PROBABILISTIC GRAPHICAL MODELING FOR LATENT FEATURE LEARNING

LU WEI

Ph.D

The Hong Kong Polytechnic University

2017

The Hong Kong Polytechnic University

Department of Computing

Probabilistic Graphical Modeling for Latent Feature Learning

Lu Wei

A thesis submitted in partial fulfilment of the requirements

for the degree of Doctor of Philosophy

Sep 2016

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

__________________________(Signature)

__________LU Wei_________(Name of student)

Dedicate to my parents.

Abstract

With increasing availability of digitized knowledge, it has been increasingly important to develop statistical models to manage large-scale and high-dimensional heterogeneous data, making hierarchical learning on these various kinds of data a challenging problem. Despite the extensive research on hierarchical topic mining and deep representations, there are still numerous issues that have not been sufficiently addressed, such as dealing with sparsity issues, interpretability of inner structure, serendipitous recommendation and transfer of learned deep features to new domains. To overcome these challenges, there is pressing need to develop hierarchical learning methods for various kinds of dataset problems with diverse feature sets. The aims of this work are to develop novel probabilistic graphical models that can automatically learn good feature representation from sparse data using multiple sources and types of auxiliary data, and apply the models to machine learning tasks including semantic topic understanding, video recommender system and unsupervised/semi-supervised image classification.

Targeting at the sparsity issue of text data applications, the first two approaches are introduced from topic modeling perspectives. Firstly, we investigate how auxiliary information can benefit content analysis for hierarchical topic mining when the text length are biased short. Through incorporating relational meta information, this algorithm takes advantage of the natural hierarchical structure and infers topics by jointly modeling word and taxonomic node assignments for documents.

Secondly, addressing the sparseness phenomenon in a recommender system application scenario, instead of regard one of the two observations as auxiliary information, we consider the problem in a collaborative way. Motivated by a real world online video recommendation problem, we target at the long tail phenomena of user behavior and scarceness issues of item features, and propose a personalized compound recommendation framework for online video recommendation called Dirichlet mixture probit model for information scarcity (DPIS), a probit classifier utilizing record topical clustering on the user part for recommendation.

The third and fourth models also start from an unsupervised perspective while incorporating multi-layer features for recommendation and domain adaptation. The third model is based on a useful approach for complex multi-relational data learning and missing element completion from a tensor perspective, where a deep probabilistic tensor decomposition model for item recommendation and tag completion is proposed. We also apply the proposed algorithm to computational creativity, an emerging domain of application, emphasizing the use of big data to automatically design new knowledge, resulting to attain serendipitous recommendation.

The fourth model is based on multi-layer sparse factorization. Deep architectures can now be well trained on massive labeled data. However, there exist many application scenarios, where labeled data are sparse or absent. Domain adaptation and multi-task transfer learning provide attractive options when related labeled data or tasks are abundant from different domains. In this part, a new graphic modeling approach to multi-layer factorization based domain adaptation is explored to address the scenarios that sufficient labeled data are available from the source domain while only a small subset or no labeled data can be used for supervised learning. A deep convolutional factorization based transfer learning (DCFTL) is proposed to facilitate layer-wise transfer learning between domains. Completely based on graphical model representation, the proposed framework can seamlessly merge inference and

learning, and has clear interpretability of conditional independence. The empirical performances on image classification tasks in both supervised and semi-supervised adaptation settings illustrate the effectiveness and generalization of knowledge transfer framework.

Publications

CONFERENCE PAPERS

- **Wei Lu** and Fu-lai Chung, "Computational Creativity based Video Recommendation", in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR'16), July 17−21, 2016, Pisa, Italy.

- **Wei Lu**, Fu-lai Chung and Kunfeng Lai, "Scarce Feature Topic Mining for Video Recommendation", in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (CIKM'16), October 24−28, 2016, Indianapolis, Indiana, USA.

- **Wei Lu** and Fu-lai Chung, "Deep Bayesian Tensor for Recommender System", in *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases Doctoral Consortium* (ECMLPKDD'15), September 7−11, 2015, Porto, Portugal.

JOURNAL PAPERS

- **Wei Lu**, Fu-lai Chung, Kunfeng Lai and Liang Zhang, "Recommender System Based on Scarce Information Mining", under 3rd round review by *Neural Networks*, 2017.

- **Wei Lu** and Fu-lai Chung, Auxiliary Information based Hierarchical Topic Model: Mining Biased Short text, under review by *Information Sciences*, 2016.

- **Wei Lu** and Fu-lai Chung, “A Deep Graphical Model for Layered Knowledge Transfer”, submitted.

- Anna Tyler, **Wei Lu**, Justin J. Hendrick, Vivek M. Philip, Gregory W. Carter, “CAPE: An R package for Combined Analysis of Pleiotropy and Epistasis”, in *PLoS Computational Biology*, 9.10, 2013.

Acknowledgements

The whole research process is a bitter sweet journey. First and foremost, I would like to express my sincere gratitude to my supervisor, Professor Korris Fu-lai Chung, for his enlightening guidance, great support and inspirations. Thank you for always encouraging me to think critically and independently, and giving me enough freedom to explore diverse areas.

I would also like to show my deep appreciation for my dissertation committee member Professor Maggie Wenjie Li, Professor James Tin-Yau Kwok and Professor ChengXiang Zhai for their time and helpful feedbacks. I am also grateful to Professor Qin Lu for kindly providing me with help during my study.

I also want to thank my colleagues, friends and family members for providing an excellent research atmosphere and caring environment for my daily life. Special thanks go to Wenhao Jiang, Chengyao Chen, Shaobo Han, Yumeng Guo, Xiao Shen, Jiaxin Chen, Sitong Mao, Wengen Li, Yanxing Hu, Yanran Li, Kunfeng Lai, Beiyuan Jingzi, Jing Liu, Jing Zeng, my cousins and grandparents. It is really wonderful and lucky to have all of you around.

Finally and most importantly, I would like to dedicate this dissertation to my beloved parents Zhu Hong and Lu Li for their love and support. You both influenced me in many ways and helped me become a person I am proud to be. Especially, to my dad, let this be your strength to defeat the disease!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

With advanced data acquisition and storage techniques, the rapid proliferation of various data types are being generated on a daily basis in many applications. These technologies provide a great boost to the data repository architecture [33], which is organized by multiple heterogeneous data sources under a unified schema. Meanwhile, internet-based global information bases, such as social media, computational biology and e-commerce, have emerged and raised great challenges to the traditional information retrieval, data mining and network analysis technologies. This stimulates *learning from data* a prevalent and essential problem in the current era of Big Data. Particularly, in the field of machine learning, there has been a rapid growth in the development of computational tools for tackling the unprecedented challenges introduced by the increasing availability of data.

The wide diversity of data sources nowadays brings various aspects of challenges to machine learning. From the methodology perspective, we primarily focus on two aspects.

1. **Complex and new kinds of knowledge:** Diverse sources generate different formats of data, from continuous to discrete, structured to unstructured, and from simple data objects to sequential ones. It is unrealistic to expect one

algorithm or model to analyze all kinds of data, given the diversity of data types and mining purposes. The design of effective framework and construction of efficient mining tools remains a challenging and active research task.

2. **Dynamic and sparse data features:** Numerous data repositories available nowadays are connected via different kinds of networks, forming distinctively distributed, and heterogeneous global information systems. As data often contain noise, errors, uncertainty, or incompleteness, how to effectively discover the latent patterns and knowledge in those heterogeneous data sets is a fast evolving research focus. Particularly, we focus on the sparseness issue of different data set, and try to explore how to incorporate various interacted datasets to solve the sparsity issue and find useful representations from their dynamic features.

1.2 Probabilistic Graphical Models

Graphical models, as a combination of probability and graph theory, provide a natural tool for dealing with uncertainty and complexity in both applied mathematical and engineering problems, and play an important role in the design of machine learning algorithms [40]. They treat inference and learning together with a merging of supervised and unsupervised learning. Graphical models can handle missing data, and provide interpretable results.

There are two perspectives to interpret the structure of the graph [45]. First, the graph can compactly represent the distribution independencies; Second, the graph also defines a structure for high-dimensional distribution representation, where the overall distribution can be represented as a production of smaller factor spaces. These two perspectives are, in a deep sense, equivalent. As the product separation guarantees the independency of the distributions.

1.3 Feature Representation and Deep Learning

Latent feature learning allow us to learn a compact representation that can simultaneously explain our observations as well as unobserved ones. To avoid overfitting, we transform each n-dimensional input into a new representation, i.e. Φ composed of K features. Depending on the availability of training sample label, this learning process can be further divided into “supervised”,“semi-supervised” and “unsupervised” settings. We can think of the feature functions, that transfer original input into the feature matrix, as providing a way to encode prior knowledge into the learning system. Each feature value could represent a “higher-level” concept that usually could not be directly expressed by sample attribute.

Recently, deep networks prove great empirical success in various domains [6], [50]. Researches have been done on convolutional/deconvolutional networks [48], [102], sparse auto-encoders [91], deep convolutional neural networks [51], and restricted Boltzmann machines (RBM) [81]. They are capable of providing more compact nonlinear representations for feature learning.

Besides the main focus of applications on supervised deep learning, which has dominated pattern recognition area, there are growing attention on unsupervised deep learning, i.e. deep generative models [80]. Hinton et al. [35] introduced a moderately fast, unsupervised learning algorithm for deep generative models called deep belief networks (DBNs). DBNs are probabilistic graphical models that contain multiple layers of hidden variables. Each nonlinear layer captures progressively more complex patterns of data.

Several frameworks including deep Gaussian Process [19], deep Poisson factorization analysis [26] have been put forward for semi-supervised or unsupervised multi-layer structure mining, dealing with scenarios where only a small subset or even no observations have corresponding labels.

Comparing to traditional probabilistic mixture-based latent variable topic models, which can be viewed as graphical models, models that use nonlinear distributed representations are able to generalize better than latent Dirichlet allocation in terms of both the log-probability on previously unseen data vectors and the retrieval accuracy [19]. Besides, the unsupervised setting provides an effective approach for transfer learning, where the labeled samples are scarce. Hence, the power of deep generative models still remains as a challenge for exploitation.

1.4 Overview

Targeting at learning effective hierarchical representation, the thesis proposed several graphical models. The first three models are all expected to solve the data sparseness issue, with the first two from a co-direction matrix factorization perspective, and the third one from a multidimensional tensor decomposition perspective. Considering from the feature learning methodology aspect, the first two models target at hierarchical feature learning of the diverse heterogeneous data types from traditional topic model perspective using shallow topical features. The third and fourth models present unified generative frameworks for analyzing data from diverse repositories using deep canonical features. We elaborate further on these models below, which also serves as a brief summary and provides the context for the remainder of this thesis.

The first model investigates how auxiliary information can benefit content analysis for hierarchical topic mining. Focused on several modern types of digitized collective knowledge, which are characterized by dynamically large vocabulary but much less tokens than traditional corpus, an auxiliary information based hierarchical Latent Dirichlet Allocation (ai-hLDA) model is proposed. Through incorporating relational information, this algorithm takes advantage of the natural hierarchical structure and infers topics by jointly modeling word and taxonomic node assignments for documents. Experiments on short-length discrete data sets including Enron E-mail corpus, course selection database and video text data set in Chinese characters show that ai-hLDA achieves satisfactory performance in obtaining concise structural representation and accurate clustering.

The second model focuses on recommendation for user generated content sites. To satisfy the niche tastes of users, long tail product recommendation poses more challenges due to the data sparsity issue. The proposed model is motivated by a real world online video recommendation problem, where the database of click records suffers from two aspects of sparseness. First, the video inventory is volatile and long tailed. So most of the records are concentrated on popular ones. Second, there is a lack of holistic content information. The tags for videos have the features of being short, missing or mistaken. Targeting the long tail phenomena of user behavior and scarceness issues of item features, we propose a personalized compound recommendation framework for online video recommendation called Dirichlet mixture probit model for information scarcity (DPIS). Assuming that each record is generated from a representation of user preferences, DPIS is a probit classifier utilizing record topical clustering on the user part for recommendation. As demonstrated by the real-world application, the proposed DPIS achieves better performance than traditional methods in both warm-start and cold-start scenarios.

The third model discusses a useful approach for complex multi-relational data learning and missing element completion from a tensor perspective. A deep probabilistic tensor decomposition model for item recommendation and tag completion is proposed. Extended from the Canonical PARAFAC (CP) decomposition, this method provides a fully conjugate Bayesian treatment for parameter learning. Through incorporating multi-layer factorization, a richer feature representation facilitates a better and comprehensive understanding of user behaviors and hence gives more helpful recommendations. The new algorithm, called Deep Canonical PARAFAC Factorization (DCPF), is evaluated on both synthetic and large-scale real world problems. Empirical results demonstrate the superiority of the proposed method and indicate that it can better capture the latent patterns of interaction relationships.

We also enhance the proposed DCPF with computational computational creativity, an emerging domain of application, emphasizing the use of big data to automatically design new knowledge. Based on the availability of complex multi-relational data, one aspect of computational creativity is to infer unexplored regions of feature space and novel learning paradigm, which is particularly useful for online recommendation. Tensor models offer effective approaches for complex multi-relational data learning and missing element completion. Targeting at constructing a recommender system that can compromise between accuracy and creativity for users, a deep Bayesian probabilistic tensor framework for tag and item recommendation is adopted. Empirical results demonstrate the superiority of the proposed method and indicate that it can better capture latent patterns of interaction relationships and generate interesting recommendations based on creative tag combinations.

The fourth model corresponds to a multi-layer approach for inner structure understanding. Deep architectures are well trained on massive labeled data. However, there are many application scenarios, where labeled data are sparse or absent. Domain adaptation and multi-task transfer learning provide attractive options when related labeled data or tasks are only abundant from different domains. Hence, we propose a new approach to domain adaptation based multi-layer factorization that can be trained on labeled data from the source domain, while only a subset or no labeled data from the target domain are available. Assuming that domains sharing greater similarities can benefit more from class-specific information encoded at higher layers of the source data, the approach promotes the emergence of layer-wise feature extraction. Hence, for less invariant domains, local features computed in lower layers will yield better discriminative power. Thus, a deep sparse factorization for transfer learning method is proposed and offers empirical performances in a series of image classification experiments in both supervised and semi-supervised adaptation settings, which exceeds baseline and some previous classification methods.

Chapter 2

Auxiliary Information Based Hierarchical Topic Mining for Biased Short Text

2.1 Introduction

With increasing availability of large-scale digitized data, there is a growing need of computational techniques for effective text mining. New mediums and systems of information diffusion, including emails, online video description, e-shopping reviews, and social network posts, allow their users to acquire and spread information much more effectively than ever before. These prevalent and distinctive sources of data often contain sampling biases. With considerably large vocabularies, but short entity length, they are differently distributed comparing to traditional document collections, such as journal or news corpus. As a result, new approaches are required for discovering meaningful knowledge from the large-scale user-generated biased texts.

