{y}_{i,r} \) and marginalizing over the \( u^{(k)}_{i_k} \)'s in (4.13), we have \( s_r \sim \operatorname{Pois}(\lambda_r) \). Using this, along with (4.5), we can express \( s_r \) using a negative binomial distribution, i.e., \( s_r \sim \operatorname{NB}(g_r, p_r) \). Due to the conjugacy between negative binomial and beta, we can then sample \( p_r \) as

\[ p_r \sim \operatorname{Beta}(c\epsilon + s_r, c(1 - \epsilon) + g_r) \]  

**Sampling** \( \lambda_r \): Again using the fact that \( s_r \sim \operatorname{Pois}(\lambda_r) \) and (4.5), we have

\[ \lambda_r \sim \operatorname{Gamma}(g_r + s_r, p_r) \]  

As can be observed, when updating \( \mathbf{u}^{(k)}_r, p_r \) and \( \lambda_r \), the latent counts \( y_i \)'s and \( \tilde{y}_{i,r} \) corresponding to zero entries in \( \mathcal{B} \) are all equal to zero, and have no contribution to sufficient statistics \( s^{(k)}_{j,i} \) and \( s_r \). Therefore, only the nonzero entries in tensor need to be considered in the computations.

**Tensor with Mode Network(s)**

In the presence of mode network(s), the update equations for the latent variables \( p_r, \lambda_r, \tilde{y}_{i,r} \) and \( y_i \), that are associated solely with the binary tensor \( \mathcal{B} \), remain unchanged, and can be sampled as described in Section 4.3.1. We however need to sample the additional latent variables associated with mode-\( k \) network \( \mathbf{A}^{(k)} \), and the latent factors \( \mathbf{u}^{(k)}_r \) of mode-\( k \) that are shared by the binary tensor \( \mathcal{B} \) as well as the mode-\( k \) network.

**Sampling** \( X^{(k)}_{i_k,j_k} \): The latent counts \( X^{(k)}_{i_k,j_k} \) are sampled as

\[ X^{(k)}_{i_k,j_k} \sim A^{(k)}_{i_k,j_k} \cdot \operatorname{Pois} \left( \sum_{r=1}^{R} \beta_r u^{(k)}_{i_k,r} u^{(k)}_{j_k,r} \right) \]  

This only needs to be done for the nonzero entries in \( \mathbf{A}^{(k)} \).
Sampling $\tilde{X}_{ikjr}$: The latent counts $\tilde{X}_{ikjr}$ are sampled from a multinomial as equation (4.16). This also only needs to be done for the nonzero entries in $A^{(k)}$.

Sampling $u_{r}^{(k)}$: The columns of each factor matrix have a Dirichlet posterior, and are sampled as

$$u_{r}^{(k)} \sim \text{Dir}(a^{(k)} + s_{1,r}^{(k)} + v_{1,r}, \ldots, a^{(k)} + s_{n_{k},r}^{(k)} + v_{n_{k},r})$$  \hspace{1cm} (4.22)

Note that in the absence of the mode-$k$ network, the terms $v_{r}$ go away and Eq. 4.22 simply reduces to Eq. 4.18.

Sampling $h_r$: $h_r \sim \text{Beta}(d\alpha + v_r, d(1 - \alpha) + f_r)$.

Sampling $\beta_r$: $\beta_r \sim \text{Gamma}(f_r + v_r, h_r)$.

**Per-iteration Time-complexity**

For the binary tensor $B$, computing each $\zeta_{ir}$ (Eq. 4.15) takes $O(K)$ time and therefore computing all the $\{\zeta_{ir}\}$ takes $O(\text{nnz}(B)RK)$ time. Likewise, for the binary mode-$k$ network $A^{(k)}$, computing all the $\{K_{ikjr}^{(k)}\}$ (Eq. 4.16) takes $O(\text{nnz}(A^{(k)})R)$ time. These are the most dominant computations in each iteration of our MCMC procedure; updating each $u_{r}^{(k)}$ takes $O(n_k)$ time and updating $\{p_r, h_r\}_{r=1}^{R}$ and $\{\lambda_r, \beta_r\}_{r=1}^{R}$ takes $O(R)$ time each. Therefore, the per-iteration time-complexity of our batch MCMC method is $O(\text{nnz}(B)RK + \text{nnz}(A^{(k)})R)$. The linear dependence on $\text{nnz}(B)$, $\text{nnz}(A^{(k)})$, $R$ and $K$ suggests that even massive, sparse binary tensors and mode network(s) can be handled easily even by our simple batch MCMC implementation. Also note that our model scales linearly even w.r.t. $R$, unlike most other methods (Ermis and Bouchard, 2014; Rai et al., 2014) that have quadratic dependence on $R$.

The above computations can be further accelerated using a distributed/multi-core setting; we leave this for future work. In Section 4.3.2, however, we present an online MCMC method based on the idea of Bayesian Conditional Density Filtering (Guhaniyogi et al., 2014), which leads to further speed-ups, even in single-machine
4.3.2 Online MCMC Inference

We develop an efficient online MCMC sampler for the model, leveraging ideas from the Conditional Density Filtering (CDF) (Guhaniyogi et al., 2014). The CDF algorithm for our model selects a minibatch of the tensor (and mode network, if the side-information is available) entries at each iteration, samples the model parameters from the posterior, and updates the sufficient statistics \( s_{j,r}^{(k)}, s_r, v_{ik,r} \) and \( v_r \) using the data from the current minibatch.

Tensor without Mode Network(s)

We first provide the update equations for the case when there is no side-information (mode network). Denote \( I_t \) as indices of entries of tensor \( B \) from the minibatch selected at iteration \( t \). The CDF algorithm at iteration \( t \) proceeds as:

**Sampling \( y_i \):** For all \( i \in I_t \), sample \( y_i \) according to equation (4.17); like in the batch MCMC case, the sampling only needs to be done for the nonzero \( b_i \)’s.

**Sampling \( \tilde{y}_{ir} \):** For all \( i \in I_t \), sample the latent counts \( \tilde{y}_{ir(\in I_t)} \) using (4.15), again only for the nonzero \( b_i \)’s.

**Updating the conditional sufficient statistics:** Update the conditional sufficient statistics \( s_{j,r}^{(k)} \) as \( s_{j,r}^{(k,t)} = s_{j,r}^{(k,t-1)} + \sum_{i \in I_t: i_k = j} \tilde{y}_{ir} \) and update \( s_r \) as \( s_r^{(t)} = s_r^{(t-1)} + \sum_{i \in I_t} \tilde{y}_{ir} \). These updates basically add to the old sufficient statistics, the contributions from the data in the current minibatch. In practice, we also reweight these sufficient statistics by the ratio of the total number of ones in \( B \) and the minibatch size, so that they represent the average statistics over the entire tensor. This reweighting is akin to the way average gradients are computed in stochastic variational inference methods (Hoffman et al., 2013).

**Updating \( u_r^{(k)}, p_r, \lambda_r \):** Draw \( M \) samples \( \{u_r^{(k,m)}, p_r^{(m)}, \lambda_r^{(m)}\}_{m=1}^M \) using the follow-
ing conditionals
\[
\mathbf{u}^{(k)}_r \sim \text{Dir}(a^{(k)} + s^{(k,t)}_{1,r}, \ldots, a^{(k)} + s^{(k,t)}_{n_k,r})
\]
(4.23)
\[
p_r \sim \text{Beta}(c\epsilon + s^{(t)}_{r}, c(1 - \epsilon) + g_r)
\]
(4.24)
\[
\lambda_r \sim \text{Gamma}(g_r + s^{(t)}_{r}, p_r)
\]
(4.25)
and either store the sample averages of \(\mathbf{u}^{(k)}_r, p_r,\) and \(\lambda_r,\) or their analytic means to use for the next CDF iteration (Guhaniyogi et al., 2014).

Tensor with Mode Network(s)

For all the latent variables associated solely with the tensor \(\mathbf{B},\) the sampling equations for the CDF algorithm in the presence of mode network(s) remain unchanged as the previous case with no network. In the presence of the mode network, the additional latent variables include the sufficient statistics \(v_{i_k,r}\) and \(v_r,\) and these need to be updated in each CDF iteration.

Denote \(J_t\) as indices of entries selected from the mode-\(k\) network \(\mathbf{A}^{(k)}\) in iteration \(t.\) The update equations for the latent variables that depend on \(\mathbf{A}^{(k)}\) are as follows:

**Sampling** \(X_{i_k,j_k}:\) For \((i_k, j_k) \in J_t,\) latent count \(X_{i_k,j_k}\) is sampled using Eq. (4.21).

**Sampling** \(\tilde{X}_{i_k,j_k,r}:\) For \((i_k, j_k) \in J_t,\) latent counts \(\tilde{X}_{i_k,j_k,r}\) are sampled from a multinomial using Eq. (4.16).

**Updating the conditional sufficient statistics:** Update the sufficient statistics associated with the mode-\(k\) network as \(v^{(t)}_{i_k,r} = v^{(t-1)}_{i_k,r} + \sum_{j_k,(i_k,j_k) \in J_t} \tilde{X}_{i_k,j_k,r}\) and \(v^{(t)}_r = v^{(t-1)}_{r} + \sum_{i_k} \sum_{j_k,(i_k,j_k) \in J_t} \tilde{X}_{i_k,j_k,r}.\) Just like the way we update the tensor sufficient statistics \(s^{(k)}_{j,r}\) and \(s_r,\) we reweight these mode-\(k\) sufficient statistics by the ratio of the total number of ones in \(\mathbf{A}^{(k)}\) and the minibatch size, so that they represent the average statistics over the entire mode-\(k\) network.

**Updating** \(\mathbf{u}^{(k)}_r, p_r, \beta_r,:\) Using the following conditionals, draw \(M\) samples \(\{\mathbf{u}^{(k,m)}_r, h^{(m)}_r, \beta^{(m)}_r\}_{m=1}^M.\) We draw \(\mathbf{u}^{(k)}_r \sim \text{Dir}(a^{(k)} + s^{(k,t)}_{1,r} + v^{(t)}_{1,r}, \ldots, a^{(k)} + s^{(k,t)}_{n_k,r} + v^{(t)}_{n_k,r}).\)
and $h_r$ and $\beta_r$ as

$$
h_r \sim \text{Beta}(d\alpha + v_r^{(t)}, d(1 - \alpha) + f_r)$$

$$
\beta_r \sim \text{Gamma}(f_r + v_r^{(t)}, h_r)
$$

and either store the sample averages of $u_r^{(k)}$, $h_r$, $\beta_r$, or their analytic means to use for the next CDF iteration.