Probabilistic topic models have been proved to be effective analytic tools of text content [8]. For classical topic models, assuming that a document is generated from a mixture of topics, the words belonging to each document are sampled from a distribution over the vocabulary. The generated topic proportions can hence be used as a high-level semantic representation depicted by those top weighted words.

While topic models have many successful applications in text mining domains, the experiences on new medium are mixed. For example, a model is expected to build on a corpus consisting of sentences with less than 40 words, the biased samples contain more noisy and dynamic features comparing to traditional media data [98]. When represented in an unordered "bag of words" way, such characteristics raise challenges for conventional text topic mining models, such as Latent Dirichlet Allocation (LDA) [10], which depends on co-occurring frequencies for topic discovery. When a rich collection of words is a basic guarantee for satisfactory results, the limited amount of texts would cause difficulty for topic detection.

It is crucial to organize short text segments, such as keywords of queries, into a well-formed hierarchy. In this kind of information retrieval system, deriving topic hierarchies from document corpus, could provide a comprehensive format for presenting those documents [16]. As the document-level word co-occurrence possesses the characteristics of being sparse and noisy, single layer representation may fail to grasp the essential semantics and hence generates ambiguity. Since complex topics are naturally organized in a hierarchical way, granular control such as tree structure arrangement can provide more meaningful and reasonable interpretations. This inspires us to address the sparsity of both the topic mixtures and the word distributions along a hierarchical path.

There are many existing approaches purely leverage one modality of information, e.g. the text itself [36]. Similar needs for auto-generation of topic hierarchies could occur in question answering systems, where short texts are defined as a meaningful word string. They are often short in length but represent specific concepts in a certain subject domain, such as a keyword in a document set and a natural language query from a user. Since these short-length forms of digital datasets usually have a richer set of auxiliary information (e.g. author, recipient, degree type, time, friend links, followers, hash tagged themes, etc.), which is connected through inner taxonomy, a layered mechanisms using supportive information modalities will in turn provide better browsing and interpreting usage of large data sets.

In this chapter, a weakly supervised hierarchical model is proposed to discover topical hierarchies of biased short data samples. Incorporating the available auxiliary information, the model, called ai-hLDA (abbreviated for auxiliary information based hierarchical Latent Dirichlet Allocation), arranges shorter length documents along paths of a tree for taxonomies. A Beta-Bernoulli framework is adopted to model the link of supplemental attribute information between the documents. The proposed model takes advantage of the hierarchical nature, and jointly models word and auxiliary information as a generative process. As there is no aggregation involved to expand short document into longer one, which is a common practice of handling short text [37], the mechanism would not interfere with the original structure of data. Quantitative evaluations of perplexity and clustering prediction accuracy were conducted for comparing with several other related algorithms.

2.2 Related Work

The state-of-the-art work on Latent Dirichlet Allocation (LDA) provides an extensible framework for many of the following topic models. In LDA, each document is viewed as a mixture of latent probabilistic topics, and the words in that document could be a representation of certain subsets of those topics. However, as an unsupervised generative model, even when certain resources exist, e.g. document labels, it fails to provide any suitable tools to tune the generated topics. The labeled LDA model [75] extends LDA by defining a correspondence between latent topics and user tags for learning word-tag correspondences, and can be used as a multi-label text classifier. Balasubramanyan et al. [3] propose a biased model to address entity-topic analysis and apply it to hidden information detection. McCallum et al.

[1] put forward a topic model for mining the social networks and people's roles in email communication using the author-recipient information. Although the concept of auxiliary information is not specifically used in their work, it steers the discovery of topics according to the personal relationships.

Another extension of LDA is to explore the hierarchical structures. Based on the idea of using nested Chinese restaurant process (nCRP) as a prior, and constructing tree paths for topics. It is inspired by the situation of a restaurant with infinite tables, a total number of n customers walked in and the i^{th} customer chooses certain table to sit down with the following probability:

$$\begin{aligned} \mathbb{P}(choose\ a\ preoccupied\ table) &= \frac{N}{i-1+\alpha} \\ \mathbb{P}(choose\ a\ new\ table) &= \frac{\alpha}{i-1+\alpha} \end{aligned} \tag{2.1}$$

where the first customer is assumed always choosing the first table (actually the order of the table does not matter). α is a scalar parameter and N indicates how many customers are sitting at the chosen table when the i^{th} customer walks in. Thus, the CRP represents a process where "the rich gets richer".

A lot of related work on the nested tree analysis have been put forward in the past years. It is generalized to the Chinese restaurant franchise mode, where the child Dirichlet process groups share mixture components for information retrieval [86]. To relieve the restriction on single root, the nested Dirichlet process (DP) uses a stick-breaking representation of the DP to distribute information across centers and thus realizes automatical clustering. Another generalization of nCRP is the nested hierarchical Dirichlet process [66], where each word is allowed to follow its own path at any level of the tree. This greedy subtree selection alleviates the single-style formulation of nCRP.

Targeting at better exploration of the meaning at semantic level, a natural extension is to combine supervised topic model with its hierarchical representation. Li et al. [52] use image tags as auxiliary information to realize image organization and annotation. Perotte et al. [68] also propose a hierarchically supervised Latent Dirichlet Allocation model for labeled bag-of-word data with a primary goal of out-of-sample label prediction. But they focus on "is-a" hierarchies to simplify the implementation, which is complementary to our model under different scenarios.

Although LDA and other related topic models have been successfully applied to short-length data, most previous work focuses on pre-training of the model to expand the length of the documents, thus are very data-dependent. For example, Hong et al. [37] proposed several schemes of aggregating individual texts into one document before training on a topic model, which is less effective if the word counts of user's posts are small and the aggregated contexts are still sparse. Zhao et al. [107] used the Twitter-LDA for topic categorization based on the assumption that each individual document can have only one topic, which alleviates the flexibility of multiple meaning capturing. Yan et al. [98] illustrate a bi-term topic model which traverses all combinations of word pairs to aggregate the whole document. This model may manually add connection between two words that are not closely related, and unnecessarily increases the computations. Besides all these applications, as far as we know, there is not much work done on the multi-level taxonomy of short text combining rich sources of supportive information. This hence inspires the proposed ai-hLDA model, which is intuitively expected to build hierarchical schemes for topic knowledge discovery based on the taxonomy support of auxiliary information.

2.3 Model

In this section, we introduce the auxiliary information based hierarchical Latent Dirichlet Allocation (ai-hLDA) model for applications where limited amounts of labeled data, the auxiliary information are available to help documents arranged along the paths of a tree. In our framework, there are two types of data: the primary information and the auxiliary information. The primary part is constituted of the document-word relations. For a collection of documents $D = (D_1 ... D_d)$, each document is represented in a bag-of-word format, with a short total length (range from $50-100$). The auxiliary part includes the relations between documents and additional attributes $T = \{T_1, T_2, ..T_m\}$, such as the authors or viewers of the documents. The possession of auxiliary T_m by document d is indicated by the binary variable y_{dt} of binary $D \times T$ matrix Y. If the value equals to 1, then the document has the corresponding auxiliary data, and vise versa.

The intuition behind ai-hLDA is based on the consideration that digitized texts are related through their rich auxiliary information. Due to the vibrant expression and short text length, correlations between the documents would not seem apparent. One tempting approach is to propagate both into a new set. However, since the text and its auxiliary information are often in different format, e.g. word occurrence and binary indicator, it is difficult to find a compromised way of representation. Since text and auxiliary information itself are bounded in a hierarchical way inherently, at levels closer to the root node, the knowledge is more general and thus more commonly shared by all documents, while at nodes closer to the leafs, the specificity increases and acts as distinctions toward topic differentiation. Incorporating a generative process for the auxiliary data themselves not only preserves the dependency, but also provides a prior bias for the path weighting to prevent meaningless topic partition as well, which will hence be expected to enhance the interpretation of their hidden semantic structures. The notation of the variables for ai-hLDA model are given in Table 2.1.

As a generative model, each node along a path of the tree represents a combinational distribution of observed samples and auxiliary data. The path is indicated by $N_l \times 1$ vectors, where N_l is the number of tree level. For example, if there are

Table 2.1: Notations

w	Observed documents with words
Y	Observed auxiliary information for documents
$[C_d]$	Paths in the tree generated from $nCRP$
θ	Proportion vector of level topics
z_d	Level index generated from $Multinomial(\theta)$
Φ	Vocabulary of words
π	Tag probabilities
$\upsilon,\ \nu$	Hyper-parameters for the stick-breaking θ
γ	Hyper-parameter for the nCRP tree
ϕ	Hyper-parameter for the Dirichlet distribution Φ
η	Hyper-parameter for the Beta distribution π

three levels along a path, then the path vector $(1, 2, 1)$ indicates that a document starts from the root to the second node of the second layer, and then chooses the first branch of the second node. The nCRP provides a good approach to construct the tree topology with no limit on the spreading size. As it is incorporated in the model, the root node for all documents will always be one. With the records of all the paths that the documents follow, we can have a general picture of how the tree looks like, and draw the structure correspondingly.

For ai-hLDA, each document is treated as bag-of-words data with available auxiliary information. Suppose there are D documents, each consists of N_d observations. $w_{d,n}$ is the n^{th} observation of document d, and the size of the vocabulary is W. Each document has a variable, indicating whether it has the label or not. The probabilities for label indicator variable are drawn from a Beta distribution. The Beta-Bernoulli process [62] is a natural choice for a label-generating distribution since the observed labels in a document are mostly represented using binary variables. Distributed over binary matrices of fixed size, it can be used to model whether there is a particular auxiliary information feature for the sample in the form of a binary matrix format. Under Beta-Bernoulli prior, for each column, a tag with weight pi is chosen.

The overall generative process for the proposed model is summarized as follows:

1. For each sample d,
    (a) Draw the paths of hierarchy $C_d \sim nCRP(\gamma)$
    (b) Draw a distribution over words $\Phi \sim Dirichlet(\phi)$
    (c) Draw level proportions θ_d from *stick-breaking* $\{\upsilon,\ \nu\}$
    (d) For each entity n (e.g. word)
        i. Draw level assignment $z_{d,n} \sim Multinomial(\theta_d)$
        ii. Draw word $w_{d,n} \sim Multinomial(\Phi_{z_{d,n})}$
2. For each tag y,
    (a) draw label probabilities $\pi_y \sim Beta(\frac{\eta}{T}, 1)$
    (b) draw $Y_{d,l} \sim Bernoulli(\pi_y)$

2.4 Inference

This section gives an overview of the approximate inference method, i.e. Gibbs sampling, used by the ai-hLDA model due to the intractability of exact inference. We adopt a technique similar to the sampling scheme in hierarchical LDA [9], where the latent variables that need to be sampled are path vector $[c_d]$, and level topic z_{dl}.

2.4.1 Sampling the Path

The path assigned to a document is influenced by the previous arrangement of the paths and the likelihood of the auxiliary information; the following equation shows the sampling probability,

$$\mathbb{P}(C_d = c) \propto \mathbb{P}(C_d|C_{-d}, \gamma)\mathbb{P}(W_d, Y_d|W_{-d}, Y_{-d}, Z, C) \tag{2.2}$$

where $\mathbb{P}(C_d|C_{-d}, \gamma)$ is the prior probability induced by nCRP (equation (2.1)), and $\mathbb{P}(W_d, Y_d|W_{-d}, Y_{-d}, Z, C, B)$ is the likelihood of observed documents and tags, i.e.

$$\begin{aligned}
&\mathbb{P}(W_d, Y_d | W_{-d}, Y_{-d}, Z, C, B) \\
\propto &\mathbb{P}(W_d | Y, W_{-d}, Z, C, B)\mathbb{P}(Y_d | Y_{-d}, W_{-d}, Z, C, B)
\end{aligned} \tag{2.3}$$

For a given path c_d, the words' sampling depends on two entities: the path and the level. If we define the entity c as the path and l as the node level, after sampling probabilities generated from a Dirichlet distribution, words will be selected from the vocabulary library Φ, using the corresponding c and l. Thus, the distribution of w is derived as follows:

$$\begin{aligned}
\mathbb{P}(W_d | Y, W_{-d}, Z, C, \pi) &= \frac{P(W|Y, Z, C, \pi)}{P(W_{-d}|Y, C, Z, \pi)} \\
&\propto \frac{P(W|Z, C, \beta)}{P(W_{-d}|C, Z, \beta)} \\
&= \prod_{k=1}^{K} \frac{\prod_w \Gamma(n_{c_{d,l},w} + \phi)}{\Gamma(n_{c_{d,l},\cdot} + W\phi)} \frac{\Gamma(n^{-d}_{c_{d,l},\cdot} + W\phi)}{\prod_w \Gamma(n^{-d}_{c_{d,l},w} + \phi)}
\end{aligned} \tag{2.4}$$

where $n_{c_{d,l},w}$ indicates the times that word w has been assigned to the node at the l^{th} level of the path with document d included. $n_{c_{d,l},\cdot}$ indicates the total number of times that all the words have been assigned to the node on the l^{th} level of document d's path. For $n^{-d}_{c_{d,l},w}$, the superscript $-d$ indicates a deduction of the number of times that w appears in current document d.

As to the tag likelihood, since the distributions are Bernoulli with a Beta prior, they are conjugate distribution pairs. Through collapse Gibbs sampling, by marginalizing out Φ, we can have the distribution of tag variables Y in the following form:

$$\begin{aligned}
&\mathbb{P}(Y_d | Y_{-d}, W_{-d}, Z, C, B) \propto \frac{P(Y|C, B)}{P(Y_{-d}|C, B)} \\
&\propto \frac{\Gamma(n_{c,t=1,\cdot} + \frac{\eta}{T})\Gamma(n_{c,t=0,\cdot} + 1)\Gamma(n_{c,\cdot,-d} + \frac{\eta}{T} + 1)}{\Gamma(n_{c,t=1,-d} + \frac{\eta}{T})\Gamma(n_{c,t=0,-d} + 1)\Gamma(n_{c,\cdot,\cdot} + \frac{\eta}{T} + 1)}
\end{aligned} \tag{2.5}$$

where $n_{c,t=1,-d}$ and $n_{c,t=0,-d}$ indicate the exclusion of current document d in counting the number of auxiliary tags being used and not being used respectivly by documents on the same path c.

2.4.2 Sampling the Topics Along the Path

For word w in document d, the probability is:

$$\begin{aligned}&P(z_{d,w}=k|z_{-(d,w)},\mathbf{c},\mathbf{w},-)\\&\propto P(z_{d,w}|z_{d,-w},m,\pi)P(w_{d,w}|\mathbf{z},\mathbf{w}_{-(\mathbf{d},\mathbf{w})},\mathbf{c},\phi)\end{aligned}\tag{2.6}$$

The first part of equation (2.6) is a stick-breaking conjugate construction over levels, where $z_{d,-w}$ indicates the level allocation in document d, with the word w excluded. From the conjugacy properties, we have:

$$\begin{aligned}&p(z_{d,w}|z_{d,-w},m,\pi)\\&=\frac{m\pi+n_{d,-w,k}}{\pi+\sum_{i=k}^{L}n_{d,-w,i}}\prod_{i=1}^{k-1}\frac{(1-m)\pi+\sum_{j=i+1}^{K}n_{d,-w,j}}{\pi+\sum_{j=i}^{K}n_{d,-w,j}}\end{aligned}\tag{2.7}$$

where $n_{d,-w,k}$ denotes the count of occurrence for word w in document d at the k^{th} level, excluding the current assignment. The second term is the word probabilities given the level and path assignments and it can be expressed as follows:

$$P(w_{d,w}|\mathbf{z},\mathbf{w}_{-(\mathbf{d},\mathbf{w})},\mathbf{c},\phi)=\frac{n_{k,c,-(d,w)}+\phi}{\sum_{w}n_{k,c,-(d,w)}+W\phi}\tag{2.8}$$

2.5 Experiments

In this section, we test the proposed model on two datasets: email corpus and course selection dataset. We analyze the results in four primary aspects: (1) auxiliary information choice elaboration; (2) perplexity comparison; (3) topic taxonomy evaluation and (4) application illustration.

2.5.1 Datasets and Comparative Models

Both datasets adopted for our experiments contain very sparse text, that is comparatively short length of each sample but large size in the vocabularies, as detailed below:

1. The Enron email corpus [1] is a publicly available dataset, which was prepared from a real organization's communication records for a period of more than three years time [22]. The contents vary from business related topics, including regulations, project progress and strategy, energy issue, internal policies and so on, to employment and logistic arrangements (such as meeting scheduling and technical support, etc). After removing numbers and meaningless stop words, the number of words for each email document ranges from 2 to 2034. We have selected two collection of documents, each with an average length of 100 and 40 words to manifest the sparseness feature, and illustrate the text length's influence on model performance. The author-recipient information is used as the auxiliary dataset.

2. The course selection dataset [106] constitutes of undergraduate records of study. Students took an average of 32 courses freely chosen from a total 2434 courses offered by different programs and departments without restriction from their majors. In our analysis, we use their overall major degree information as auxiliary labels for our analysis, which will be further addressed in section 2.5.3.

The statistics of the three datasets we used for the experiments are summarized in Table 2.2.

[1] https://www.cs.cmu.edu/ enron/

Table 2.2: Statistics of the tested datasets

Dataset	Sample Size	Average Sample Length	Vocabulary Size	Auxiliary Label Size
Email subset 1	700	40	5583	2359
Email subset 2	700	100	8190	2359
Course set	1696	32	2434	3

The other three models in this comparative study are author-recipient topic (ART) model [1], labeled LDA (LaLDA) [75] (the implementation is provided from toolbox "TMBP" [103]) and hierarchical LDA. The first two models provide flat structures through different utilization of auxiliary information while hLDA provides a tree construction for comparison. Besides LaLDA where a belief propagation (BP) method is adopted for inference as the way in the toolbox, all the other experiments use Gibbs sampling with 1000 iterations (500 for burn-in), which is enough for convergence.

To quantize how the model fits the contents, a common measure criteria called perplexity is introduced. According to the work of Blei et al. [10] and Shan et al. [83], for each word $\hat{w_i}$ in a held-out subset $\hat{w_1}, ..\hat{w_n}$of observed data, based on its log-likelihood , the complexity for the training data can be computed by:

$$Perplexity = exp(-\frac{\sum_i \log p(\hat{w_i})}{\sum_i N_i})$$

where N_i is the number of observed features, i should sum from 1 to the number of entities and $\log p(\hat{w_i})$ is the log-likelihood of the term w as per the learned model. Since larger likelihood would imply a better chance of observing the true value, with an applying of monotonically decreasing function on it, a lower perplexity would indicate a better model explanation of the data.

2.5.2 Enron Email Performance Comparison

The auxiliary data involved in the learning process are the author and recipients information. This choice is based on a special feature of email messages – possessing one sender and one or more recipients. For a company email corpus, the contents of the messages are largely influenced by the social structure in which messages are sent and received. For example, a department manager would send emails to his secretary on meeting arrangement or canceling, while the communicational contents between him and the vise president would more focus on strategic decisions, e.g. purchase plan. If two documents adopt the same personnel label, there are large chances they have topics in common. The clustering performance would hence be enhanced conditioned on the mixture of document word and auxiliary author-recipient information.

We compare topical taxonomy produced by our approach to those learned by ART and hLDA. Overall, ai-hLDA model outperforms the other two algorithms in terms of perplexity. Figure 2.1 summarizes the experimental results for sample size ranging from 200 to 600. We set the common parameter $\beta = \gamma = \eta = 1$ for all experiments. To better illustrate how the proposed model deals with corpus sparsity, another subset with documents of an average length of 100 was prepared. The perplexities all increased when the word length drops from 100 to 40, however the worsen percentage of ai-hLDA is the smallest, which indicates that our approach is less sensitive to the length of word.

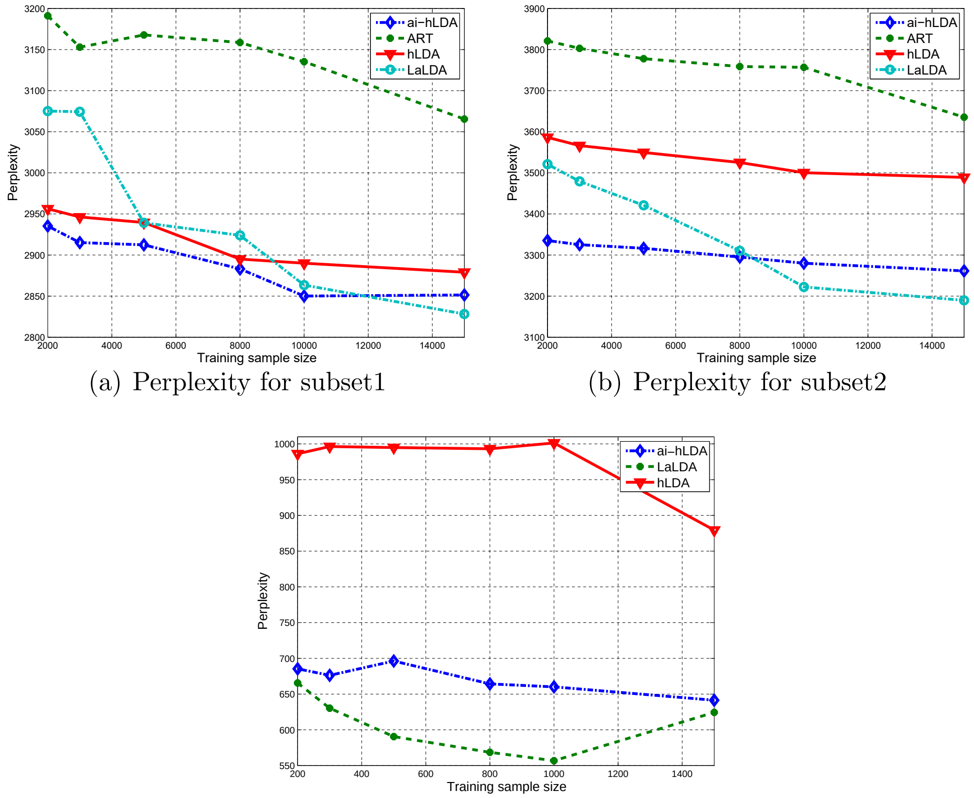
To further elaborate how ai-hLDA could provide a meaningful hierarchical grasp of the documents content through utilizing the underlying related auxiliary information, the top words assigned to nodes on part of the paths of the generated tree and words assigned to topics of ART model are examined. The whole generated tree expands to three layers with a total of 18 leaves and 10 nodes in the second layer. Ten topics generated by the ART model and sample paths generated by ai-hLDA are

Table 2.3: 10 topics inferred from Enron corpus using ART.

Topic No.	Top 10 words
1	companies way regarding better especially replace match recommendations internet presentation
2	meter needs king understanding resolve accomodate executed sitara nominated latest
3	contracts company capacity wants vince joe between global epmi sent
4 *Trade*	counterparty trading counterparties products review european trade group company
5	letter take right value understanding team process updated management told
6 *Routine Schedule*	group product people discuss upon confirm update houston better person
7	wto president world seattle use negotiations working report tuesday wednesday
8	copier copies copy including click month volume notes ready
9	mark aquila risk legal again change jennifer asked kathryn deboisblanc
10	issue taking settlement businesses determined hope options financial access already

shown in Table 2.3 and Figure 2.2 respectively. For the ART model, we manually named two typical topics. For most topics based on ART, only general words are picked out to infer, while with ai-hLDA, the root of the tree consists of the most common words in their daily communication such as “gas” and “energy” and more specific corporation or trading words are uncovered along the expanding of the tree, such as “cob” on the second layer, which is an abbreviation of California-Oregon Border. The third layer can be generally divided into two categories: one is related to external business, where corporate partner’s name Conoco Inc. etc. is occurred, and the other is more related to company’s own strategies and financial services.

Email messages are not the only short text type that can benefit from ai-hLDA model. It can be applied to other sorts of online data, if there are user link information. For example, follower information would be helpful for decreasing ambiguity of



(a) Perplexity for subset1

(b) Perplexity for subset2

(c) Perplexity for course data

Figure 2.1: Perplexity comparison on Enron Email corpus subsets and course dataset.

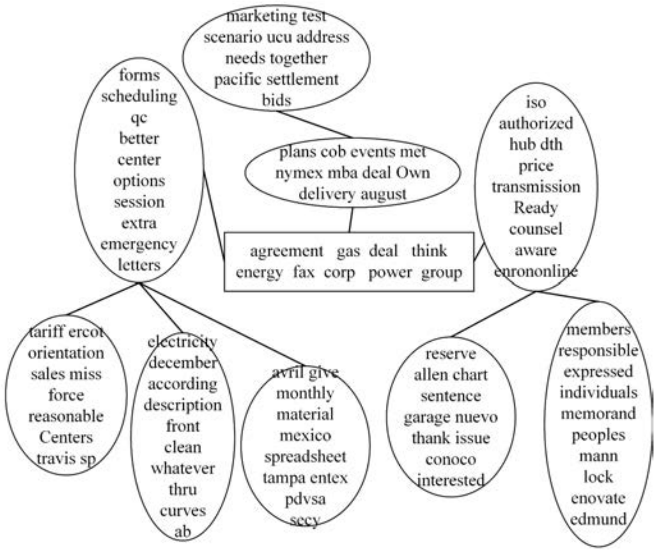

marketing test scenario ucu address needs together pacific settlement bids
plans cob events met nymex mba deal Own delivery august
agreement gas deal think energy fax corp power group
forms scheduling qc better center options session extra emergency letters
iso authorized hub dth price transmission Ready counsel aware enrononline
tariff ercot orientation sales miss force reasonable Centers travis sp
electricity december according description front clean whatever thru curves ab
avril give monthly material mexico spreadsheet tampa entex pdvsa secy
reserve allen chart sentence garage nuevo thank issue conoco interested
members responsible expressed individuals memorand peoples mann lock enovate edmund

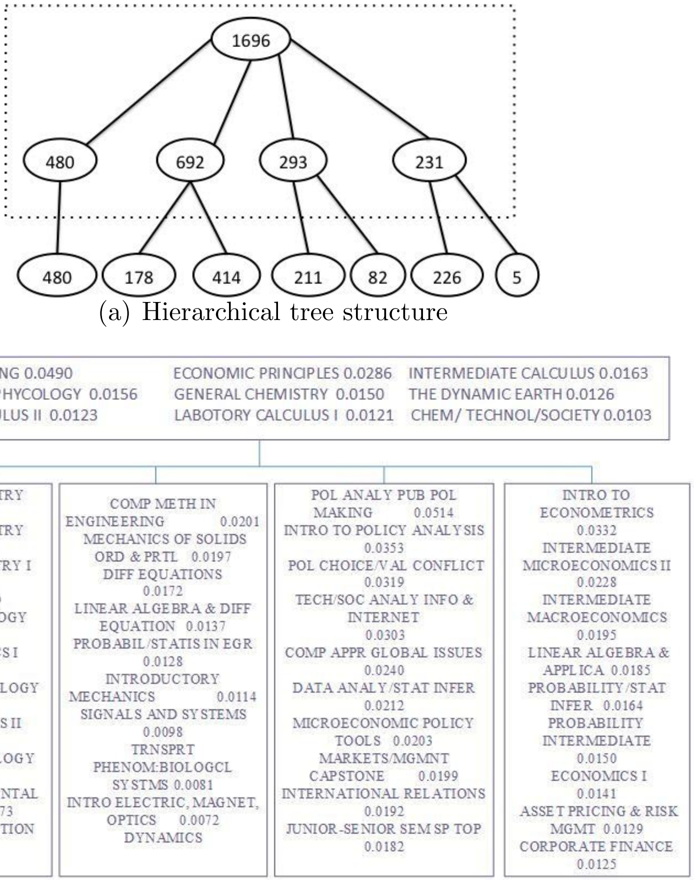
Figure 2.2: Topical tree inferred from Enron corpus shorter-length subset using ai-hLDA model.

semantics due to more exposure of shared tweets. Mutual-friend relations would also be an essential supplementary for online social gamer behavior analysis, considering the frequencies of their interacts.

2.5.3 Course Selection Analysis

Among 1696 students in the course selection dataset, over 90% of them have chosen a major. The rest, with less than length 10 academic records, left the major choice undecided or were not available. However, even for those who have fixed their program, might also choose courses outside their own departments due to interest or knowledge requirement. This uncertainty raises challenges for automatically accurate academic track detection. Considering the nature of curriculum design for different programs, we define three labels for auxiliary information – 'BS', 'BSE' and 'AB', which stand for general categories of their major program. For each record, if a student has specified the major, e.g. 'Biomedical Engineering (BME)–BSE', a binary indicator 1 will be put into the auxiliary matrix at the corresponding entity. If a student has not yet decided the major, we will not add any labels, leaving the whole row in the auxiliary matrix consisting of only zero. In this turn, we construct an auxiliary information matrix with size 1696 × 3, where 1696 is the sample size and 3 corresponds to the three major categories. There are also cases when students choose same major title under different degree type, e.g. 'Biology-BS' and 'Biology–AB'. The model is expected to use these person-conditioned topic distributions to measure similarity between the students, and make appropriate suggestions to their major or course choices.