**Per-iteration time-complexity**

The per-iteration time-complexity of the CDF based online MCMC is linear in the number of nonzeros in each minibatch (as opposed to the batch MCMC where it depends on the number of nonzeros in the *entire* tensor and network). Therefore the online MCMC is attractive for *dense* binary data, where the number of nonzeros in the tensor/network is also massive; using a big-enough minibatch size (that fits in the main memory and/or can be processed in each iteration in a reasonable amount of time), the online MCMC inference allows applying our model on such dense binary data as well, which may potentially have several billions of nonzero entries.

4.4 Related Work

With the increasing prevalence of structured databases, social networks, and (multi)relational data, tensor decomposition methods are becoming increasingly popular for extracting knowledge and doing predictive analytics on such data (Bordes et al., 2011; Nickel et al., 2012; Kang et al., 2012). As the size of these data sets continues to grow, there has been a pressing need to design tensor factorization methods that can scale to massive tensor data.

For low-rank factorization of *binary* tensors, methods based on logistic and probit likelihood for the binary data have been proposed (Jenatton et al., 2012; London et al., 2013; Rai et al., 2014; Xu et al., 2013). However, these methods are not suited
for massive binary tensors where the number of observations (which mostly consist of zeros, if the tensor is also sparse) could easily be millions or even billions (Inah et al., 2015). As a heuristic, these methods rely on subsampling (Rai et al., 2014) or partitioning the tensor (Zhe et al., 2015), to select a manageable number entries before performing the tensor decomposition, or alternatively going for a distributed setting (Zhe et al., 2013).

In the context of tensor factorization, to the best of our knowledge, the only method (and one that is closest in spirit to our work) that scales linearly w.r.t. the number of ones in the tensor is (Ermis and Bouchard, 2014). Their work explored quadratic loss (and its variations) as a surrogate to the logistic loss and proposed a method (Quad-Approx) with a per-iteration complexity $O(\text{nnz}(B)R + R^2\sum_{k=1}^{K} n_k)$. Note that its dependence on $R$ is quadratic as opposed to our method which is also linear in $R$. They also proposed variations based on piecewise quadratic approximations; however, as reported in their experiments (Ermis and Bouchard, 2014), these variations were found to be about twice as slow than their basic Quad-Approx method (Ermis and Bouchard, 2014). Moreover, their methods (and the various other methods discussed in this section) have several other key differences from our proposed model: (1) our model naturally imposes non-negativity on the factor matrices; (2) $R$ can be inferred from data; (3) our method provides a fully Bayesian treatment; (4) in contrast to their method, which operates in a batch setting, the online MCMC inference allows our model to scale to even bigger problems, where the number of nonzeros could also be massive; and (5) our model also allows incorporating (fully or partially observed) mode-networks as a rich source of side-information.

In another recent work (Zhou, 2015), a similar zero-truncated Poisson construction, as ours, was proposed for edge-partitioning based network clustering, allowing the proposed model to scale in terms of the number of edges in the network. Our model, on the other hand, is more general and can be applied to multiway binary tensor data,
with an optionally available binary network as a potential source of side-information. Moreover, the Dirichlet prior on the factor matrices, its reparametrizations (Section 4.2.2), and the online MCMC inference lead to a highly scalable framework for tensor decomposition with side-information.

Another line of work on scaling up tensor factorization methods involves developing distributed and parallel methods (Kang et al., 2012; Inah et al., 2015; Papalexakis et al., 2012; Beutel et al., 2014). Most of these methods, however, have one or more of the following limitations: (1) these methods lack a proper generative model of the data, which is simply assumed to be real-valued and the optimization objective is based on minimizing the Frobenius norm of the tensor reconstruction error, which may not be suitable for binary data; (2) these methods usually assume a parallel or distributed setting, and therefore are not feasible to run on a single machine; (3) missing data cannot be easily handled/predicted; and (4) the rank of the decomposition needs to be chosen via cross-validation.

Leveraging sources of side-information for tensor factorization has also been gaining a lot of attention recently. However, most of these methods cannot scale easily to massive tensors (Acar et al., 2011; Rai et al., 2015a), or have to rely on parallel or distributed computing infrastructures (Beutel et al., 2014). In contrast, our model, by the virtue of its scalability that only depends on the number of nonzero entries in the tensor and/or the mode network, easily allows it to scale to massive binary tensors, with or without mode-network based side-information.

4.5 Experiments

We report experimental results for our model on a wide range of real-world binary tensors (with and without mode-network based side-information), and compare it with several baselines for binary tensor factorization. We use the following data sets for our experiments:
Table 4.1: Tensor completion accuracies in terms of AUC-ROC scores. Results are averaged over 10 splits of training and test data. Note: (1) Bayesian CP was infeasible to run on the Scholars and Facebook data; (2) Due to the lack of publicly available code for Quad-App and PQ-QuadApp, we only report its results on Kinship, UMLS, and MovieLens data (results taken from (Ermis and Bouchard, 2014)).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Kinship</th>
<th>UMLS</th>
<th>Movielens</th>
<th>DBLP</th>
<th>Scholars</th>
<th>FB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quad-App</td>
<td>0.819</td>
<td>0.821</td>
<td>0.851</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PW-QuadApp</td>
<td>0.921</td>
<td>0.939</td>
<td>0.950</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Bayesian-Logistic-CP</td>
<td><strong>0.987</strong></td>
<td><strong>0.997</strong></td>
<td>0.980</td>
<td>0.931</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ZTP-CP(Batch)</td>
<td>0.967</td>
<td>0.994</td>
<td><strong>0.990</strong></td>
<td><strong>0.976</strong></td>
<td><strong>0.996</strong></td>
<td>0.983</td>
</tr>
<tr>
<td>ZTP-CP(Online)</td>
<td>0.963</td>
<td>0.994</td>
<td>0.984</td>
<td>0.974</td>
<td><strong>0.996</strong></td>
<td><strong>0.984</strong></td>
</tr>
</tbody>
</table>

- **Kinship**: This is a binary tensor of size $104 \times 104 \times 26$, representing 26 types of relationships between 104 members of a tribe (Nickel et al., 2011). The tensor has about 3.8% nonzeros.

- **UMLS**: This is a binary tensor of size $135 \times 135 \times 49$ representing 56 types of verb relations between 135 high-level concepts (Nickel et al., 2011). The tensor has about 0.8% nonzeros.

- **Movielens**: This is a binary matrix (two-way tensor) of size $943 \times 1682$ representing the binary ratings (thumbs-up or thumbs-down) by 943 users on 1682 movies\(^1\). This data set has a total of 100,000 ones.

- **DBLP**: This is a binary tensor of size $10,000 \times 200 \times 10,000$ representing (author-conference-keyword) relations (Zhe et al., 2015). This tensor has only about 0.001% nonzeros, and is an ideal example of a massive but sparse binary tensor.

- **Scholars**: This is a binary tensor of size $2370 \times 8663 \times 4066$, constructed from a database of research paper abstracts published by researchers at Duke University; the three tensor modes correspond to authors, words, and publica-

\(^1\) [http://grouplens.org/datasets/movielens/](http://grouplens.org/datasets/movielens/)
tion venues, respectively. Just like the DBLP data, this tensor is also massive but extremely sparse with only about 0.002% nonzeros. In addition, the co-authorship network (i.e., who has written papers with whom) is also available, which we use as a source of side-information, and use this network to experiment with the cold-start setting (i.e., when the main tensor has no information about some authors).

- **Facebook**: The Facebook data is a binary tensor of size $63731 \times 63730 \times 1837$ with the three modes representing wall-owner, poster, and days (Papalexakis et al., 2013). This tensor has only $737498$ nonzeros. In addition to the binary tensor, the social network (friendship-links) between users is also given in form of a symmetric binary matrix of size $63731 \times 63731$, which has $1634180$ nonzeros.

  We use the network to experiment with the cold-start setting.

We use all the 6 data sets for the tensor completion experiments (Section 4.5.1). We also use the Scholars and Facebook data in the cold-start setting, where we experiment on the tensor completion task, leveraging the mode-network based side-information (Section 4.5.4).

The set of experiments we perform includes: (1) binary tensor completion (Section 4.5.1) using only the tensor data; (2) scalability behavior of our model (both batch as well as online MCMC) in terms of tensor completion accuracy vs runtime (Section 4.5.2); we compare our model with Bayesian CP based on logistic-likelihood (Rai et al., 2014); (3) a qualitative analysis of our results using a multiway topic modeling experiment (Section 4.5.3) on the Scholars data, with the entities being authors, words, and publication venues; and (4) leveraging the mode network for tensor completion in the cold-start setting (Section 4.5.4); for this experiment, we also demonstrate how leveraging the network leads to improved qualitative results in the multiway topic modeling problem.
In the experiments, we refer to our model as ZTP-CP (Zero-Truncated Poisson based CP decomposition). We compare ZTP-CP (using both batch MCMC as well as online MCMC inference) with the following baselines: (1) the quadratic loss minimization (Quad-App) proposed in (Ermis and Bouchard, 2014); (2) the refined piecewise quadratic approximation algorithm (PW-QuadApp) (Ermis and Bouchard, 2014); and (3) Bayesian CP decomposition based on logistic likelihood for binary data (Rai et al., 2014).

**Experimental settings:** All experiments are done on a standard desktop computer with Intel i7 3.4GHz processor and 24GB RAM. Unless specified otherwise, the MCMC inference was run for 1000 iterations with 500 burn-in iterations. The online MCMC algorithm was also run for the same number of iterations, with minibatch size equal to one-tenth of the number of nonzeros in the training data. For all the data sets, except Scholars and Facebook, we use $R = 20$ (also note that our model has the ability to prune the unnecessary factors by shrinking the corresponding $\lambda_r$ to zero). For Scholars and Facebook data, we set $R = 100$. The hyperparameters $g_r, f_r$ were set to 0.1, and $\epsilon$ and $\alpha$ are set to $1/R$, which worked well in our experiments.

### 4.5.1 Tensor Completion

In Table 4.1, we report the results on the tensor completion task (in terms of the AUC-ROC - the area under the ROC curve). For this experiment, although available, we do not use the mode network for the Scholars and the Facebook data; only the binary tensor is used (the results when also using the network are reported in Section 4.5.4). For each data set, we randomly select 90% of the tensor observations as the training data and evaluate each model on the remaining 10% observations used as the held-out data.