The generated hierarchical tree structure using our method is shown in Figure 2.3(a), where the number in the circle indicates the number of students at that node. The detail course clustering of the first and second layer in dotted square is shown in Figure 2.3(b). In each block, the probabilities indicate the chances that those


1696
480
692
293
231
480
178
414
211
82
226
5
ACADEMIC WRITING 0.0490
INTRODUCTORY PHYCOLOGY 0.0156
LABOTORY CALCULUS II 0.0123
ECONOMIC PRINCIPLES 0.0286
GENERAL CHEMISTRY 0.0150
LABOTORY CALCULUS I 0.0121
INTERMEDIATE CALCULUS 0.0163
THE DYNAMIC EARTH 0.0126
CHEM/ TECHNOL/SOCIETY 0.0103
ORGANIC CHEMISTRY 0.0646
ORGANIC CHEMISTRY 0.0625
INTRO BIOCHEMISTRY I 0.0521
GENETICS AND MOLECULAR BIOLOGY 0.0521
GENERAL PHYSICS I 0.0377
GENERAL MICROBIOLOGY 0.0350
GENERAL PHYSICS II 0.0309
PRINCIPLES OF BIOLOGY 0.0184
CELL & DEVELOPMENTAL BIOLOGY 0.0173
ECOLOGY & EVOLUTION 0.0133
COMP METH IN ENGINEERING 0.0201
MECHANICS OF SOLIDS
ORD & PRTL 0.0197
DIFF EQUATIONS 0.0172
LINEAR ALGEBRA & DIFF EQUATION 0.0137
PROBABIL/STATIS IN EGR 0.0128
INTRODUCTORY MECHANICS 0.0114
SIGNALS AND SYSTEMS 0.0098
TRNSPRT PHENOM:BIOLOGCL SYSTMS 0.0081
INTRO ELECTRIC, MAGNET, OPTICS 0.0072
DYNAMICS
POL ANALY PUB POL MAKING 0.0514
INTRO TO POLICY ANALYSIS 0.0353
POL CHOICE/VAL CONFLICT 0.0319
TECH/SOC ANALY INFO & INTERNET 0.0303
COMP APPR GLOBAL ISSUES 0.0240
DATA ANALY/STAT INFER 0.0212
MICROECONOMIC POLICY TOOLS 0.0203
MARKETS/MGMNT CAPSTONE 0.0199
INTERNATIONAL RELATIONS 0.0192
JUNIOR-SENIOR SEM SP TOP 0.0182
INTRO TO ECONOMETRICS 0.0332
INTERMEDIATE MICROECONOMICS II 0.0228
INTERMEDIATE MACROECONOMICS 0.0195
LINEAR ALGEBRA & APPLICA 0.0185
PROBABILITY/STAT INFER 0.0164
PROBABILITY INTERMEDIATE 0.0150
ECONOMICS I 0.0141
ASSET PRICING & RISK MGMT 0.0129
CORPORATE FINANCE 0.0125

(a) Hierarchical tree structure

(b) Course clustering on the first and second layer

Figure 2.3: Student course selection distribution over a three layer tree

top 10 courses are chosen for each node. The tree has three layers and seven leafs. Along each possible path, each node includes the most-probable classes. We would expect the tree paths correspond to different majors while showing us some non-obvious regulations of student's course selection tendency. From the tree, we can see that the root corresponds to courses that are popular in their early year of study. The second-layer nodes briefly separate students into four categories. Comparing to hLDA, which generates a tree of 10 paths, the auxiliary information helps to get more concentrative clusters.

An important aspect of application is using ai-hLDA for accurate classification of student's major interest and hence providing sound course recommendation to them. The averaged accuracy of the first three decisive major reaches 92.3%. The

Table 2.4: Perplexity comparison. Training sample sizes range from 500 to 1500 with the left using for testing

	500	1000	1500
LaLDA	590.56	556.70	624.59
hLDA	994.87	1001.02	879.45
ai-hLDA	696.30	632.21	614.30

low accuracy for category 4 is actually quite intuitive. Since the purpose is to provide suggestions to students who have not yet decided their majors. For a total of 68 students whose auxiliary lab is 'UNDEC', 40% are classified into 'BS' programs and 51% are 'AM' programs. A comparison of perplexity among labeled LDA, hLDA and ai-hLDA is presented in Table 2.4. The ai-hLDA model has competitive performance comparing with labeled LDA, which is supervised, on perplexity, but outperforms on the classification task of labeled LDA, whose averaged accuracy is 53.7%.

2.5.4 Computational Cost

We used the synthetic dataset generated in [32] for 600 iterations. The toy dataset consists of a set of 90 images, each containing 25 pixels in a 5 × 5 grid. The computation time for different methods are compared in Table 2.5. We can see that the computational cost of ai-hLDA is similar to hLDA.

Table 2.5: Computational costs for 600 iterations (in seconds)

ai-hLDA	LDA	LaLDA	hLDA
4598.9431	2032.3218	3092.3456	4501.2322

2.6 Summary

Due to the emerging sources of digitized data, there is a growing need for analysis of texts with diverse vocabularies, rich auxiliary information but biased short sentences of contexts. Targeting this kind of sparse data, a hierarchical topic model, ai-hLDA is proposed. Through a fully conjugate Bayesian construction, this chapter puts forward a flexible way of using the supplemental/auxiliary tagging observations to guide the topical tree generation. Through a nested Chinese Restaurant Process prior, hierarchical topic patterns as latent features can be discovered for documents with short length. Experiments were conducted on two real-word text collections, with results demonstrating the clustering and predictive capability of our model.

This work is currently under review by a journal [55]. There is still room to improve this work in the future. Currently, some evaluations are hindered by the relatively high computational cost. We hope to use other inference methods to enhance the convergent speed and reduce the complexity of the model. Since the weakly supervised mechanism introduces more hyper-parameters, how to optimize them is also an important topic.

Besides the topic feature learned under the guidance of auxiliary information, it is natural to think whether the latent feature of the meta-knowledge can also be learned to enhance the understanding of the other observed data matrix. Hence, we try to design a model in a two-way collaborative filtering as further elaborated in the next chapter.

Chapter 3

Dirichlet Mixture Probit Model for Information Scarcity

3.1 Introduction

Recommender systems have changed the way people discover items on the web. How to suggest new items based on their needs and interests is an important task. To model the interaction between users and items, we must understand the hidden facets and the high level topics of their favors. This kind of topics can be obtained collaboratively from diverse sources, such as clicking history, rating scores or objective reviews, and often have comprehensive characteristics. For example, two users may have similar preferences towards detective films, but different viewing samples of action movies due to their different preferences of actors. Hence, how to model these hidden factors is a key to obtaining satisfactory recommendation performance [61].

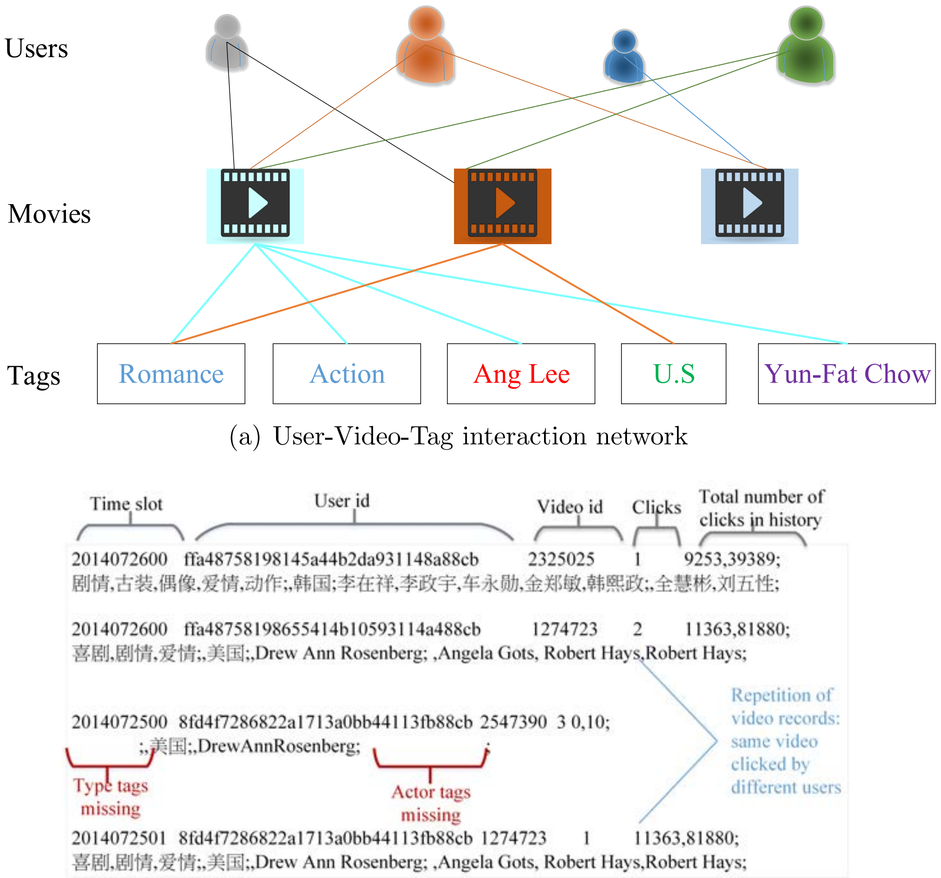
Recommender systems can be roughly divided into two categories: collaborative filtering (CF) and content-based collaborative filtering (BCF). CF predict additional topics or products to new users based on their past preferences [85]. CF algorithms are required to be capable of dealing with challenges, including highly sparse data, increasing numbers of users and items, data noise and missing issues.

Besides CF techniques, another important class of recommender systems is content-based filtering. This type of recommender systems make recommendations by analyzing the content of item information and finding regularities in the content. The main difference between CF and CBF is that CF only uses the user-item interactions to make predictions, while content-based recommender systems rely on the features of users and items.

Both CF and CBF systems have limitations. For CF, the similarity values are based on common items and therefore are unreliable when data are sparse and the common items are few. On the other hand, while CF systems do not explicitly incorporate feature information, CBF systems do not necessarily incorporate the information in preference similarity across individuals [2]. Hence, how to construct a hybrid system that combines both user-item, as well as item-feature information is the start point of our consideration.

As to online video recommendation, a typical database stores the historical viewing samples of active users, with each sample referring to one video clicked. Many past researches relying on human feedbacks are typically in a text review form, with the goal of predicting rating scores for unseen items [46]. However, this kind of holistic information for contents is not always available. Fortunately, with increasing availability of e-commerce sources, new recommender tasks are not only based on traditional aspects regarding contents and users, but also different kinds of meta knowledge such as tags. Taking the records from Tencent QQ Browser as an example, there are four categories of information provided as video word tag (in Chinese), namely, type, region, director and actor (Figure 3.1). As understanding the topic factors is helpful for justifying user preference, the challenge hence lies in discovering both user-item correlation and user preference topics represented by this tag-form text in a single learning stage.

The topical modeling based recommender system encounters two aspects of sparseness. First, users concentrate on a relatively small number of items comparing



(a) User-Video-Tag interaction network

```
2014072600   ffa48758198145a44b2da931148a88cb       2325025     1      9253,39389;
剧情,古装,偶像,爱情,动作;,韩国;李在祥,李政宇,车永勋,金郑敏,韩熙政;,全慧彬,刘五性;

2014072600   ffa48758198655414b10593114a488cb       1274723     2      11363,81880;
喜剧,剧情,爱情;,美国;,Drew Ann Rosenberg; ,Angela Gots, Robert Hays,Robert Hays;


2014072500  8fd4f7286822a1713a0bb44113fb88cb 2547390  3 0,10;
           ;,美国;,DrewAnnRosenberg;                     ;

2014072501  8fd4f7286822a1713a0bb44113fb88cb 1274723     1      11363,81880;
喜剧,剧情,爱情;,美国;,Drew Ann Rosenberg; ,Angela Gots, Robert Hays,Robert Hays;
```

Time slot
User id
Video id
Clicks
Total number of clicks in history
Repetition of video records: same video clicked by different users
Type tags missing
Actor tags missing

(b) A clip of click log. For videos clicked by different users, the tag words repeat. There are some missing issues with certain videos.

Figure 3.1: Sample samples and their corresponding interactions obtained from Tencent QQ browser.

to the large inventory [30]. Due to the huge volume of users, videos clicked everyday have features of being extremely dynamic and long tailed. In this case, traditional Collaborative Filtering (CF) method tends to recommend popular items rather than those cold ones which might actually correspond to the taste of the target users [85]. So a content-based recommender model would be a better solution to handle this type of sparseness. As shown in Figure 3.2, when videos are sorted from the most popular to the least ones, with normalized rank between 0 and 100, 18% of the top popular videos account for about 80% of the user clicking records. Yin et al. describe this “long tail” phenomenon of niche products as a significant generator of revenue if

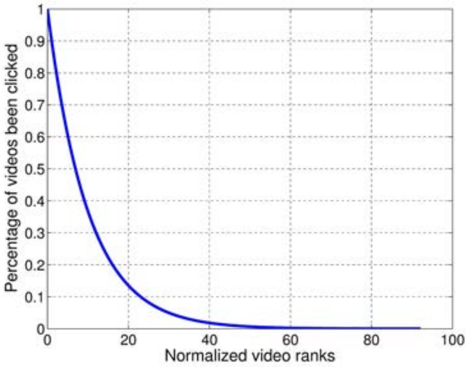


Figure 3.2: Relation between videos and clicking samples. Around 80% clicking samples are from 18% top popular videos

we could dig out the niche tastes of users [100]. This coincides with the importance of accurate description of user preference through explicit collection or prediction by the system.

The topic model based recommender system encounters two aspects of sparseness. First, users concentrate on a relatively small number of items comparing to the large inventory [30]. Due to the huge volume of users, videos clicked everyday have the features of being extremely dynamic and long tailed. In this case, traditional Collaborative Filtering (CF) method tends to recommend popular items rather than those cold ones which might actually correspond to the taste of the target users [85]. As shown in Figure 3.2, videos are sorted from the most popular to the least ones, with normalized rank between 0 and 100. 18% of the top popular videos account for about 80% of the user clicking samples. Yin et al. describe this "long tail" phenomenon of niche products as a significant generator of revenue if we could dig out the niche tastes of users [100]. This coincides with the importance of accurate description of user preference through explicit collection or prediction by the system.

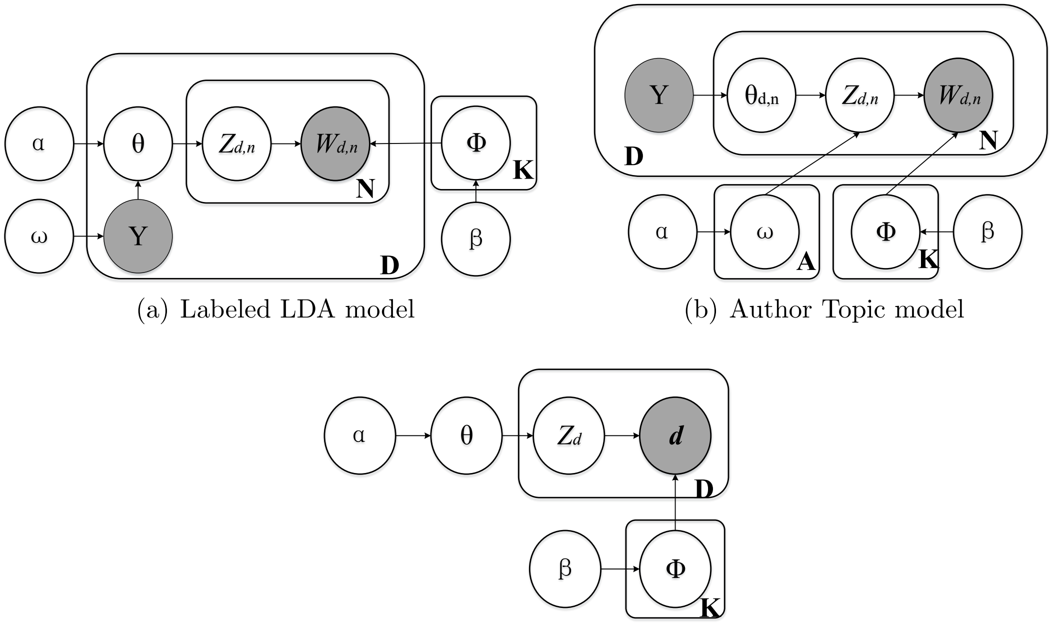
Second, the associated words are scarce. Since the textual features are extracted from product descriptions, no attributes with well-defined values could be found due to the unstructured nature of the data [77]. Thus, it is a difficult task to categorize each item with just one- or two-keyword based string matching. For topic model based filtering, one common practice is to treat item feature as bag-of-word. Models such as DIGTOBI [43] and TopRec [104] have been proved give satisfying results on recommender framework. However, one limitation of direct topic modeling is that holistic information for contents, such as full articles [23] or tags from all aspects [21] are often required but cannot always be acquired. In our application, since the associated word length for each video is not long, treating each record as a single document indicates a considerably large vocabulary size but a relatively small amount of context per document. Such an information scarcity (IS) problem poses challenges for conventional text mining topic model such as Latent Dirichlet Allocation (LDA), which depends on co-occurring words for topic discovery.

In this chapter, we address the aforementioned challenges by proposing a unified Dirichlet mixture probit model for information scarcity (DPIS). It learns topics over scarce information by directly modeling the generation of records. Specifically, the clicking history of a user is assumed to consist of a mixture of topics, and each record sample as a whole is assigned a specific topic with certain probability, from which its corresponding tags are generated. In the generative model, the words describing each item are drawn from the vocabulary of the record. The topic proportions, formulated as a multinomial vector, are also used as an *item vector* to describe whether a user viewing the item or not. Although DPIS does not model the user-level generative process, the link between active users and video pools can be learned through a probit model, with a Laplacian prior enforced on the sparse parameters. Hence, our work differs from standard approaches, as the *item vectors* now serve two roles: explaining both the words that tag the video record; and capturing the collaborative component. In this way, the overall model can not only cluster the semantic topics of the video records based on aggregated item features for active users, but can also combine the observed features and generates topics for newly synthesized data. Here, we employ a collapsed Gibbs sampling inference to find the posterior solution to the topic discovery and the probit parameters, which is convenient to implement, and conduct experiments on real-world data collections.

3.2 Related Work

In this section, we briefly review the related work of this study. For topic exploration, one typical approach is through topic modelling. Topic models are originally designed for text mining tasks, including latent topics exploration and document clustering [10]. The key idea is to generate a topic vector that can describe the desired relational proportions, e.g. in Latent Dirichlet Allocation (LDA), the vector represents the probabilities of words belonging to the topics. Due to their flexible extendibility, topic models have been successfully leveraged in recommendation [93] [65]. The state-of-the-art work on LDA provides a fundamental framework for many of the following topic models. In LDA, each document is viewed as a mixture of latent probabilistic topics, and the words in that document could be a representation of certain subsets of those topics. However, even when certain resources exist, e.g. document labels or side information (author, venue etc.), it fails to provide any suitable tools to tune the generated topics [75].

The labeled LDA model extends LDA by defining a correspondence between latent topics and user tags for learning word-tag correspondences [75], and has been applied to Twitter posts modeling [74]. Graphically represented in Figure 3.3 (a), a Multinomial distribution is selected over the label sets, with every word in the document picked from the corresponding document labels according to the likelihoods of document-label and label-word selection. To incorporate the user information, Author Topic (AT) model [78] [63] aims to learn the hidden topics of the documents, as well as the clustering of authors. Author interests on these topics can be regarded



(a) Labeled LDA model

(b) Author Topic model

(c) Dirichlet Mixture model

Figure 3.3: Graphic representations of related topic models.

as a two-way filtering, where the side information of authors is linked to each word through a uniform distribution, and each author is associated with a symmetric Dirichlet distribution over topics (Figure 3.3 (b)). Both models, based on the LDA mechanism, rely heavily on the word co-occurrence.

Targeting the feature sparsity issue, several related topic models have been successfully applied to short-length texts, where the co-occurrence patterns of words are not obvious. Zhao et al. use the Twitter-LDA for topic categorization based on the assumption that each individual document can have only one topic, which alleviates the flexibility of multiple meaning capturing [107]. Work done by Yin et al. [101] is most related to our model, as the proposed collapsed Gibbs Sampling algorithm for the Dirichlet Multinomial Mixture model (abbr. to GSDMM), estimates the mixture component for each document targeting the short text clustering problem. However, GSDMM also limits each document to express only one topic.

3.3 Problem Formulation

Our aim is to incorporate user preference into the sparse feature based semantic learning. The first straight forward approach is to extend two-level, document-word hierarchy into a three-level, author-document-word hierarchy. The following analogies can be made for video recommender system: user to author, sample to document, and tag to word. In this way, the application scenario can also be regarded as a special case of AT model, with only one author belonging to each document. Specifically, a video tag count matrix $W \in \mathbb{Z}_+^{S \times V}$ can be obtained, where $\mathbb{Z}_+$indicates positive integer, S is the sample sample size, V is the size of word tag vocabulary. There is also a binary user-sample matrix $Y \in \{0, 1\}^{D \times S}$, indicating which user each sample belongs to, where D is the number of active users. With an averaged four million samples coming in daily and a total of over $500,000$ evolving tag words, the following statistics are reported: the tag word length is usually less than 30, leaving the matrix W with a sparsity of 99.2%. Moreover, 94% of the users have fewer than 20 clicking samples, so the word expressions and video sets are extremely vibrant comparing to its library. In such case, when attempting to incorporate the item attribute (tag word) into finding the shared latent low-dimensional space between users and items, how to appropriately explore the hidden interactions behind the scarce item features becomes a key challenge.

We tackle this information scarcity from a semantic learning prospective, regarding user preferences as combinations of topics. For example, some users may have interest in a video due to its plots, while others may favor it according to the director or actors. The personal preference is hence collaboratively expressed by the combination of tags. Moreover, different from the normal text topic mining, most tags occur only once, which makes the Term Frequency-Inverse Document Frequency (TF-IDF) measure lose effectiveness. Semantic information, therefore, explored

Table 3.1: An example of user sample

user	video	tag words	clicks
u_1	v_1	horror suspense	10
	v_2	comedy romance drama	1
	v_3	action comedy drama	1
u_2	v_2	comedy romance drama	2
	v_3	action comedy drama	1

from single sample is limited. Thus, effective capturing of tag correlation becomes the primary concern during the interaction mining processes.

As each sample belongs to only one user, in the DPIS metaphor, behaviors of each user can be regarded as a document, while each sample is the sentence constructed by tag words. Thus, the *author (document)-sentence-word* hierarchy not only builds linkage between user and video, but also preserves the correlations among tags belonging to the same video. To further elaborate our intuition for the proposed hierarchical model, we start with some exploratory analysis of the data to illustrate why traditional topic models, i.e. LDA would tend to fail in this scarceness characterized application. Consider a set of two users $U = \{u_1, u_2\}$ and a set of three videos $V = \{v_1, v_2, v_3\}$ in Table 3.1. Each video has its corresponding tags W_{v_i} (to make the table clear, only 2-3 words are included). Traditional topic models utilize word co-occurrences for topic assignment probability calculation. Inferred from the last column of Table 3.1, the highest frequency of word-pair is dominated by the ones from v_1. Even though $\{v_2, v_3\}$ would be a common topic intersection, the recommendation would still bias towards the suspense type, and overlooks users' interest in the other type, i.e. comedic drama.

In comparison, if a clicking sample is treated as a whole, it is obvious that $\{v_1, v_2\}$ and $\{v_1, v_3\}$ will each co-occur once, while $\{v_2, v_3\}$ will appear twice. Through this kind of aggregation, the biased co-occurrence frequencies are equilibrated and the correlation among tags in the same sample are maintained. Using the sample-level co-occurrence frequency as the corresponding tag co-occurrence frequency will hence enhance the latent topic discovery for items and user preferences representation. To preliminarily justify the advantage of concatenating samples of each active user, we conduct Pareto test with 10 pre-specified factors. Percentage of the data expressed by the first factor increases from 10 to 78 when samples from the same user are concatenated, which means that this approach indeed improves the interpretation of the data.

3.4 Approach

3.4.1 A Compound Dirichlet Mixture Model

Our compound model, which is a probabilistic generative model, as graphically shown in Figure 3.4, combines both user-item information and item-feature level clustering in the representation process. There are three assumptions for the generative process of the model: (1) user collection is generated by a topic mixture model; (2) there exists one-to-one correspondence between mixture components and clusters; (3) tag words for each sample are generated simultaneously according to the probability of sample-level topic assignment. Its hierarchical Bayesian structure is characterized by hyper-parameters$\{\gamma,\ \alpha,\ \beta\}$ and column vector sets $\{\theta_s,\ \mathbf{t}_d,\ \omega_d\}$. The shaded nodes denote two observations. The key difference comparing to other topic models is that, each sample of individual user, instead of each word, is generated from a mixture component, topic k, according to the weights of topics.

The observed data are the viewing samples collection of active users. Each sample has the information of its video id and user id, and is stored in a bag-of-words format. The same set of tag words can repeat multiple times since each video could be viewed by the same person or other users multiple times. We assume that there exist K hidden topics in a collection of video watching samples. To deal with the

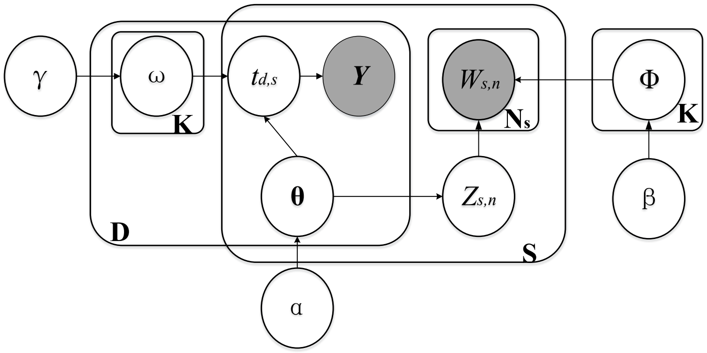


Figure 3.4: Generative process of the DPIS model.

sparseness issue of word tags, each sample is assumed to express one topic, without separately considering each individual word tag. Specifically, we consider that the clicking samples belonging to one user as a mixture of topics, where each sample as a collection of tags is drawn from a topic capturing one type of user preference independently. So the tag word in each sample is drawn with the same probability from the topic independently. In this approach, DPIS is capable of capturing the strong correlations of tag words in the same sample, and hence represents the latent semantics of a sample as a whole.

The generative process for the model is summarized as follows:

1. For each sample s,
   - Draw topic proportions $\theta_s \sim Dir(\alpha)$
   - Draw topic assignment $\mathbf{z}_s \sim Multinomial(\theta_s)$
   - For each word in the sample s,
     - ○ Draw word $\mathbf{w}_s \sim Multinomial(\Phi_{\mathbf{z}_s)}$
2. For each user d,
   - Draw the parameters $[\omega_d]_{K\times 1}$ by

$$\omega_{\mathbf{d}} \sim \mathcal{N}(0, \gamma^{-1}\mathbf{I}_K) \tag{3.1}$$

where $\mathbf{I}_K$ represents the $K \times K$ identity matrix.

3. For each user-sample tuple $y_{d,s}$, draw the indicator by

$$y_{d,s} = \begin{cases} +1, \ if \ t_{d,s} \geqslant 0, \\ -1, \ otherwise, \end{cases}$$

$$t_{d,s} \sim \mathcal{N}(\omega_d \theta_s^{\mathrm{T}}, 1) \tag{3.2}$$

Based on the assumption that tag words in a sample are generated independently when the topic assignment for the sample is known, we can write the probability for sentence generation as:

$$p(s|z = k) = \prod_{w \in s} p(w|z = k) \tag{3.3}$$

For $W \in \mathbb{Z}_{S \times V}$ being a video tag count matrix, with S equal to sample size and V sample vocabulary size, each row of W represents a user sample, with its corresponding tag words. The response matrix Y is represented by a latent matrix $T \in \mathbb{R}_{D \times S}$, where D is the number of users. θ acts as a shared matrix between topical clustering and response classification. Thus the latent matrix T and the probit link can be generated according to equation (3.2). This kind of prior together with the probit link are similar to [105], but in our application, it combines with semantic topic modeling and is utilised for recommendation.

For better understanding DPIS method, we compare it with two other typical models for topic learning, i.e. LDA and Dirichlet Multinomial Mixture (DMM) model. When applying LDA and related topic models, a form of K-factor latent matrix factorization is carried out, generally defined as

$$\mathbf{W} = \mathbf{\Theta} \mathbf{\Phi}^{\mathrm{T}} \tag{3.4}$$

where $\mathbf{\Theta} \in \mathbb{R}^{S \times K}$, $\mathbf{\Phi} \in \mathbb{R}^{V \times K}$. D is the number of documents and V is the vocabulary of words. Its factorization relies on the Dirichlet prior placed on the columns of Θ [38]. For DMM model, the restriction on corpus-level topic assignment is strong.