Since the code for Quad-App and PW-QuadApp baselines (both proposed in (Ermis and Bouchard, 2014)) is not publicly available, we are only able to report
the results for the Kinship, UMLS, and MovieLens data set (using the results reported in (Ermis and Bouchard, 2014)). For Bayesian CP (Rai et al., 2014), we use the code provided by the authors. Moreover, the Bayesian CP baseline was found infeasible to run on the Scholars and Facebook data (both of which are massive tensors), so we are unable to report those results. For fairness, on Kinship, UMLS, and MovieLens data, we use the same experimental settings for all the methods as used by (Ermis and Bouchard, 2014).

As shown in Table 4.1, our model outperforms Quad-App and PW-QuadApp in terms of the tensor-completion accuracies, and performs comparably or better than Bayesian CP, while being an order of magnitude faster (Section 4.5.2 shows the results on running times).

4.5.2 Scalability

We next compare our model with Bayesian CP (Rai et al., 2014) in terms of the running times vs tensor completion accuracy on Kinship and UMLS data sets. As shown in Fig. 4.2 (top-row), our model (batch as well as online MCMC) runs/converges an order of magnitude faster than Bayesian CP in terms of running time. On Scholars and Facebook, since Bayesian CP was infeasible to run, we are only able to show the results (Fig. 4.2, bottom-row) for our model, with batch MCMC and online MCMC inference. On all the data sets, the online MCMC runs/converges faster than the batch MCMC.

We would like to note that, although the model proposed in (Ermis and Bouchard, 2014) also scales linearly \(^2\) in the number of ones in the tensor, the per-iteration time-complexity of our model, which is linear in both \(\text{nnz}(B)\) as well as rank \(R\), is better than the model proposed in (Ermis and Bouchard, 2014) (which has quadratic complexity).

\(^2\) Although (Ermis and Bouchard, 2014) reported run times on Kinship and UMLS data sets, those number are not directly comparable with our run times reported here (due to possibly different machine configuration, which they do not specify in the paper).
dependence on $R$). Moreover, the tensor completion results of our model (shown in Table 4.1) on these data sets are better than the ones reported in (Ermis and Bouchard, 2014).

4.5.3 Multiway Topic Modeling

We also apply our model for a multiway topic modeling task on the Scholars data. The binary tensor represents authors $\times$ words $\times$ venues relationships. We apply our model (with batch MCMC) and examine the latent factors of each of the three dimensions. Since each factor is drawn from a Dirichlet, it is non-negative and naturally corresponds to a “topic”. In Table 4.2, after examining the words factor matrix,
we show the top-10 words for four of the factors (topics) inferred by our model; these factors seem to represent topics Evolutionary Biology, Medical Imaging, Machine Learning/Signal Processing, and Oncology. For the Machine Learning/Signal Processing topic, we also examine the corresponding topic in the venues factor matrix and show the top-10 venues in that topic (based on their factor scores in that factor). In Fig. 4.3, we also show the histograms of authors’ department affiliations for each of the four topics and the results make intuitive sense. The results in Table 4.2 and Fig. 4.3 demonstrate the usefulness of our model for scalable topic modeling of such multiway data.

4.5.4 Leveraging the Model Network

Finally, to investigate the usefulness of leveraging the mode network, we experiment with using both the tensor and the mode network on Scholars and Facebook data sets. For each data set, we report the AUC-ROC (area under the ROC curve) and AUC-PR (area under the precision-recall curve) on the tensor completion task, with and without network. For both data sets, we experiment with the more challenging...
cold-start setting. In particular, for the Facebook data, we hold out all the entries of the tensor slices after the first 50,000 wall-owners and predict those entries (using only the rest of the tensor, and using the rest of the tensor as well as the friendship network). We run the experiment with $R = 20$ and minibatch size of 50,000 for the online MCMC. The results in Table 4.3 show that using the network leads to better tensor completion accuracies.

We also perform a similar experiment on the Scholars data where we hold out all the entries in tensor slices after the first 1000 authors and predict those entries (using only the rest of the tensor, and using the rest of the tensor as well as the
co-authorship network). We run the experiment with $R = 100$ and minibatch size of 50,000 for the online MCMC. The results shown in Table 4.3 again demonstrate the benefit of using the network.

Table 4.3: Cold-start setting

<table>
<thead>
<tr>
<th></th>
<th>Facebook</th>
<th>Scholars</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC-ROC</td>
<td>AUC-PR</td>
</tr>
<tr>
<td>Without network</td>
<td>0.8897</td>
<td>0.6076</td>
</tr>
<tr>
<td>With network</td>
<td><strong>0.9075</strong></td>
<td><strong>0.7255</strong></td>
</tr>
</tbody>
</table>

In Fig. 4.4, we show another result demonstrating the benefit of using the co-authorship network for the Scholars data. Note that in the cold-start setting, there is no information in the tensor for the held-out authors. Therefore the topics associated with such authors are expected to be roughly uniformly random. As shown in Fig. 4.4 (left column), the set of held-out authors assigned to the topics medical imaging and oncology seem very random and arbitrary (we only show the aggregate department-affiliations). Using side-information (in form of the co-authorship network), however, the model sensibly assigns authors who are indeed related to these topics, as shown in right column of Fig. 4.4.
Figure 4.4: Histogram of the department-affiliations of the top 15 held-out authors associated with the factors of medical imaging (top) and oncology (bottom). The left column is obtained using no co-authorship information, and the right column is obtained using co-authorship information.
Learning from Multi-relational Data using Binary Tensor Factorization

5.1 Introduction

Multi-relation learning has drawn significant attentions in many applications, such as social network analysis and knowledge graph analysis. Multi-relational data usually define an entity by its relationship with other entities. Thus the data are in the form of entity-relation-entity triplets and can be represented in the form of a three-way binary sparse tensor $\mathbf{Y}$ of size $N \times N \times R$, where $N$ and $R$ denote the number of entities and relations, respectively. The $r$-th slice $\mathbf{Y}^r \in \{0, 1\}^{N \times N}$ corresponds to the $r$-th relation type where $\mathbf{Y}^r_{ij} = 1$ (denoting a positive example or a “valid” fact ) denotes that existence of the relationship of type $r$ between entities $i$ and $j$. On the other hand, $\mathbf{Y}^r_{ij} = 0$ means that this relationship is either known to be invalid, or is unknown. Subsequently, at places, we will refer to $\mathbf{Y}^r_{ij}$ as a fact (valid/invalid). The number of positives (valid facts) is typically much smaller than the number of negatives (invalid/unknown facts). For multi-relation learning, we are usually interested in two problems: (1) predicting the existence of unknown links
between entities; and (2) clustering entities and/or relations.

In this chapter, we present a scalable Bayesian framework for the problem of learning from multi-relational data. Our embedding-based framework defines each triplet/fact $Y_{ij}^r$ to be generated via a bilinear model of the form $p(Y_{ij}^r = 1|u_i, u_j, \Lambda^r) = f(u_i^\top \Lambda^r u_j)$. Here, $u_i, u_j$ denote $K$-dimensional embeddings of entities $i$ and $j$, respectively, $\Lambda^r$ is a $K \times K$ matrix which parameterizes the relation type $r$, and $f$ denotes the link function (defined subsequently in Sec. 5.2).

In contrast to the existing embedding-based approaches for multi-relational data (Socher et al., 2013; Bordes et al., 2013; Yang et al., 2014; Dong et al., 2014), an attractive aspect of our framework is that the learned embeddings naturally correspond to “topics” (each topic is a distribution over entities), which leads to improved interpretability and easy qualitative analyses, e.g., clustering entities/relations based on the topic(s) in which they are most “active”.

Another appealing aspect of the proposed framework is its computational scalability. Instead of the commonly employed squared loss or logistic/probit likelihood for modeling binary multi-relational data (Nickel et al., 2011; Sutskever et al., 2009; Jenatton et al., 2012; Socher et al., 2013; Bordes et al., 2013), we leverage the Bernoulli-Poisson likelihood model (Zhou, 2015; Hu et al., 2015b) for triplet generation (Sec. 5.2), which leads to an inference time-complexity that scales in the number of positive triplets in the data. Therefore our framework scales considerably better than the existing approaches for which the computational cost depends on the number of both positive and negative triplets (Nickel et al., 2011; Socher et al., 2013; Sutskever et al., 2009; Bordes et al., 2013; Yang et al., 2014; Dong et al., 2014). Moreover, as we discuss in Sec. 5.2.3, as compared to logistic/probit models, the Bernoulli-Poisson link function is also a more realistic model for imbalanced binary data, which is a characteristic exhibited in most real-world multi-relational data sets for which the number of positive triplets is far fewer than the number of negative triplets.
triplets.

To handle the potentially massive number of relations commonly encountered in modern multi-relational data sets, we also extend our model to allow sharing of parameters across multiple relations, which leads to a substantial reduction in the number of parameters to be learned and also makes the model more robust in cases where the number of observations available for each relation is very small. To accomplish this, we model each of the relation-specific parameter matrices \( \Lambda_r \) as a combination of a small set of “basis” relation matrices \( \Gamma_m \), where \( M \ll R \).