Specifically, each document can only have one topic and all the words in the document can only be sampled from the same topic. If the sample collection of a user is analogized to a document, samples can be analogized to multi-length phrases. Thus, by breaking documents into phrases, DPIS overcomes the data sparsity problem of LDA by drawing topic assignment from the phrase-level and also alleviates the topical restrictions of DMM for overfitting avoidance.

3.4.2 Prediction and Recommendation

Recall that the goal of our analysis is to quantify the advantage of concatenating samples user-wisely. Through the discovery of hidden semantics, a classifier can be trained based on the sharing information among the samples. For a new sample s with its corresponding words W_{new}, classification to a binary indicator is made using the maximum a posteriori (MAP) rule as follows

$$\begin{aligned} y_d^* &= \arg\max_{y_d} P(y_d|\mathbf{W}_{new}) \\ &= \arg\max_{y_d} P(y_d, \mathbf{W}_{new}) \\ &\approx \arg\max_{y_d} P(y_d|t_d^*) \int_{t_d} P(t_d|\mathbf{W}_{new}) dt_d \\ &= \arg\max_{y_d} P(Y_d|t_d^*), \end{aligned} \tag{3.5}$$

where

$$t_d^* = \arg\max_{t_d} P(t_d|\mathbf{W}_{new}) \approx \arg\max_{t_d} P(t_d, \mathbf{W}_{new}). \tag{3.6}$$

To estimate t_d^* , the marginal likelihood is given by

$$P(t_d, \mathbf{W}_{new}) = \int_\Phi \int_\omega \int_\theta \sum_Z P(\mathbf{W}_{new}, t_d, Z, \theta, \omega, \Phi) d\theta d\omega d\Phi. \tag{3.7}$$

Since the above integral is intractable, after marginalizing out Z, the MAP estimation procedure approximates the integral by using point estimates of $\{\theta, \omega, \Phi\}$.

Table 3.2: Notations

K	pre-specified number of topics
$\mathbf{z}$	topic labels for each sample $k = 1...K$
$\mathbf{s}$	samples collection
V_s	number of words in the sample vocabulary
N_s	total number of samples
I	number of iterations
n_k	number of samples in cluster k
$n_k^{w_s}$	number of occurrences of word w in cluster k in sample s

3.5 Inference

In this section, the inference part of DPIS using collapsed Gibbs Sampling algorithm is formally derived.

3.5.1 Gibbs Sampling for DPIS

Given samples with observed tag words and corresponding response variables, the inference task is to find the posterior distribution over: the topic structure including topic ϕ_k and the latent probit parameter $\mathbf{t}_d$ for each active user. Since exact tracking is not available, we do the estimation using Gibbs sampling. The detail updating steps of the algorithm is shown in Algorithm 3.1. The notations of the parameters used throughout this chapter are shown in Table 3.2. The primary conditional distribution is the sample level topic distribution. For initialization, we randomly assign samples to K clusters, sample the initialized values for the number of n_k and $n_k^{w_{d_s}}$. Then, we traverse all the samples in the user watch list for I iterations. In each iteration, we reassign a topic for each sample according to the conditional distribution. After every updating of topic, we re-sample the information accordingly.

3.5.2 Derivation

Update of Latent Topic Distribution: First we define the conditional density of a sample in a collection of the d^th user being assigned to topic k given all other

Algorithm 3.1 DPIS

```
**Input:** User sample collection $D$, topic number $K$, $\alpha$ and $\beta$
**Result:** Topics for each sample $z$
Begin
Initialize $n_k$, $n_k^{w_s}$
**for** each sample $s$ of all user **do**
    Sample a topic $z$ for $s$
    $n_k = n_k + 1$ and $n_k^{w_s} = n_k^{w_s} + 1$
**end for**
**for** iteration $i$ **do**
    **for** each sample $s$ **do**
        Locate its current topic $z$
        $n_k = n_k - 1$ and $n_k^{w_s} = n_k^{w_s} - 1$
        Sample a topic for s
        $n_k = n_k + 1$ and $n_k^{w_s} = n_k^{w_s} + 1$
    **end for**
    Update $\mathbf{t}$, $\omega$
**end for**
```

assignments as

$$p(z_s = k|\mathbf{z}_{\neg s}, \mathbf{s}) \propto \frac{p(\mathbf{z}, \mathbf{s}|-)}{p(\mathbf{z}_{\neg s}, \mathbf{s}_{\neg s}|-)} \tag{3.8}$$

where the symbol $\neg$ indicates the item it pointed is deducted from calculation. Hence, $\neg s$ indicates that current sample s is excluded from counting. From the generative process, we can get:

$$p(\mathbf{z}, \mathbf{s}|-) = p(\mathbf{z}|\alpha) \times p(\mathbf{s}|\mathbf{z}, \beta) \tag{3.9}$$

The first part indicates the likelihood of sentences being assigned to the k^{th} topic. In a collapsed way, we can obtain:

$$p(\mathbf{z}|\alpha) = \int p(\mathbf{z}|\mathbf{\Theta}) p(\mathbf{\Theta}|\alpha)\, \mathrm{d}\mathbf{\Theta} \tag{3.10}$$

Thus,

$$p(\mathbf{z}|\alpha) = \frac{\prod_{k=1}^{K} \Gamma(n_k + \alpha)}{\Gamma(\sum_{k=1}^{K}(n_k + \alpha))} = \frac{\prod_{k=1}^{K} \Gamma(n_k + \alpha)}{\Gamma(N_{\mathbf{s}} + K\alpha)} \tag{3.11}$$

For $p(\mathbf{s}|\mathbf{z}, \beta)$, assume that each tag word in the s^{th} sample is generated independently with the topic assignment probability, then:

$$
\begin{aligned}
p(\mathbf{s}|\mathbf{z},\beta) =& \prod_{s=1}^{N_s}\prod_{w\in s} p(w|z=k) \\
=& \prod_{s=1}^{N_s}\int\prod_{w\in s} p(w|z,\Phi)p(\Phi|\beta)\,\mathrm{d}\Phi \\
=& \prod_{w_s=1}^{V_s}\frac{\Gamma(n_k^{w_s}+\beta)}{\Gamma(\sum_{w=1}^{V_s} n_k^{w_s}+\sum_{w=1}^{V_s}\beta)}
\end{aligned}
\tag{3.12}
$$

Similarly, we can get these two probabilities excluding the current sample. Substituting them into equation (3.8), we can have:

$$
\begin{aligned}
p(z_s=k|\mathbf{z}_{\neg\mathbf{s}},\mathbf{s}) =& \frac{\prod_{k=1}^{K}\Gamma(n_k+\alpha)}{\Gamma(N_{\mathbf{s}}+K\alpha)} \\
&\times\frac{\Gamma(N_{\mathbf{s}}-1+K\alpha)}{\prod_{k=1}^{K}\Gamma(n_k-1+\alpha)} \\
&\times\prod_{w_s=1}^{V_s}\frac{\Gamma(n_k^{w_s}+\beta)}{\Gamma(\sum_{w=1}^{V_s} n_k^{w_s}+\sum_{w=1}^{V_s}\beta)} \\
&\times\prod_{w_s=1}^{V_s}\frac{\Gamma(\sum_{w=1}^{V_s} n_k^{w_{\neg s}}+\sum_{w=1}^{V_s}\beta)}{\Gamma(n_k^{w_{\neg s}}+\beta)}
\end{aligned}
\tag{3.13}
$$

Use the property of Γ function, we can further simplify the equations as:

$$
\begin{aligned}
&p(z_s=k|\mathbf{z}_{\neg\mathbf{s}},\mathbf{s})\propto \\
&\frac{n_k-1+\alpha}{N_s-1+K\alpha}\frac{\prod_{w_s=1}^{V_s}\prod_{i=1}^{n_k^{w_s}}(n_k^{w_{\neg s}}+i+\beta-1)}{\prod_{i=1}^{n_k^{w_s}}(\sum_{w=1}^{V_s} n_k^{w_{\neg s}}+i+\sum_{w=1}^{V_s}\beta-1)}
\end{aligned}
\tag{3.14}
$$

Update of Probit Classifier: Using the defined priors (equations (3.1) and (3.2)) for the hierarchical Bayesian construction, the conjugate posterior distribution can be inferred straight forwardly as

$$
t_d\sim\mathcal{N}(\omega_d\theta^{\mathrm{T}},1) \tag{3.15}
$$

where $\mathcal{N}$ stands for truncated normal distribution.

$$\omega \sim MVN[(\theta^{\mathrm{T}}\theta + \mathbf{I}_K^{-1})\theta^{\mathrm{T}}\mathbf{t}^{\mathrm{T}}, (\theta^{\mathrm{T}}\theta + \mathbf{I}_K^{-1})^{-1}] \tag{3.16}$$

3.6 Experiments

Let us introduce our data in detail and report our experimental results in this section. To provide recommendation evaluation in a formal and rigorous way, we examine the quality of the generated category and prediction using DPIS model in both warm-start and cold-start situations. The performances with different topic number K are reported. We consider assessing our model in how it works comparing to direct tag match, traditional topic model and topic model for short texts. So, baseline methods including tagommender [82], and five other probabilistic models: Latent Dirichlet Allocation (LDA), Author Topic (AT) Model, labeled LDA (LaLDA) [75], fast discriminant LDA (DLDA) [83] and collapsed Gibbs Sampling algorithm for the Dirichlet Multinomial Mixture model (GSDMM) [101] are considered.

The Gibbs sampling algorithm in section 3.5 is implemented for the proposed DPIS model. In our experiments, 1000 iterations for burn-in were performed for the training part, and every 50 collected samples were taken for 1500 iterations. As proved in [69] that LDA is not directly suited for short text, we expanded the tag words of each sample using the number of times it has been clicked in the past five days and noted it as 'LDAexpanded'. We consider using the topic distributions learned by LDA as features for following recommendation. The similarity between videos are first measured using KL distance [49] based on the posterior probabilities for topic assignment θ, and the predicted rating $P_{u,v}$ with respect to an active user u is based upon the weighted average score from all the samples that have been viewed. For AT model, besides the probabilities of tag to topics, we also get user-topic matrix for all users. Similar to LDA based recommendation, user similarity is measured through KL distance based on user-topic matrix. According to the similarity score, we then rank the top videos clicked by similar users incorporating the similarity of the video topics. We use the implementation of labeled LDA provided from toolbox"TMBP" [103] with a belief propagation (BP) method adopted for inference. The user information is utilized as the labels for training. To compare with direct tag match approach, we adopt the idea from Tagommender model [82] for comparison, through first applying tf-idf to all the tag words of the samples, and then matching the candidate video tags that have the highest scores. We also implemented the GSDMM [101] as well. It applies the Dirichlet Multinomial Mixture (DMM) model for short text clustering. The recommendation ranking is similar to our model and LDA, however due to the assumption of GSDMM, each user can only belong to one latent topic.

3.6.1 Data and Experimental Setting

We use samples in three consecutive weeks at the same time slot from Tencent QQ browser. Each sample consists of the video id, the user id, and four categories of word tags: (1) type (e.g. action, mystery); (2) region (e.g. US, main land China); (3) director and (4) actor. There are $4,071,811$ samples in total with $137,465$ unique users, and $9,393$ unique videos they clicked. For each of the recommendation scenarios, we generated the test data sets as follows.

- Warm-start Test: For warm-start recommendation, both active users and the candidate videos have occurred before. We select a subset of users who have over 10 historical samples, which generates a sample collection of $335,783$ samples with $42,652$ distinct word tags. This is also inferred to as In-matrix prediction [93], and is similar to traditional collaborative filtering.

A 5-fold cross-validation is used for evaluation. To guarantee that all the users and items have already appeared in the training set, for samples that have been

Table 3.3: A comparison of accuracy performance

Distinct Video Size	100	500	1000	5000
DPIS	0.9523 ± 0.0032	0.9505 ± 0.0103	0.9546 ± 0.0084	0.9550 ± 0.0118
MLR-LDA	0.9510 ± 0.0100	0.9417 ± 0.0087	0.9401 ± 0.0114	0.9374 ± 0.0169
LaLDA	0.9110 ± 0.0025	0.9023 ± 0.0112	0.9198 ± 0.0021	0.9503 ± 0.0010
Fast DLDA	0.9710 ± 0.0005	0.9743 ± 0.0032	0.9698 ± 0.0011	0.9563 ± 0.0003

viewed by each user for over five times, we distributed at least one of them into each fold. For samples that have appeared less than five times ($246,642$ in total), we always put them into the training fold. For each fold, we train the topic clustering parameters of the model and fit the probit prediction part for testing fold. The list of top n recommended videos is based on the prediction probit matrix.

• Cold-start Test: For the three types of cold-start problems mentioned in [43], we focus on the second type for quantitative comparison, that is in the testing set there are newly added videos not exposed before since this is a very common situation for online video recommendation where every day has new videos fed into the collection. Still, 5-fold cross-validation is adopted. We first group videos into five folds. Each time for the test fold, only users that have appeared in the four training folds are considered. We also provide evaluation under the situation where there are new users with no previous historical samples coming in. Since DPIS does not consider user attributes, it is difficult to directly handle this kind of cold-start. Experiments are done in a vertical time direction, where at first, some top recommended videos are generated based on the probability from the training set. Gradually we add in samples of the new users and examine how the model performs with gradual increase knowledge of the user preference.

3.6.2 Warm-start Scenario Evaluation

We start with the warm-start cases, where both video and users in the testing set have appeared in the training sets. We compare DPIS with two other supervised topic models on classification accuracy as a function of the number of positive responses. As proved in [83], fast discriminant LDA (DLDA) performs better than DLDA and fast LDA, we choose it as a state-of-art model. The results presented in Table 3.3 are averaged over 5 random initializations. For each test, the number of unique videos being watched ranges from 100 to 5000, leaving the item sparsity from $65\% - 92\%$. It shows that the proposed DPIS model performs better than labeled LDA (LaLDA) [75], and has competitive accuracy results comparing to fast DLDA. Since fast DLDA adopts supervised multi-label logistic regression for label classification, the number of classes is allowed to be different from the topic number in the generative model. This would contribute to the better performance comparing to DPIS with extra tuning of the class number parameter. Moreover, the performance of DPIS is consistent throughout all testing cases, which suggests that this model tackles sparsity well.

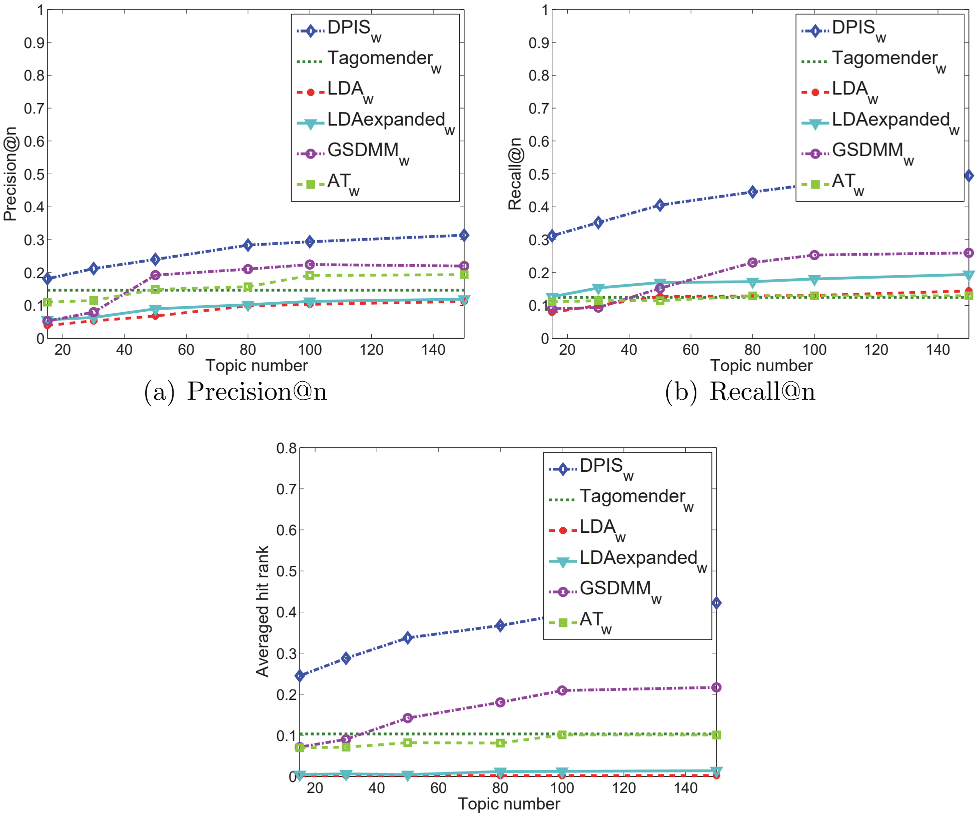
Influence of topic number

To evaluate the model quantitatively, we also investigate the influence of topic number K on the performance of recommendation. Using common metrics for recommendation evaluation, here we define precision, recall with n candidates and averaged hit rate as:

$$Precision = \frac{N_r}{n}$$

$$Recall = \frac{N_r}{N_t}$$

$$Averaged\ hit\ rate = \frac{\sum_i^{N_r} \frac{1}{p_i}}{N_t}$$



(a) Precision@n

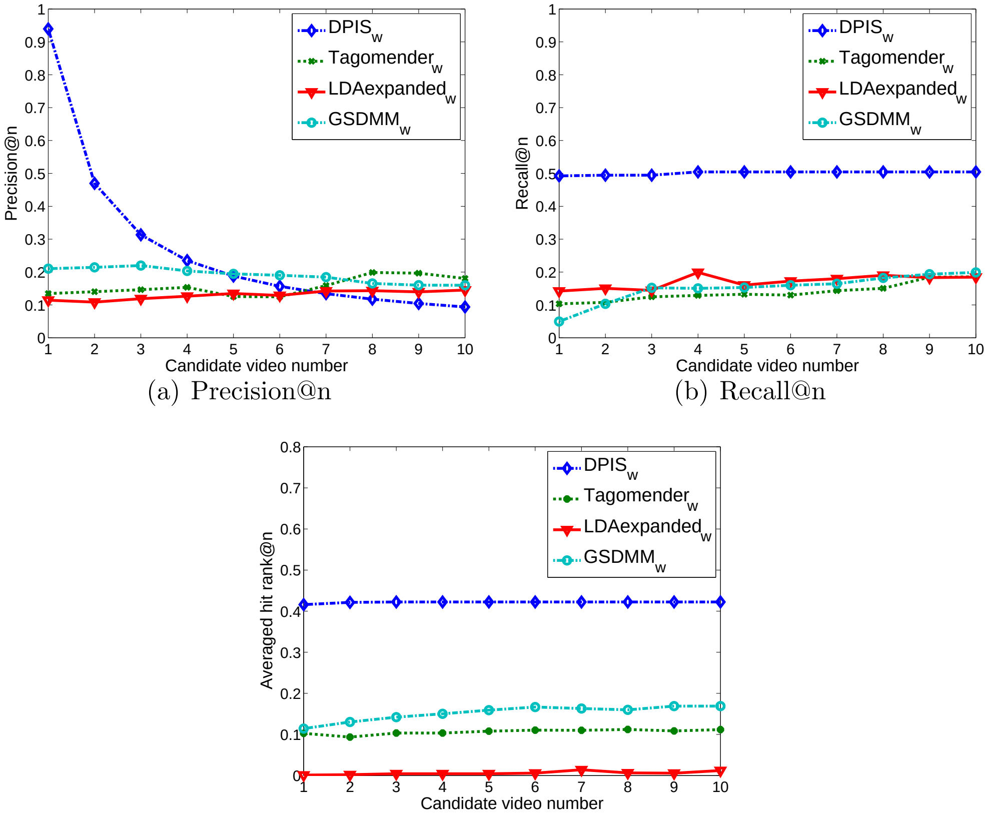
(b) Recall@n

(c) Averaged hit rate

Figure 3.5: Vary topic number K from 15 to 150 with n equal to 3 for 5-fold cross validate warm-start test. The error bar is too small to show.

where N_r is the number of relevant videos in top n item candidates, N_t is the ground truth of relevant videos in the testing set, p_i is the rank of each relevant video in the recommender pool.

In our warm-start scenario, Figure 3.5 (a)–(c) show these three metrics of the four algorithms respectively when n equals to 3 with varying cluster number K from 15 to 150. The graphs illustrate that DPIS for topic number larger than 15 outperforms the other recommendation algorithms in terms of every quality measure. As we increase the topic number, the three measures have an overall trend of improving, stabilized around topic number equalling to 80. This can also be utilised as a way of determining the appropriate topic number for this dataset. From a user perspective, the averaged hit rate count is a more important factor for judging recommender



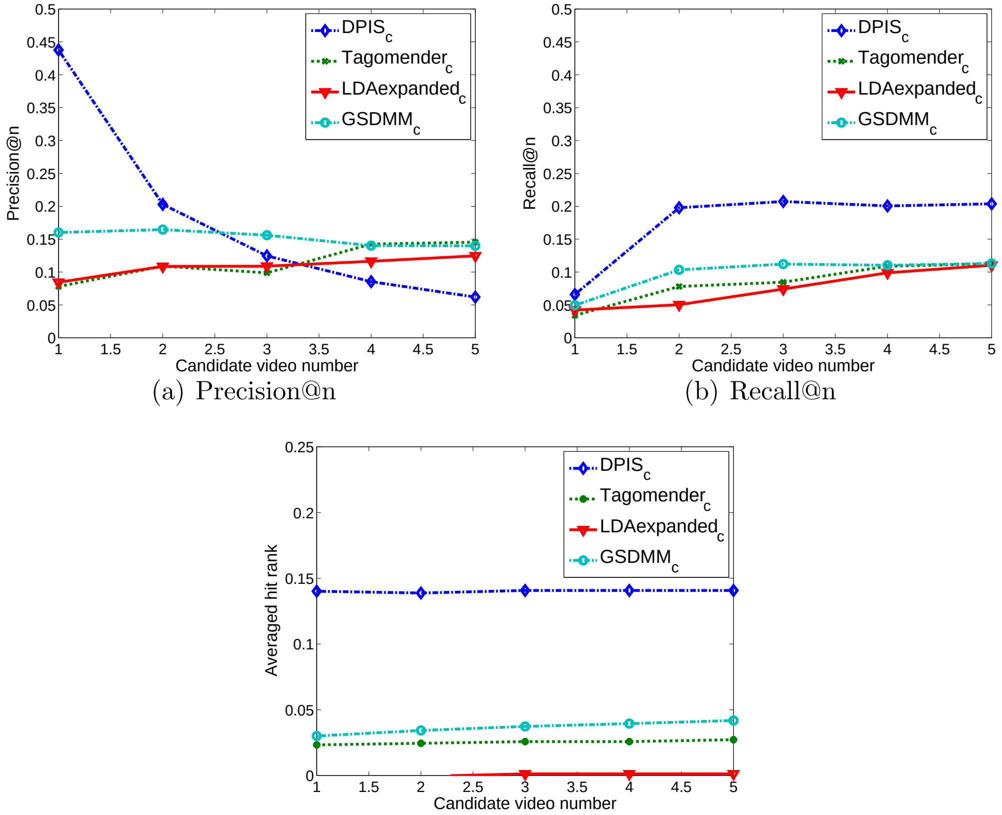
(a) Precision@n

(b) Recall@n

(c) Averaged hit rate

Figure 3.6: Vary candidates number n from 1 to 10 with topic number equal to 150 for warm-start test.

system, since it indicates whether the recommended items would fit users' interest at earlier order. Although DPIS resembles to AT model most, due to the different way of placing the topic probabilistic distribution, the single topic assumption for each sample estimates the sparse data better and hence generates better results comparing to AT which relies on the co-occurrence of tag words. What's more, since AT does not model the direct relation between user and sample, but only clusters user and tag topics separately, the hit rate does not change much with increasing number of topics. This is due to a lack of efficient ranking mechanism to differentiate the priority of preferences within similar user groups.



(a) Precision@n

(b) Recall@n

(c) Averaged hit rate

Figure 3.7: Vary candidates number n from 1 to 5 with topic number equal to 150 in cold-start scenario where new videos are added to the database.

Influence of recommendation candidate number

The influence of how many videos should be recommended is also worth discussing. Usually for the browser interface, $5 - 10$ videos are listed as a guessing of user interest. The values for all three metrics mentioned in the previous subsection are illustrated in Figure 3.6 (a)–(c). The increase of pool number will enhance the probability to recommend the favored video. Thus, recall rate and average hit-rank grow gradually and stabilize. DPIS has outstanding performance comparing to other methods on precision when the recommendation candidate number is small. The prediction precision decreases below the average performances of other methods when the candidate video number is larger than seven. Both the recall values and average hit rates for DPIS are more consistent comparing to other methods. All these phenomena indicate that DPIS can hit the correct preferences of users at an earlier stage of recommendation which makes it more suitable for short list of video recommendation.

3.6.3 Cold-start Scenario Evaluation

For the cold-start case, we first examine the case where new videos are added for recommendation. The precision, recall with recommended item size ranging from 1–5, and averaged hit rate are illustrated in Figure 3.7 (a)–(c). The overall trend is similar to the warm-start case, GSDMM does better comparing to LDA and Tagomender, which supports our choice of targeting the scarceness of information. However, its performance is limited due to the restriction of single topic assignment for each user, which is usually not the case. As a contrary, although DPIS assigns only one topic to each sample for the purpose of solving tag sparsity issue, each video can appear multiple times, which indicates that there is still probability that the video could be assigned to multiple topics based on the high-level semantic clustering of user preference. Hence, overall, the performance of DPIS is better than GSDMM, when both methods are targeting at the short-length of text features.

Considering the third type of cold-start recommendation mentioned in [43], since DPIS is independent of the user attribute feature, it is difficult to directly predict the preference of new user with no previous historical samples. So for the initial recommendation, we just use a way similar to random guess, that is to recommend the most popular videos in the past three days. As a horizontal test, with a gradual increase knowledge of the viewing history of user, we can use the posterior topic probability to gradually ensure our recommendation. The improvement of prediction precision when user samples increasing to 20 is presented in Figure 3.8 comparing with the maximum precision values of recommending the most popular one as well as AT and GSDMM model. It is clear that with accumulative gathering of viewing information, the recommender system can get more robust to new users. Since the recommendation of AT is based on the clustering of users, the precision is higher at an earlier stage. However, with growing samples, the user preference becomes more specific, DPIS, which bases its recommendation on user topics, outperforms.

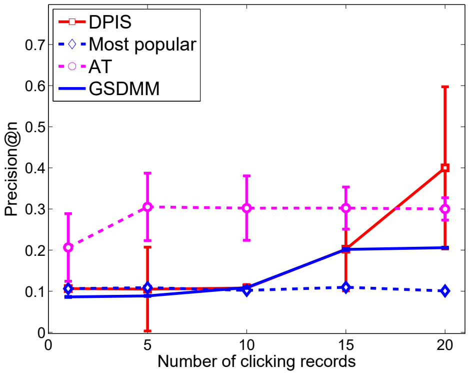


Figure 3.8: Prediction accuracy improvement with increasing clicking samples of new users

3.6.4 Topic Examination

In this section, we present some sample topics discovered using DPIS, and also provide a concrete example of topic discovery based recommendation for an active user to illustrate how the recommender system works. Consider the topic-modeling component part of the DPIS model alone, we compare the perplexity of LDA with and without data expansion, fast DLDA and GSDMM. Each of the models was initialized with the same hyper-parameters and topic number, if required, to impose a fair comparison. Since larger likelihood would imply a better chance of observing the true value, with a monotonically decreasing function applied on it, a lower perplexity would indicate a better model explanation of the data. Table 3.4 summarizes the perplexities using each of the methods. The degradation of using expanded words for LDA suggests that simple augmentation will not improve the co-occurrence pattern, but worsens the topic interpretation due to a destruction of the original data

Table 3.4: Perplexity of different methods on Tencent browser dataset

Method	LDA	LDA expand	fast DLDA	GSDMM	DPIS
Perplexity	1432	1627	1982	1074	**1035**

structure and an overlook of the latent relations between the word combinations. We can see that although supervised fast DLDA method outperforms DPIS in prediction accuracy, its performance on perplexity is not as competitive as DPIS. This might be benefited from the sample level modeling of topics, which makes DPIS more suitable for scarce information exploration.

Another advantage of DPIS is to provide an exploration of user latent preference space using the comprehensive topic representation learned from the historical data. Table 3.5 shows one example of three top matched topics for user and eight recommended videos provided by DPIS. The topics are represented by the four categories of tag words (translated into English). The last column indicates whether the user has already clicked the video or not. We can see from the topic tag word, that learned topics serve as a summary of what this user might be interested in. For example, this user has clicking samples of Korean soup dramas. The model detects his primary interest in love story, thus suggests several Korean dramas accordingly, some of which are indeed watched already. Also considering his interest in comedy, suspense and horror videos, the system recommends “The Conman” (comedy) and “The Missing” (horror), which keep the diversity of recommendation.