Finally, our fully Bayesian framework admits full local conjugacy, which allows deriving closed-form Gibbs sampling updates for all the model parameters. This, combined with the fact that the inference cost only depends on the number of positive triplets in the data, enable a fully Bayesian analysis for large-scale multi-relational data. We also develop an online inference algorithm that can process data in small minibatches and therefore can easily handle data sets that are too massive to deal with using batch algorithms.

\[
Y_{ij}^r = 1(X_{ij}^r \geq 1) \\
X_r \sim \text{Poisson}(U^T \Lambda U) \\
U_{r,k} \sim \text{Dirichlet}(a, \ldots, a) \quad \forall k = 1, \ldots, K \\
\Lambda_{r,k_1,k_2} \sim \begin{cases} 
\text{Gamma}(\epsilon d_{k_1}^r, \frac{1}{2}), & \text{if } k_1 = k_2 \\
\text{Gamma}(d_{k_1}^r, d_{k_2}^r, \frac{1}{2}), & \text{if } k_1 \neq k_2 
\end{cases} \\
d_{k_1}^r \sim \text{Gamma}(\gamma_0/K, 1/c_0) \\
\gamma^r \sim \text{Gamma}(\epsilon_0, 1/f_0)
\]

\[
Y_{ij} = 1(X_{ij}^r \geq 1) \\
X_r \sim \text{Poisson}(U^T \Lambda U^T) \\
U_{r,k} \sim \text{Dirichlet}(a, \ldots, a) \quad \forall k = 1, \ldots, K \\
\Lambda_{r,k_1,k_2} = \sum_{m=1}^M \eta_{mr} \Gamma_m^{r,k_1,k_2} \eta_{mr} \sim \text{Gamma}(h_0, 1/y_0) \\
\Gamma_m^{r,k_1,k_2} \sim \begin{cases} 
\text{Gamma}(\epsilon_m d_{k_1}^m, \frac{1}{2}), & \text{if } k_1 = k_2 \\
\text{Gamma}(d_{k_1}^m, d_{k_2}^m, \frac{1}{2}), & \text{if } k_1 \neq k_2 
\end{cases} \\
d_{k_1}^m \sim \text{Gamma}(\gamma_0/K, 1/c_0) \\
\epsilon_m \sim \text{Gamma}(\epsilon_0, 1/f_0)
\]

**Figure 5.1:** Left: Model-1 with each relation \( r \) having its own independent parameter matrix \( \Lambda_r \). Right: Model-2 with parameter sharing across relations via a set of basis matrices

### 5.2 Bayesian Non-negative Bilinear Factor Model

We first describe the basic setup of our Bayesian framework which is based on a bilinear non-negative latent factor model (Fig. 5.2) for multi-relational data, with
the property that leads to scaling in the number of positive triplets. Then, in Section 5.2.1, we describe in more detail our first model with its properties that lead to efficient, fully Bayesian inference. Subsequently, in Section 5.2.2, we will generalize the first model to allow further sharing of statistical strength across the parameters of multiple relations. In Section 5.2.3, we will also provide a justification of why both these models can more realistically model imbalanced binary data (very few positives), such as real-world multi-relational data sets.

![Figure 5.2: The basic setup of the bilinear latent factor model for multi-relational data](image)

One key aspect of both the proposed models is their departure from the standard logistic/probit link functions for binary-valued triplets, and the use of thresholded counts (Zhou, 2015; Hu et al., 2015b) to model the binary-valued triplets. Specifically, each binary-valued triplet $Y_{ij}^r$ is assumed generated by thresholding a latent count $X_{ij}^r$ at 1, where the latent count $X_{ij}^r$, in turn, is assumed drawn from a bilinear non-negative latent factor model

$$Y_{ij}^r = 1(X_{ij}^r \geq 1), \quad X_{ij}^r \sim \text{Poisson}(u_i^T \Lambda^r u_j) \quad (5.3)$$

Intuitively, for relation $r$, the strength of the interaction between entities $i$ and $j$ depends the score $u_i^T \Lambda^r u_j$, which defines the Poisson rate for $X_{ij}^r$. Marginalizing out $X_{ij}^r$ from Eq. 5.3, we have

$$Y_{ij}^r \sim \text{Bernoulli}(1 - e^{-u_i^T \Lambda^r u_j}) \quad (5.4)$$

80
Also note that the conditional posterior of the latent count $X_{ij}^r$ can be written as

$$(X_{ij}^r | Y_{ij}^r, u_i, \Lambda^r, u_j) \sim Y_{ij}^r \cdot \text{Poisson}_+ (u_i^T \Lambda^r u_j)$$  (5.5)

From Eq. 5.5, if $Y_{ij}^r = 0$ then $X_{ij}^r = 0$, almost surely (a.s.), and if $Y_{ij}^r = 1$ then $X_{ij}^r \sim \text{Poisson}_+ (u_i^T \Lambda^r u_j)$, a draw from zero-truncated Poisson. Therefore we only need to sample $X_{ij}^r$ if $Y_{ij}^r = 1$. We leverage this property in Section 5.3 to design scalable inference algorithms for our framework. We next describe both of our models which are based on this overall framework.

5.2.1 Model-1

The complete generative story for the first model, along with the prior distributions over the various model parameters is shown in Fig. 5.1 (left). The $N \times K$ matrix $U = [u_1 \ldots u_N]^T$ contains the $K$-dimensional embeddings of each of the $N$ entities. We further assume each $N$-dimensional column of $U$ is drawn from a Dirichlet, which can therefore be thought of as a distribution (or “topic”) over the $N$ entities (akin to a topic model). In contrast to the other embedding based models (Nickel et al., 2011; Sutskever et al., 2009; Jenatton et al., 2012; Socher et al., 2013; Bordes et al., 2013), this aspect of our model provides a nice interpretability to the entity embeddings, because each of the $K$ embedding coordinates of an entity can now be thought of as how “active” it is in each of the $K$ topics. This naturally allows to group/cluster the entities based on topics, without having to perform a separate step of running a clustering algorithm over the learned embeddings.

Although the model in Fig. 5.1 (left) is not originally conjugate, using the Poisson-multinomial equivalence (Dunson and Herring, 2005; Zhou et al., 2012b), we are able to develop a Gibbs sampler with closed-form sampling updates for all the model parameters. To see this, note that a Poisson distributed count-valued random variable can be expressed as a sum of Poisson distributed latent counts which, in turn,
can be generated by repeatedly sampling from a multinomial. To illustrate this in the context of our model, note that we can express each latent count $X_{r_{ij}} \sim \text{Poisson}(\sum_{k_1}^{K} \sum_{k_2}^{K} u_{ik_1} \Lambda_{r_{k_1 k_2}} u_{jk_2})$ as a sum of latent counts, i.e., $X_{r_{ij}} = \sum_{k_1}^{K} \sum_{k_2}^{K} X_{ik_1 k_2 j}$ where $X_{ik_1 k_2 j} \sim \text{Poisson}(u_{ik_1} \Lambda_{r_{k_1 k_2}} u_{jk_2})$, and then using the Poisson-multinomial equivalence, we have $\forall k_1, k_2$

$$\{X_{ik_1 k_2 j}\} \sim \text{Mult} \left( X_{r_{ij}}, \frac{\{u_{ik_1} \Lambda_{r_{k_1 k_2}} u_{jk_2}\}}{\sum_{k_1=1}^{K} \sum_{k_2=1}^{K} u_{ik_1} \Lambda_{r_{k_1 k_2}} u_{jk_2}} \right)$$

This, coupled with the multinomial-Dirichlet conjugacy, allows us to develop a Gibbs sampler for our model. Section 5.3 briefly describes the Gibbs sampler (both batch as well as a more efficient online version) and the Supplementary Material provides the additional details.

5.2.2 Model-2: Sharing Parameters Across Relations

Model-1 parametrizes each relation type $r$ by $\Lambda^r$, a $K \times K$ matrix. In real-world multi-relational data sets with potentially thousands of relation types, often many relation types may be similar to each other and therefore, instead of modeling each $\Lambda^r$ independently, it may be more appropriate to jointly model these in order to share statistical strength across relations. This significantly reduces the number of parameters that need to be learned and can also be helpful to handle the data sparsity problem, i.e., when the number of triplets observed per relation is very small. Our second model, with generative story shown in Fig. 5.1 (right), allows such a sharing by modeling the parameters $\Lambda^r$ of each relation type as a linear combination of $M$ “basis” relation parameter matrices $\{G^m\}_{m=1}^{M}$ shared by all the relations, i.e.,

$$\Lambda^r = \sum_{m=1}^{M} \eta_{mr} G^m$$
Fig. 5.3 illustrates this idea pictorially. If two relations \( r \) and \( r' \) are similar, their combination weight vectors \( \eta_r \) and \( \eta_{r'} \) are also expected to be similar. Note that, in this model, we can also view the combination weights \( \eta_r = [\eta_{1r}, \ldots, \eta_{Mr}] \in \mathbb{R}^M \) as an embedding of relation type \( r \). Therefore, unlike other bilinear models (Nickel et al., 2011; Sutskever et al., 2009), this model also provides a vector embedding for relations as well. Interestingly, this model structurally resembles a non-negative variant of Tucker tensor factorization model (Kolda and Bader, 2009) with \( U \) as the factor matrix of the entity dimension, \( \eta = [\eta_1, \ldots, \eta_R] \) as the factor matrix of the relation dimension, and \( \{G^m\}_{m=1}^M \) being the core tensor. Just like model-1, leveraging the Poisson-multinomial equivalent allows us to develop closed-form Gibbs sampling updates for all the model parameters (Section A.1 provides further details).

\[ \Lambda^R = \begin{bmatrix} K \end{bmatrix} \quad \Lambda^1 \quad \Lambda^2 \quad \Lambda^3 \]
\[ G^1 \quad G^2 \quad G^3 \quad G^4 \]
\[ R \quad M \quad \eta \]

**Figure 5.3:** Parameter sharing across relations for Model-2

### 5.2.3 Connection with the Complementary Log-log Link Function

It is interesting to note that the form of the likelihood function \( Y_{ij}^r \sim \text{Bernoulli}(1 - e^{-u_i^\top \Lambda^r u_j}) \), arising in both our models, resembles the complementary log-log (cloglog) function (Piegorsch, 1992; Collett, 2002) often used to model imbalanced binary data. In particular, the rate of growth of the function \( p(Y_{ij}^r = 1) = 1 - e^{-u_i^\top \Lambda^r u_j} \) (where \( u_i^\top \Lambda^r u_j \) is non-negative) along the \( Y \) axis from 0.5 to 1 tends to be much slower than the rate it drops from 0.5 to 0. Therefore, our generative model more realistically
captures the data imbalance in real-world multi-relational data sets that have very few positive triplets.

5.3 Inference

As discussed in Section 5.2.1, using data-augmentation and Poisson-multinomial equivalence (Dunson and Herring, 2005; Zhou et al., 2012b; Hu et al., 2015b,a), we are able to derive closed-form Gibbs sampling updates for all the model parameters. Note again that since we only need to sample the latent counts $\mathcal{X}^{r}_{ij}$ for the positive triplets, our sampling algorithms scale in the number of positive triplets, thereby leading to a very efficient inference. In Section 5.3.1 and A.1, we first describe the batch Gibbs sampling algorithms for model-1 and model-2, respectively, and then, in Section A.2, we briefly describe an online Gibbs sampler for both models (further details given in the Supplementary Material). In the rest of the exposition, we refer to the model-1 as BPBFM-1 and model-2 as BPBFM-2, where BPLFM denotes Bernoulli-Poisson Bilinear Factor Model to reflect our generative model.

5.3.1 Gibbs Sampling for BPBFM-1

Note that, for BPBFM-1 (generative model shown in Fig. 5.1-left), the latent count for each triplet $\mathcal{Y}^{r}_{ij}$ is defined as $\mathcal{X}^{r}_{ij} = \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \mathcal{X}^{r}_{ik_1k_2j}$. In what follows, we will also make use of the following quantities, defined in terms of the $\mathcal{X}^{r}_{ik_1k_2j}$’s:

\begin{align}
\mathcal{X}^{r}_{ik\ldots} &= \sum_{k_2=1}^{K} \sum_{j=1}^{N} \mathcal{X}^{r}_{ikk_2j} \quad (5.6) \\
\mathcal{X}^{r}_{i\ldots} &= \sum_{r=1}^{R} \mathcal{X}^{r}_{i\ldots} \quad (5.7) \\
\mathcal{X}^{r}_{k_1k_2\ldots} &= \sum_{i=1}^{N} \sum_{j=1}^{N} \mathcal{X}^{r}_{ik_1k_2j} \quad (5.8)
\end{align}

84
Sampling $X_{ij}^r$: For each triplet $Y_{ij}^r = 1$, the latent count $X_{ij}^r$ can be sampled as

$$X_{ij}^r \sim Y_{ij}^r \cdot \text{Poisson}(\sum_{k_1=1}^{K} \sum_{k_2=1}^{K} u_{ik_1} \Lambda_{k_1 k_2}^r u_{j k_2})$$  \hspace{1cm} (5.9)

Note that this only needs to be done for the observed triplets (i.e., if $Y_{ij}^r = 1$).

Sampling $X_{ik_1 k_2 j}^r$: Due to the Poisson-multinomial equivalence, $X_{ik_1 k_2 j}^r$ can be sampled as

$$\{X_{ik_1 k_2 j}^r\} \sim \text{Mult}(X_{ij}^r, \frac{\{u_{ik_1} \Lambda_{k_1 k_2}^r u_{j k_2}\}}{\sum_{k_1=1}^{K} \sum_{k_2=1}^{K} u_{ik_1} \Lambda_{k_1 k_2}^r u_{j k_2}})$$  \hspace{1cm} (5.10)

Sampling $U_{i,k}$: Using Dirichlet-multinomial conjugacy, each column of $U$ can be sampled as

$$U_{i,k} \sim \text{Dirichlet}(a + X_{1k}, a + X_{2k}, \ldots, a + X_{Nk})$$  \hspace{1cm} (5.11)

Sampling $d_k^r$: Using the additive property of the Poisson draws, we have

$$X_{k_1 k_2}^r \sim \text{Poisson}(\sum_{i=1}^{N} \sum_{j=1}^{N} u_{ik_1} \Lambda_{k_1 k_2}^r u_{j k_2})$$  \hspace{1cm} (5.12)

Marginalizing out $\Lambda_{k_1 k_2}^r$ from Eq.(5.12), we have

$$X_{k_1 k_2}^r \sim \text{NegBin}(\theta_{k_1 k_2}^{-\delta_{k_1 k_2}^r} d_k^r (d_k^r)^{-\delta_{k_1 k_2}^r}, p_{k_1 k_2})$$  \hspace{1cm} (5.13)

where $\delta_{k_1 k_2} = 1$ if $k_1 = k_2$, and $\delta_{k_1 k_2} = 0$ otherwise, and NegBin denotes the Negative Binomial distribution. In the above, $p_{k_1 k_2}$ is define as $p_{k_1 k_2} = \frac{\theta_{k_1 k_2}}{\theta_{k_1 k_2} + \beta}$ where $\theta_{k_1 k_2} = \sum_{i=1}^{N} \sum_{j=1}^{N} u_{ik_1} u_{j k_2}$. Using Eq.(5.13) and the data augmentation scheme proposed in ((Zhou et al., 2012b)), $d_k^r$ can be sampled by first sampling $\ell_{kk_2}^r \sim \text{Bernoulli}(\theta_{k_1 k_2}^r (d_k^r)^{-\delta_{k_1 k_2}^r})$ and then sampling $d_k^r$ as $d_k^r \sim \text{Gamma}(\frac{\gamma_0}{K} + \sum_{k_2=1}^{K} \ell_{kk_2}^r / (e^r)^{\delta_{k_1 k_2}^r} d_k^r (d_k^r)^{1-\delta_{k_1 k_2}^r} + 1, \ln(1-p_{kk_2}))$.
Sampling $\epsilon^r$: $\epsilon^r$ can be sampled as $\epsilon^r \sim \text{Gamma}(\epsilon_0 + \sum_{k=1}^{K} \ell_k^r, \frac{1}{f_0 - \sum_{k=1}^{K} d_k^r \ln(1 - p_k)})$.

Sampling $\Lambda_{k_1 k_2}$: Using Gamma-Poisson conjugacy, $\Lambda_{k_1 k_2}$ can be sampled as $\text{Gamma}((\epsilon^r)^\delta_{k_1 k_2} d_{k_1}^r (d_{k_2}^r)^{1-\delta_{k_1 k_2}} + \mathcal{X}_{k_1 k_2}^{r}, \frac{1}{\beta + \theta_{k_1 k_2}})$.

5.3.2 Gibbs Sampling for BPBFM-2

Proceeding in a manner similar to as we did for BPBFM-1, we can express each latent count $\mathcal{X}_{ij}^{r}$ in BPBFM-2 (which models each $\Lambda^r$ as $\sum_{m=1}^{M} \eta_{mr} G^m$) as a sum of the following form: $\mathcal{X}_{ij}^{r} = \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \sum_{m=1}^{M} \mathcal{X}_{ik_1 k_2 m j}^{r}$ where

$$\mathcal{X}_{ik_1 k_2 m j}^{r} \sim \text{Poisson}(u_{ik_1} \eta_{mr} G_{k_1 k_2}^m u_{jk_2})$$

We further define

$$\mathcal{X}_{k_1 k_2 m}^{r} = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{r=1}^{R} \mathcal{X}_{ik_1 k_2 m j}^{r}$$

$$\mathcal{X}_{...m}^{r} = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \mathcal{X}_{ik_1 k_2 m j}^{r}$$

Using additive property of Poisson distribution

$$\mathcal{X}_{k_1 k_2 m}^{r} \sim \text{Poisson}(\theta_{k_1 k_2} G_{k_1 k_2}^m G_{k_1 k_2}^m \sum_{r=1}^{R} \eta_{mr})$$  \hspace{1cm} (5.14)$$

$$\mathcal{X}_{...m}^{r} \sim \text{Poisson}(\eta_{mR} \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \theta_{k_1 k_2} G_{k_1 k_2}^m G_{k_1 k_2}^m)$$  \hspace{1cm} (5.15)$$

The Gibbs sampler updates for BPBFM-2 depend on these quantities. For brevity, we provide the detailed update equations in the Supplementary Material.

5.3.3 Online Gibbs Sampling

Although the Gibbs sampler we presented in the previous sections is efficient for moderate-sized data sets, when the number of observed triplets (and the number of
entities and/or relations), batch Gibbs sampling can be prohibitive to run and/or may have slow mixing. We therefore also develop online Gibbs sampling algorithms for both of our models. The proposed online Gibbs sampling algorithms for our models are based on the idea of the recently developed Bayesian Conditional Density Filtering (BCDF) framework (Guhaniyogi et al., 2014). They key idea in BCDF is to process data in small minibatches, and maintain and update sufficient statistics of the model parameters with each new minibatch of the data. In our models, these sufficient statistics are the latent counts. We briefly outline the online Gibbs sampler for BPBFM-1 below:

Denote $I_t$ as indices of valid triplets in minibatch at iteration $t$, and $I$ as the indices of all the valid triplets in training data. Define $\mathcal{X}_{ik_1k_2}^t = \frac{|I|}{|I_t|} \sum_{k_2=1}^{K} \sum_{j=1, i \in I_t}^{N} \mathcal{X}_{ik_1k_2j}^r$, $\mathcal{X}_{ik}^t = \frac{|I|}{|I_t|} \sum_{r=1}^{R} \mathcal{X}_{ik}^{r,t}$, and $\mathcal{X}_{k_1k_2}^r = \frac{|I|}{|I_t|} \sum_{i,j \in I_t} \mathcal{X}_{ik_1k_2j}^r$, where $|I|$ and $|I_t|$ are cardinalities of the two sets. Then similar to batch Gibbs Sampling, define following quantities for $t \leq 2$: $\mathcal{X}_{ik}^{r,t} = (1 - \rho)\mathcal{X}_{ik}^{r,t-1} + \rho \frac{|I|}{|I_t|} \sum_{k_2=1}^{K} \sum_{j=1, i \in I_t}^{N} \mathcal{X}_{ik_1k_2j}^r$, $\mathcal{X}_{ik}^t = (1 - \rho)\mathcal{X}_{ik}^{t-1} + \rho \frac{|I|}{|I_t|} \sum_{r=1}^{R} \mathcal{X}_{ik}^{r,t}$, and $\mathcal{X}_{k_1k_2}^r = (1 - \rho)\mathcal{X}_{k_1k_2}^{r,t-1} + \rho \frac{|I|}{|I_t|} \sum_{i,j \in I_t} \mathcal{X}_{ik_1k_2j}^r$. Here $\rho = (t + t_0)^{-w}$ is a decaying learning rate, as used in other online inference algorithms, such as stochastic variational inference (Hoffman et al., 2013). Here, $t_0 > 0$ and $w \in (0.5, 1]$ are required to guarantee convergence.

We omit the full details of the online Gibbs sampler here due to the lack of space. The Supplementary Material provides more details of the update equations for online inference for both BPBFM-1 and BPBFM-2.

5.4 Related Work

There has been a significant amount of recent interest in the problem of learning from multi-relational data, both in the social/biological network analysis problems, as well as in modeling of large knowledge bases (such as YAGO, NELL, Freebase, etc.) that
consist of a massive number of triplet-based facts of the form entity-relation-entity, involving very large number of entities and relations. Because a natural representation for multi-relational data is in form of a three-way tensor (or a collection of matrices), a number of methods, closely related to each other, such as those based on tensor decomposition, collective matrix factorization, and generalizations of stochastic blockmodels for multi-relational data, have been proposed for learning from such data (Nickel et al., 2011; Sutskever et al., 2009; Jenatton et al., 2012; Rai et al., 2014; Zhu, 2012).

Another class of methods, especially in the context of learning from large knowledge bases, involve learning the embeddings of entities and relations, and using these embeddings to predict the unknown facts (or “links”) involving pairs of entities, given a specific relation. This class of methods typically includes (i) models such as those based on minimizing an energy function (Bordes et al., 2011; Socher et al., 2013; Bordes et al., 2014) involving valid entity-relation-entity triplets; and (ii) translated embedding based approaches (Bordes et al., 2013; Wang et al., 2014b; Yang et al., 2014) that embed the entities and the relations in a vector space based on the criteria that, for each valid triplet \( (h, r, t) \), the distance between the embeddings of the head and the tail, after the head entity embedding translated by the embedding of the relation, is smaller than the corresponding distance computed for an invalid triplet generated by perturbing either the head or the tail entity. We use some of these methods as baselines in our experiments.

There has been relatively little work on Bayesian methods for learning from large multi-relational data and knowledge bases (Sutskever et al., 2009; Rai et al., 2014; Zhu, 2012; Hu et al., 2015b). Although some of the existing Bayesian methods, such as the Bayesian Clustered Tensor Factorization (Sutskever et al., 2009), do provide nice modeling flexibility (e.g., discovering clusters of entities and relations in addition to being applicable for tasks such as link prediction), these methods are not able to
scale to large modern-day knowledge bases. In contrast, our proposed framework offers the various benefits of a generative, fully Bayesian model, in addition to being easily scalable for large multi-relational data sets, due to its dependence only on the positive triplets and the computational efficiency of the accompanying batch and online Gibbs sampling algorithms.

Finally, we would like to note that the Bernoulli-Poisson link for binary data has also been used recently in (Zhou, 2015). However, there are several key differences from our proposed framework: (1) the model in (Zhou, 2015) can only deal with a single relation type, whereas our framework allows learning from multi-relational networks and knowledge bases that consist of multiple types of relations; (2) in addition to handling multiple types of relations (BPBFM-1), our second model (BPBFM-2) allows further sharing of statistical strengths across multiple relations; and (3) while the model in (Zhou, 2015) relies on batch Gibbs sampling, we also develop online Gibbs sampling algorithms for both of our models, which allows us to apply these models to large-scale multi-relational data sets and knowledge bases.

5.5 Experiments

We evaluate both of our proposed models BPBFM-1 and BPBFM-2 on several benchmark data sets consisting of both moderate-sized multi-relational data sets, as well as large-scale benchmark knowledge bases (Section 5.5.1 provides more details of these data sets). In our experiments, we compare both our models (BPBFM-1 and BPBFM-2) with several state-of-the-art methods that include: (1) two bilinear latent factor models: RESCAL (Nickel et al., 2011) and LFM (Jenatton et al., 2012); (2) Bayesian logistic tensor factorization (Rai et al., 2014) (BLTF) for link-prediction from multi-relational data; (3) Zero-Truncated Poisson CP factorization (ZTP-CP) (Hu et al., 2015b) (4) Embedding based methods - TransE (Bordes et al., 2013) and TransH (Wang et al., 2014b), both of which are recently proposed
state-of-the-art methods for modeling large-scale knowledge bases. We report results on both quantitative comparisons (in terms of link-prediction/knowledge base completion accuracies), as well as qualitative analyses (e.g., analyzing clusters of entities/relations inferred by our models).

All our experiments were performed on a standard desktop with 24 GB RAM. In all our experiments, the hyperparameters $\beta_0$, $\beta_1$ and $\beta$ were fixed to 1, which worked well in practice.

5.5.1 Datasets

We use two groups of data sets in our experiments. The first group consists of three moderate-sized multi-relational data sets

- **Kinship**: This is a $104 \times 104 \times 26$ binary tensor (Nickel et al., 2011) containing 26 types of relations among a set of 104 individuals.

- **Nations**: This is a $14 \times 14 \times 56$ dataset describes the relations among 14 countries with respect to 56 types of interactions.

- **UMLS**: This is a $135 \times 135 \times 49$ dataset describes the causal influence among 135 biomedical concepts with respect to 49 types of interactions.

The other group consists of three large knowledge bases, and includes Freebase-15K, Wordnet-100K, and NELL-50K. Table 5.1 shows statistics of these data sets.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>FB-15K</th>
<th>NELL-50K</th>
<th>WN-100K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity #</td>
<td>14,951</td>
<td>29,904</td>
<td>38,696</td>
</tr>
<tr>
<td>Relation #</td>
<td>1345</td>
<td>233</td>
<td>11</td>
</tr>
<tr>
<td>Valid Triplet # (Train)</td>
<td>483,142</td>
<td>57365</td>
<td>112,581</td>
</tr>
<tr>
<td>Valid Triplet # (Test)</td>
<td>118,142</td>
<td>21,412</td>
<td>42,176</td>
</tr>
</tbody>
</table>
5.5.2 Experiments on Multi-Relational Data

In our first set of experiments, we evaluate both our models (using both batch as well as online inference) on the three multi-relational data sets (Kinship, UMLS, Nations) for the task of link-prediction, as well as for doing qualitative analyses (clustering entities and relations based on the topic-based embeddings inferred by our models).

5.5.3 Link Prediction and Computational Efficiency

We compare both our model with LFM, BLTF, and ZTPCP. For all methods, we set $K = 30$, and use 90% of valid triplets as training dataset and the remaining as
Table 5.5: Computational time comparison

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Kinship</th>
<th>UMLS</th>
<th>Nation</th>
</tr>
</thead>
<tbody>
<tr>
<td>LFM (Jenatton et al., 2012)</td>
<td>1.8704</td>
<td>6.4563</td>
<td>0.0726</td>
</tr>
<tr>
<td>BPBFM-1 (Batch Gibbs)</td>
<td>0.1993</td>
<td>0.1433</td>
<td>0.0709</td>
</tr>
<tr>
<td>BPBFM-2 (Batch Gibbs)</td>
<td>0.7142</td>
<td>0.4005</td>
<td>0.1240</td>
</tr>
</tbody>
</table>

testing. For LFM, we use default settings for other parameters as the code shared online. For model 2, we set \( M = 50 \). All other parameters in our two models are randomly initialized; however smarter initializations of the embeddings can also be used. Receiver Operating Characteristic AUC (AUC) and Precision Recall AUC (AUC-PR) are used to evaluate the performance for link prediction. As shown in Table 5.4, except for Kinship, our models achieve comparable or better AUC as compared to the other methods.

We also compare per-iteration computation times of our model with LFM (both methods implemented in Matlab) on the three small datasets, as shown in Table 5.5. We do not report timings of other baselines because the implementations are not directly comparable (Matlab vs Python vs C). Since the computational cost of our model scales only in the number of nonzeros in the data, we gain maximum speed-up for UMLS and minimum speed-up for Nation dataset. This is consistent with the fraction of nonzero entries for three data sets (Kinship: 0.0384, UMLS: 0.0076, Nation: 0.1844).

**Qualitative Analysis on UMLS Data**

Since each column of the matrix \( U \) inferred by our models corresponds to a topic, we use the columns of \( U \) to rank most prominent entities in each topic (based on the magnitude of entries in that column), as shown in Table 5.2. Likewise, using the \( M \times R \) non-negative matrix \( \eta \) inferred by BPBFM-2, we can group similar relation

\[ \text{http://tinyurl.com/q6a66ro} \]
Table 5.6: Most prominent entities in topics inferred for FB-15K, WN-100K and NELL-50K

<table>
<thead>
<tr>
<th>FB-15K</th>
<th>WN-100K</th>
<th>NELL-50K</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Biology</strong></td>
<td><strong>Country</strong></td>
<td><strong>Film</strong></td>
</tr>
<tr>
<td>animal kingdom</td>
<td>britain</td>
<td>valentine’s day</td>
</tr>
<tr>
<td>worm genus</td>
<td>america</td>
<td>harry potter</td>
</tr>
<tr>
<td>edible nut</td>
<td>france</td>
<td>new york stories</td>
</tr>
<tr>
<td>family</td>
<td>emerald isle</td>
<td>love actually</td>
</tr>
<tr>
<td>anacardiaceae</td>
<td>japan</td>
<td>grindhouse</td>
</tr>
<tr>
<td>bird footed dinos.</td>
<td>canada</td>
<td>terror in aisles</td>
</tr>
<tr>
<td>accipitridae</td>
<td>italia</td>
<td>who fra. roger rab.</td>
</tr>
<tr>
<td>aschelminthes</td>
<td>deutschland</td>
<td>Om Shanti Om</td>
</tr>
</tbody>
</table>

Table 5.7: Most prominent relations in topics inferred for FB-15K

<table>
<thead>
<tr>
<th>Education</th>
<th>Film/Award</th>
<th>Family</th>
<th>Individual Info</th>
<th>Sports</th>
</tr>
</thead>
<tbody>
<tr>
<td>institution</td>
<td>film</td>
<td>sibling</td>
<td>nationality</td>
<td>/sports/team</td>
</tr>
<tr>
<td>degree</td>
<td>actor</td>
<td>split to</td>
<td>location</td>
<td>/sports/position</td>
</tr>
<tr>
<td>major field of study</td>
<td>nominated for</td>
<td>parents</td>
<td>gender</td>
<td>/football/positions</td>
</tr>
<tr>
<td>student</td>
<td>award nominee</td>
<td>child</td>
<td>place of birth</td>
<td>/football/position</td>
</tr>
<tr>
<td>specialization</td>
<td>honored for</td>
<td>children</td>
<td>cause of death</td>
<td>/football/team</td>
</tr>
</tbody>
</table>

types by treating each row of $\eta$ as a “topic” and sorting the entries in that row to rank the relations (Table 5.3 shows the top 4 relations for each topic).

5.5.3 Experiments on Large Knowledge Bases

In our second set of experiments, we evaluate our models on large knowledge bases (Freebase15K, Wordnet-100K, and NELL-50K) on two tasks: knowledge base completion (predicting the validity of held-out triplets) and qualitative analyses (grouping entities and relations using the topic based embeddings learned by our models, as we did previously for UMLS data).

Knowledge Base Completion

For this task, in Table 5.8, we compare BPBFM-1 with two state-of-the-art knowledge base embedding methods - TransE (Bordes et al., 2013) and TransH (Wang et al., 2014b). We also provide, in Table 5.9, a separate comparison between BPBFM-1 and BPBFM-2 to discuss the benefits of using BPBFM-2 which is able to share
information across the different relations.

For the comparison between BPBFM-1 and BPBFM-2 (Table 5.9), for all three data sets, we set $K = 10$ for both models. For BPBFM-2, we set $M = 5$ for WN-100K since it has only 11 relations, and $M = 80$ for Freebase-15K and NELL-50 as the number of relations is much larger (Freebase-15K has 1345 relations and NELL-50K has 233 relations). As shown in Table 5.9, BPBFM-2 outperforms BPBFM-1, even if $K$ is as small as 10.

5.5.4 Qualitative Analyses on Freebase15K, NELL-50K, and WN-100K

The topic-based non-negative embeddings learned by our model can be useful for qualitative analyses. To illustrate this, we show results of our qualitative analyses on Freebase15K, NELL-50K, and WN-100K. Six of the factors (each factor represents a topic) inferred by BPBFM-2 are presented in table 5.6, and for each topic, we show top-8 entities. As the larger datasets contain more and richer entities than the smaller dataset, we can see a diverse set of factors, such as biology, countries, films, sports and musical bands, and all entities in each factor seem to be closely related to each other.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>FB-500K</th>
<th>NELL-50K</th>
<th>WN-100K</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransE</td>
<td>0.645</td>
<td>0.623</td>
<td>0.674</td>
</tr>
<tr>
<td>TransH</td>
<td>0.744</td>
<td>0.681</td>
<td>0.613</td>
</tr>
<tr>
<td>BPBFM-1</td>
<td><strong>0.780</strong></td>
<td><strong>0.774</strong></td>
<td><strong>0.681</strong></td>
</tr>
</tbody>
</table>

Table 5.8: AUC-PR for knowledge bases

<table>
<thead>
<tr>
<th>Datasets</th>
<th>FB-500K</th>
<th>NELL-50K</th>
<th>WN-100K</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPBFM-1</td>
<td>0.724</td>
<td>0.648</td>
<td>0.742</td>
</tr>
<tr>
<td>BPBFM-2</td>
<td><strong>0.727</strong></td>
<td><strong>0.665</strong></td>
<td><strong>0.783</strong></td>
</tr>
</tbody>
</table>

Table 5.9: AUC and AUC-PR (A-PR below) comparison between BPBFM-1 and BPBFM-2
In addition, to indicate BPBFM-2’s ability in learning the prominent relations in different topics, we present five topics of relations, and for each topic show top-5 relations. The results are shown in Table 5.7. In factor 1, we find that these relations are all about education, and the other relations in the other four topics make sense as well.

We also show the inferred similarities between different relations by treating each column of matrix $\eta$ as the feature vector for the corresponding relation. As WN-100K has only 11 relations, it is an ideal dataset to intuitively show relation similarities via a correlation plot. The cosine similarity among relations are shown in figure 5.4. In the plot three pairs of relations, \{type_of, has_instance\}, \{part_of, has_part\}, and \{member_holonym, member_meronym\}, are found similar to each other, which makes a lot sense because if we switch the head and tail in a triplet \{head, relation, tail\}, the two relations in each pair are basically the same relation.

Figure 5.4 the pairwise similarities between the 11 relations present in the Wordnet data as computed using the inferred $\eta$ matrix by BPBFM-2. As shown in the
plot, the model is able to correctly infer the similarities between the relations.
6

Conclusions and Future Work

6.1 Summary of Contributions

In this thesis, we have introduced a set of probabilistic factorization models for discrete (binary and count-valued) matrix and tensor data. The main contributions are summarized as follows.

1. We have presented a probabilistic non-negative matrix factorization model, which can easily incorporate structural side-information along rows/columns. The proposed model can fit both count as well as binary data under a unified framework. The topics and the embeddings learned by our model can be useful for various other downstream tasks (e.g., classification) or for qualitative analyses. And the ease of inference, makes our framework particularly attractive for applications involving discrete data with structural side-information. The model can also be extended to handle binary/count tensor data (Hu et al., 2015b,a; Schein et al., 2015) with structural side-information given along one or more of the tensor modes.

2. We have presented a fully Bayesian framework for analyzing massive tensors
with count data, and have designed a suite of scalable inference algorithms for handling massive tensor data. In addition to giving interpretable results and inferring the rank from the data, the proposed model can infer the distribution over objects in each of the tensor modes which can be useful for understanding groups of similar objects, and also for doing other types of qualitative analyses on such data, as shown by our various experiments on real-world data sets.

3. We have extended our count-valued tensor factorization to model binary observations by using a Bernoulli-Poisson link. In contrast to the models based on probit or logistic likelihood for binary tensor decomposition, the time-complexity of our model depends only in the number of ones in the tensor. This aspect of our model allows it to easily scale up to massive binary tensors. The simplicity of our model also leads to simple batch as well as online MCMC inference; the latter allows our model to scale up even when the number of ones could be massive. Our experimental results demonstrate that the model leads to speed-ups of an order of magnitude when compared to binary tensor factorization models based on the logistic likelihood, and also outperforms various other baselines. Our model also gives interpretable results which helps qualitative analysis of results. In addition, the ability to leverage mode networks (fully or partially observed) leads to improved tensor decomposition in cold-start problems.

4. We have presented a bilinear non-negative latent factor model to analyze large multi-relational data. A rich generative modeling framework enables our models to not just learn embeddings of entities and relations and perform tasks such as link-prediction and knowledge-based completion, but also gives interpretable results for further qualitative analyses. In particular, the topic-based embeddings learned by our models can be useful in itself, e.g., for grouping enti-
ties and/or relations in terms of the topics they represent. Computational cost that scales w.r.t. the number of positive triplets makes our framework an ideal choice for learning from real-world multi-relations data that are massive (in terms of number of entities and relations) yet have very few positive triplets. Our framework can be extended in several directions; for example, allowing new entities and/or relations to be added; or incorporating other sources of information, e.g., a text corpus in addition to the knowledge base (Wang et al., 2014a).

6.2 Future Work

Based on the Bayesian framework of matrix/tensor factorization discussed in this thesis, there are a few future directions applicable, listed as follows.

1. Simplicity of the inference procedure makes the proposed model amenable for parallel and distributed implementations. e.g., using MapReduce or Spark. The model can be a useful tool for analyzing data from diverse applications and scalability of the model opens door to the application of scalable Bayesian methods for analyzing massive discrete data.

2. Many observed matrix/tensor shows dynamic changes in time. Therefore, it will be interesting to add a temporal dimension (e.g., for multi-relational data, a fact may be a true over a period of time but not forever (Dong et al., 2014)) to the observations, and develop a factorization method which can model the temporal evolution change of “topics”/themes.
A.1 Gibbs Sampling for BPBFM-2

Just as we did for BPBFM-1, we can express each latent count $X_{ij}^r$ in BPBFM-2 (which models each $\Lambda^r$ as $\Lambda^r = \sum_{m=1}^{M} \eta_{mr} G_{m}^r$) as a sum of the following form:

$$X_{ij}^r = \sum_{k_1}^{K} \sum_{k_2}^{K} \sum_{m}^{M} X_{ik_1k_2mj}^r$$

where $X_{ik_1k_2mj}^r \sim \text{Poisson}(u_{ik_1} \eta_{mr} G_{k_1k_2}^m u_{jk_2})$. We further define

$$X_{i,k_1k_2m} = \sum_{i}^{N} \sum_{j}^{N} \sum_{r=1}^{R} X_{ik_1k_2mj}^r$$

and

$$X_{i..-m} = \sum_{i}^{N} \sum_{j}^{N} \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} X_{ik_1k_2mj}^r$$

using additive property of Poisson distribution.

$$X_{k_1k_2m} \sim \text{Poisson}(\theta_{k_1k_2} G_{k_1k_2}^m \sum_{r=1}^{R} \eta_{mr})$$  \hspace{1cm} (A.1)

$$X_{i..-m} \sim \text{Poisson}(\eta_{mr} \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \theta_{k_1k_2} G_{k_1k_2}^m)$$  \hspace{1cm} (A.2)

With these defined, we proceed to give the update equations for the Gibbs sampler for BPBFM-2.

**Sampling $X_{ij}^r$:** $X_{ij}^r$ is sampled just as in model-1.
Sampling $X_{ik_1k_2mj}^r$: $X_{ik_1k_2mj}^r$ can be sampled as
\[ X_{ik_1k_2j}^r \sim \text{Mult}(X_{ij}; \frac{u_{ik_1} \eta_{mr} G_{ik_1k_2}^m u_{jk_2}}{\sum_{k'_1=1}^{K} \sum_{k'_2=1}^{K} u_{ik_1} X_{ik_1k_2}^r u_{jk_2}}) \] (A.3)

Sampling $U_{:,k}$: Using Dirichlet-multinomial conjugacy, each column of $U$ can be sampled as
\[ U_{:,k} \sim \text{Dir}(a + X_{ik}\ldots, a + X_{2k}\ldots, \ldots, a + X_{Nk}\ldots) \] (A.4)
where $X_{ik}\ldots = \sum_{k'_2=1}^{K} \sum_{j=1}^{M} \sum_{m=1}^{R} X_{ikk2mj}^r$.

Sampling $d_{mk}^m$: Marginalizing out $G_{k_1k_2}^m$ from Eq.(A.1), we have
\[ X_{k_1k_2m}^r \sim \text{NegBin}((\epsilon^m)^{\delta_{k_1k_2}} d_{k_1}^m (d_{k_2}^m)^{1-\delta_{k_1k_2}}, p_{k_1k_2}) \] (A.5)
where $p_{k_1k_2} = \frac{\theta_{k_1k_2} \sum_{r=1}^{R} \eta_{mr}}{\theta_{k_1k_2} \sum_{r=1}^{R} \eta_{mr} + \beta}$. Using the data augmentation scheme proposed we used for BPBFM-1, $d_{mk}^m$ can be sampled by first sampling
\[ \ell_{kk2}^m \sim \text{Bern}(\frac{(\epsilon^m)^{\delta_{k_1k_2}} d_{k_1}^m (d_{k_2}^m)^{1-\delta_{k_1k_2}}}{(\epsilon^m)^{\delta_{k_1k_2}} d_{k_1}^m (d_{k_2}^m)^{1-\delta_{k_1k_2}} + t - 1}) \] (A.6)
and then sampling
\[ d_{mk}^m \sim \text{Ga}(\frac{\gamma_0}{K} + \sum_{k'_2}^K \ell_{kk2}^m, \frac{1}{c_0 - \sum_{k'_2}^K (\epsilon^m)^{\delta_{k_1k_2}} d_{k_1}^m (d_{k_2}^m)^{1-\delta_{k_1k_2}} \ln(1 - p_{kk2})}) \] (A.7)

Sampling $\epsilon^m$: $\epsilon^m$ can be sampled as
\[ \epsilon^r \sim \text{Ga}(e_0 + \sum_k \ell_{kk}^r, \frac{1}{f_0 - \sum_k \ell_{kk}^m \ln(1 - p_{kk})}) \] (A.8)

Sampling $G_{k_1k_2}^m$: Using Gamma-Poisson conjugacy, $G_{k_1k_2}^m$ can be sampled by
\[ G_{k_1k_2}^m \sim \text{Ga}((\epsilon^m)^{\delta_{k_1k_2}} d_{k_1}^m (d_{k_2}^m)^{1-\delta_{k_1k_2}} + X_{k_1k_2m}^r, \frac{1}{\beta + \theta_{k_1k_2} \sum_{r=1}^{R} \eta_{mr}}) \]
Sampling $\eta_{mr}$: Using equation (A.2) and Gamma-Poisson conjugacy, $\eta_{mr}$ can be sampled by

$$
\eta_{mr} \sim \text{Ga}(h_0 + \lambda_{r-m}, \frac{1}{q_0 + \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \theta_{k_1k_2} G_{m}^{k_1k_2}})
$$

(A.9)

A.2 Online Gibbs Sampling

In this section, we provide the details of the online Gibbs sampling algorithms for both of our models. Our online Gibbs sampling algorithms are based on the idea of the recently developed Bayesian Conditional Density Filtering (BCDF) framework (Guhaniyogi et al., 2014). They key idea in BCDF is to process data in small minibatches, and maintain and update sufficient statistics of the model parameters with each new minibatch of the data. In our models, these sufficient statistics are the latent counts.

A.2.1 Online Gibbs Sampling for BPBFM-1

Denoting $I_t$ as indices of valid triplets in minibatch selected at iteration $t$, and $I$ as the indices of all the valid triplets in training data. Define $\mathcal{X}_{r,t}^t = \frac{|I|}{|I_t|} \sum_{r=1}^{R} \mathcal{X}_{ik}^t$, and $\mathcal{X}_{k_1k_2}^t = \frac{|I|}{|I_t|} \sum_{i,j \in I_t} \mathcal{X}_{ik_{1}k_{2j}}^t$, where $|I|$ and $|I_t|$ are cardinalities of the two sets. Then similar to batch Gibbs Sampling, define following quantities for $t \leq 2$: $\mathcal{X}_{ik}^{t-1} = (1 - \rho) \mathcal{X}_{ik}^{t} + \rho \frac{|I|}{|I_t|} \sum_{r=1}^{R} \mathcal{X}_{ik_{r}}^{t}$, and $\mathcal{X}_{k_1k_2}^{t-1} = (1 - \rho) \mathcal{X}_{k_1k_2}^{t} + \rho \frac{|I|}{|I_t|} \sum_{i,j \in I_t} \mathcal{X}_{ik_{1}k_{2j}}^{t}$. Here $\rho = (t + t_0)^{-w}$ is a decaying learning rate, as used in other online inference algorithms, such as stochastic variational inference (Hoffman et al., 2013). Here, $t_0 > 0$ and $w \in (0.5, 1]$ are required to guarantee convergence. With these defined, online Gibbs sampling at iteration $t$ proceeds as:

Sampling $U_{i,k}$: Each column of $U$ can be sampled as
\[ U_{i,k} \sim \text{Dir}(a + \mathcal{X}_{1,k}^t, a + \mathcal{X}_{2,k}^t, \ldots, a + \mathcal{X}_{N,k}^t) \]  
(A.10)

**Sampling** \( d^r_k \): \( d^r_k \) can be sampled by first sampling

\[ \ell^r_{kk} \sim \sum_{t=1}^{\mathcal{X}_{k1,k2}^r} \text{Bern}(\frac{(\varepsilon^r)^{\delta_{k1,k2}} d^r_{k1} (d^r_{k2})^{1-\delta_{k1,k2}}}{(\varepsilon^r)^{\delta_{k1,k2}} (d^r_{k1})^{1-\delta_{k1,k2}} + t - 1}) \]  
(A.11)

and then sampling

\[ d^r_k \sim \text{Ga}(\frac{\gamma_0}{K} + \sum_{k_2}^{K} \ell^r_{kk}, \frac{1}{c_0 - \sum_{k_2}^{K} (\varepsilon^r)^{\delta_{k1,k2}} (d^r_{k1})^{1-\delta_{k1,k2}} \ln(1 - p_{kk2})}) \]  
(A.12)

**Sampling** \( \varepsilon^r \): \( \varepsilon^r \) can be sampled as

\[ \varepsilon^r \sim \text{Ga}(e_0 + \sum_{k}^{K} \ell^r_{kk}, \frac{1}{f_0 - \sum_{k}^{K} d^r_k \ln(1 - p_{kk})}) \]  
(A.13)

**Sampling** \( \Lambda^r_{k1,k2} \): \( \Lambda^r_{k1,k2} \) can be sampled by

\[ \Lambda^r_{k1,k2} \sim \text{Ga}(\frac{\varepsilon^r)^{\delta_{k1,k2}} d^r_{k1} (d^r_{k2})^{1-\delta_{k1,k2}} + \mathcal{X}_{k1,k2}^r, \frac{1}{\beta + \theta_{k1,k2}}) \]  
(A.14)

\( \mathcal{X}_{ij}^r, \varepsilon^r \) and \( \mathcal{X}_{ik1,k2j}^r \) are sampled the same way as the batch Gibbs sampling.

**A.2.2 Online Gibbs Sampling for BPBFM-2**

Similar to online BPBFM-1, we define \( \mathcal{X}_{k1,k2}^{r,t} = (1-\rho)\mathcal{X}_{k1,k2}^{r,t-1} + \rho \frac{|I|}{|I_t|} \sum_{i,j \in I_t} \sum_{r=1}^{R} \mathcal{X}_{i,k1,k2j}^r, \mathcal{X}_{i,k2}^{r,t} = (1-\rho)\mathcal{X}_{i,k2}^{r,t-1} + \rho \frac{|I|}{|I_t|} \sum_{i,j \in I_t} \sum_{k=1}^{K} \mathcal{X}_{i,k,k2}^r, \mathcal{X}_{i,k}^{r,t} = (1-\rho)\mathcal{X}_{i,k2}^{r,t-1} + \rho \frac{|I|}{|I_t|} \sum_{i,j \in I_t} \sum_{m=1}^{M} \mathcal{X}_{i,k2}^r, \mathcal{X}_{i,k}^{r,t} = (1-\rho)\mathcal{X}_{i,k2}^{r,t-1} + \rho \frac{|I|}{|I_t|} \sum_{i,j \in I_t} \sum_{m=1}^{M} \mathcal{X}_{i,k2}^r. \) With these defined, we proceed to give the update equations for the online Gibbs sampler for model-2:

**Sampling** \( U_{i,k} \): Each column of \( U \) can be sampled as

\[ U_{i,k} \sim \text{Dir}(a + \mathcal{X}_{i,k}^{t}, a + \mathcal{X}_{i,k}^{t}, \ldots, a + \mathcal{X}_{i,n}^{t}) \]  
(A.15)
**Sampling** $d_m^k$: $d_m^k$ can be sampled by first sampling

$$
\ell_{kk}^m \sim \sum_{t=1}^{X_{k1k2}^m} \text{Bern}\left( \frac{(e^m)_{\delta k1k2} d_m^k (d_m^k)^{1-\delta k1k2}}{(e^m)_{\delta k1k2} d_m^k (d_m^k)^{1-\delta k1k2} + t - 1} \right)
$$

(A.16)

and then sampling

$$
d_k^m \sim \text{Ga}\left( \frac{\gamma_0}{K} + \sum_{k2}^{K} \ell_{kk}^m, \frac{1}{c_0 - \sum_{k2}^{K} (e^m)_{\delta k1k2} (d_k^m)^{1-\delta k1k2} \ln(1 - p_{kk2})} \right)
$$

(A.17)

**Sampling** $G_m^{k1k2}$: Using Gamma-Poisson conjugacy, $G_m^{k1k2}$ can be sampled by

$$
G_m^{k1k2} \sim \text{Ga}\left( (e^m)_{\delta k1k2} d_k^{r}, (d_k^{r})^{1-\delta k1k2} + X_{k1k2}^m, \frac{1}{\beta + \theta_{k1k2} \sum_{r=1}^{R} \eta_{mr}} \right)
$$

**Sampling** $\eta_{mr}$: $\eta_{mr}$ can be sampled by

$$
\eta_{mr} \sim \text{Ga}\left( h_0 + \chi_{r-m}^t, \frac{1}{q_0 + \sum_{k1=1}^{K} \sum_{k2=1}^{K} \theta_{k1k2} G_m^{k1k2}} \right)
$$

(A.18)

$\chi_{ij}^r$, $\chi_{ik1k2m}^r$, $e^m$, $\epsilon^m$ can be sampled the same way as the batch Gibbs sampling.


107


Shashua, A. and Hazan, T. (2005), “Non-negative tensor factorization with applications to statistics and computer vision,” in ICML.


Sutskever, I., Tenenbaum, J. B., and Salakhutdinov, R. R. (2009), “Modelling relational data using Bayesian clustered tensor factorization,” in NIPS.


Biography

Changwei Hu was born in Qingdao, China on June 23, 1985. He is currently studying toward the Ph.D. degree in Electrical and Computer Engineering at Duke University, advised by Professor Lawrence Carin and Piyush Rai. Before joining Duke University, he obtained Master’s degree in Electromagnetic Field and Microwave Technology, and Bachelor’s degree in Software Engineering, both from Xiamen University in China. His research interests include machine learning and Bayesian statistics.

PUBLICATIONS:


5. P Rai, C Hu, R Henao, L Carin. Large-Scale Bayesian Multi-Label Learning via Topic-Based Label Embeddings, NIPS 2015, Montreal, Canada. (Spotlight