3.6.5 Computational Cost

Theoretically, the complexity for the generation part is $\mathcal{O}(KN_s\bar{V_s})$, where $\bar{V_s}$ is the average vocabulary size of V_s. To justify the scalability of DPIS, we run an experiment on synthetic dataset with training size ranging from $10,000$ - $100,000$ for 50 iterations. The topic number K stay fixed at 150 and the tag word length is truncated or padded to 10. The run-time of DPIS generative part is about 1.08 times

Table 3.5: Example user with top-2 highest weighted topics, and 8 suggested videos as predicted by DPIS. The last column indicates whether the recommendation has been viewed or not.

	user	clicked?
Topic 27	Drama Comedy Romance Korea micro-film ZhouXingXing YangXiaoyang ZhaoruZhen	
Topic 5	Romances horror suspense France Jean-Marc Barr	
Topic 15	Love Story Ethical Hong Kong TangNing Chen Zhiqing	
Recommend 1	How to Meet a Perfect Neighbor (Korean drama)	Yes
Recommend 2	Woman on the Beach (Korean drama)	No
Recommend 3	Warriors of the Rainbow: Seediq Bale (Taiwan film)	No
Recommend 4	Stained Glass (Korean drama)	No
Recommend 6	The Conman 2002 (HK film)	Yes
Recommend 7	John Rabe (film)	Yes
Recommend 8	The Missing (HK film)	Yes

of LDA. As shown in Figure 3.9, DPIS scales linearly with the number of training samples. A Spark version of the algorithm is also implemented and tested. With 180 machines running at the same time, the computational cost for 2000 iterations over four million samples daily can be controlled under three hours.

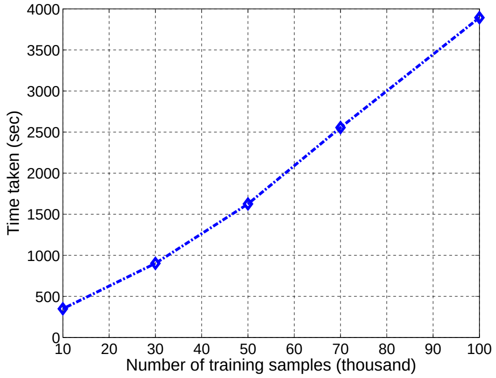


Figure 3.9: Linear scalability of DPIS for 50 iterations

3.7 Summary

With the exponential growth in online video usage, predicting the potential interest of user is a typical recommendation application. Large-scale video databases and user click logs have long-tailed and sparse features, which presents challenges to traditional collaborative filtering and content-based recommender systems. Targeting this two-aspect information scarcity, we present DPIS, a personalized recommendation algorithm for online video using a novel probabilistic probit topic model. Through concatenating viewing samples for maintaining tag word correlation, the probabilistic generating processes for sample-level topic representation of user preferences is described and the Gibbs sampling algorithm for inference and parameter learning is realised. Experiments on large-scale real-world datasets collected from Tencent QQ browser prove that the proposed model can improve the quality of recommendations based on the semantic topics of user preference in both warm-start and cold-start scenarios.

We expect this work could provide some inspirations to readers on how to link discrete count information with binary indicator information through closed form Bayesian networks. Preliminary results of this work appeared in the Proceedings of the 25th ACM International on Conference on *Information and Knowledge Management* [59].

As in this approach, we still treat user-video and video-text information as two separate matrices, it is reasonable to question how analyzing these three parts as a whole will influence the performance and whether we can utilize deep presentation for feature understanding. Hence, we turn to tensor, a natural extension of matrix, in the next chapter.

Chapter 4

Deep Bayesian Tensor Factorization for Computational Creativity based Video Recommendation

4.1 Introduction

Relational data based personalized recommendation plays an essential role in today's e-commerce operations. While those emerging web and video sites provide services of millions of TV shows, movies, music and news clips, they are also a main source of capturing browsing or operational data for a huge amount of users. As have been discussed in the previous chapter, there are two traditional and widely applied approaches to this kind of tasks, i.e., Content-Based Filtering (CBF) and Collaborative Filtering (CF). CBF compares new information with the historical profile to predict its relevance to certain users [5]. CF recommends items based on the common preferences of a user group, without using the item attributes. Although both of them have performed superiorly well on many systems, they have drawbacks facing the challenges aroused by the increasing availability of large-scale digitized data.

Taking the online video clicking log mentioned in Chapter 3 as an example, the database stores both user-item interaction and item-tag interactions. With respect to the Tencent QQ browser platform or other user generated content sites such as Youtube or Twitter, the challenge of data sparsity is often faced [30]. Due to the huge volume of users, videos clicked every day have the features of being extremely dynamic and long tailed. It is difficult to directly observe which category of tags are deterministic to a user's choices. In this case, CF tends to recommend popular items rather than those cold ones which might actually correspond to the taste of the target user [85]. On the other hand, purely item-based CBF may overlook the hidden and mixed patterns of user preferences.

To address these limitations, we propose a recommender system leveraging both user and item features. As it is reasonable to assume that the historical behavior of users is a sound source for preference estimation, the high-level semantic topics of video tags can be regarded as a comprehensive representation of user features. There are some work on incorporating user features (e.g. age, gender, geographic information) and their social relationships (e.g. community groups) into a multi-way data analysis [71] [108]. However, for video recommendation, users are more accurately clustered based on their overlapping of preferences, which can be represented by semantic relations among video tags. Integrating such multi-facet information together poses certain challenges. Moreover, since the tags are manually labeled, error, incompleteness and abundance exist. How to overcome these deficiencies and explore a better representation of user preference features is also an important concern of model construction. Regarding interaction information in a tensor format is hence a natural and sound approach. This type of multi-way tensor data [64] is prevalent and has diverse applications. For example in recommender systems, we could obtain the click records of each individual active user, and add multiple categories of tags to each video in the database. So the tensor can be constructed flexibly either in a real-valued way using the number of clicks, or a binary way using the action user takes on the video. Thus, tensor decomposition methods are becoming appealing for these diverse data types.

Traditional multi-way factor models [89] [34] suffer from the drawback of failing to capture coupled and nonlinear interactions between entities [96]. Also, they are not robust to datasets containing noisy and missing values. Through proper generative model, nonparametric Bayesian multi-way analysis algorithms (like [15] [96] [72]) are especially appealing, since they provide efficient ways to deal with distinct data types as well as data with missing values and noises. Meanwhile, deep networks prove great empirical success in various domains. They are capable of providing more compact nonlinear representations for feature learning. It would be interesting to adopt deep learning in one or more of the tensor modes and assess its effectiveness on tensor completion.

Motivated by the aforementioned consideration, this chapter presents a fully conjugate deep probabilistic approach for tensor decomposition, named Deep Canonical PARAFAC Factorization (DCPF). Based on the Canonical PARAFAC (CP) decomposition, the model is capable of clustering the three-way data along each direction simultaneously. To find a more compact representation in the latent space of each mode, a multi-layer factorization is imposed on the mode factor matrix to incorporate nonlinear mapping. Instead of relying on ad-hoc or cross-validating parameter selection, the rank of the core tensor and the factor number for each layer of the deep network can all be automatically determined. As a fully conjugate Bayesian model, efficient Gibbs sampling inference is facilitated, with an improving performance for tensor reconstruction and prediction accuracy.

Related to the diversity, novelty and serendipity perspectives of recommender system, we also consider incorporating computational creativity into recommendation. Computational creativity, a study on algorithms for computer to generate artifacts that humans perceive to be creative, is a newly emerging field in today's e-commerce operations. Colton et al. [18] provide a definition for computational creativity research as: *"The philosophy, science and engineering of computational systems exhibit impersonally creative behaviors by taking on particular responsibilities."* Based on this definition, attempts including automatic culinary recipe generating system [70] have been made through cognitive flavor assessment. For recommendation, diversity also has become an important aspect for evaluation [76], where content-based systems usually suffer from over-specialization, since only items similar to those rated by users will be more likely recommended. Computational creativity is also related to serendipity, as defined in [42], a user-oriented measurement balancing between surprise and accuracy. Serendipitous recommendations by definition are also novel. Consider a recommender system that simply recommends movies directed by the user's favorite director, comparing to recommend a movie of the same director that the user was not aware of, a movie by a new director catering to the user's taste, is more likely to be not only novel but also serendipitous.

This whole idea of calculating computational creativity for recommender system from the automated processing of mass collection of item tags requires us to address several problems. First, we face the issue of effective user profiling. A two-dimensional collaborative filtering approach cannot be directly employed to build a tag-based recommender system, since it cannot efficiently capture the multi-dimensional characteristic and hence, will result in poorer recommendation performance. Second, creativity of new ideas built upon domain knowledge is difficult to be measured quantitatively. Besides traditional performance measurements, e.g. precision, recall, F-1, new evaluation metrics are required. Past approaches use historical viewing frequency based distance measurement, but as user preference is a complicated topic combination, pure frequencies would hardly express their real interest.

Fortunately, comparing to traditional user-item based recommender system, meta-data such as tags, which are reusable and shareable, play significant roles in helping manage online resources. As an additional source for item recognition, tags help in revealing user and improving user profiles that can be used in recommendation [39]. Hence, in this chapter, we address the following challenges. Given the inordinate amount of candidate items and their corresponding textual tag information now accessible online, is it possible to automatically generate serendipitous tag combinations for users via machine learning? For example, given an online video recommender system, if a user has watched two videos with tag "Romance, Comedy" and "Thrilling, Horror", how would he react to a video tagged by "Comedy, Horror"? Will he find it creative or just as expected?

We assume that the historical behavior of users is a sound source for preference estimation and the high-level semantic topics of video tags can be regarded as a comprehensive representation of user features. There are some work on incorporating user features (e.g. age, gender, geographic information) and their social relationships (e.g. community groups) into a multi-way data analysis [71]. However, for video recommendation, users are more accurately clustered based on their overlapping of preferences, which can be represented by semantic relations among video tags. Integrating such multi-facet information, user profiles can be naturally modeled with higher-order data mining models. Tensor modeling is a well-known approach for representing latent relationships inherent in the multi-dimensional data.

There are typically three steps for tensor-based recommendation: (1) tensorial construction for multi-relational data; (2) tensor decomposition for latent feature (topic) representation; and (3) tensor reconstruction for interaction regeneration. Traditional multi-way factor models suffer from the drawback of failing to capture coupled and nonlinear interactions between entities [96]. Also, they are not robust to datasets containing noisy and missing values. Through proper generative model, nonparametric Bayesian multi-way analysis algorithms are especially appealing, s-ince they provide efficient ways to deal with distinct data types as well as data with missing values and noises. Moreover, since we are expected to recommend items not only considering the accuracy but also the creativity and serendipity, the posterior likelihood of Bayesian model can also be leveraged as a probabilistic ranking generating mechanism. Meanwhile, deep networks prove great empirical success in various domains [6]. They are capable of providing more compact nonlinear representations for feature learning. It would be interesting to adopt deep learning in one or more of the tensor modes and assess its effectiveness on tensor completion.

The aforementioned challenges are addressed through a framework using the proposed deep tensor for probabilistic recommendation. It breaks the task at hand into the following components: 1. a tensor construction stage of building user-item-tag correlation; 2. a tensor decomposition stage learning factors for each component mode; 3. a stage of tensor completion, which computes the creativity value of tag pairs; and 4. a recommender stage that ranks the candidate items according to both precision and creativity consideration. This approach is evaluated using a real world video recommender system, with large amount of users, videos and corresponding video description tags.

4.2 Related Work

4.2.1 Canonical PARAFAC (CP) Decomposition

The core of our proposed model is the Canonical PARAFAC (CP) decomposition [17]. CP, as a special case of Tucker decomposition, decomposes a tensor into a sum of rank-1 component [44], as illustrated in Figure 4.1. A K-mode tensor $X \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_K}$, where n_k denotes the dimension of each mode, can be expressed by:

$$X = \sum_{r=1}^{R} \lambda_r \cdot u_r^{(1)} \circ u_r^{(2)} \cdots \circ u_r^{(K)} \tag{4.1}$$

Here, we adopt the notations from [44]. The column vectors $\{u_r^k\}_{k=1}^{K} \in \mathbb{R}^{n_k \times 1}$

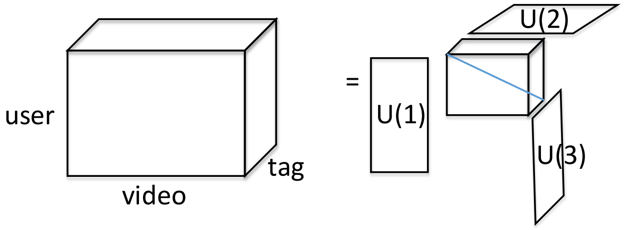


Figure 4.1: CP decomposition: A three mode user-video-tag relational dataset example.

denote the latent factors for each mode, combining which forms the factor matrices $U^{(k)}$. R is a positive integer indicating the rank of the core tensor. λ_r is the weight associated with the r^{th} rank component and $\circ$ denotes vector outer product. In an element-wise way, the tensor element $x_{\mathbf{i}}$ with subscript $\mathbf{i} = i_1, ..., i_K$ denoting the K-dimensional index of the i^{th} entry in the observed tensor can be concisely expressed as:

$$X = \sum_{r=1}^{R} \lambda_r \prod_{k=1}^{K} u_{i_k r}^{(k)} \tag{4.2}$$

The tensor can also be expressed in terms of the factor matrices $\{U^{(k)}\}_{k=1}^{K}$: $X = \{\lambda, U^{(1)}, U^{(2)}, \ldots, U^{(k)}\}$, where $U^{(k)} \in \mathbb{R}^{n_k \times R}$. In [72] and [95], each factor matrix is assumed drawn from a Gaussian prior. For inferring the rank of the core tensor, we will follow the multiplicative Gamma approach.

4.2.2 Multi-task Learning, Transfer Learning and Deep Learning

Learning computationally creative features can be linked with two other state-of-art machine learning methods, i.e. multi-task learning and transfer learning. Multi-task learning tries to learn multiple tasks simultaneously through uncovering the common latent features [67], while transfer learning is distinguished by removing the assumption that the training and test data are drawn from the same feature space and the same distribution. Since the essence of creative learning is to infer the unexplored feature space, both learning paradigms might be helpful for solving approach design.

Significant recent research on deep models has proved its effect on data representation. The form of the proposed multi-layer implementation is most related to [12] and [51]. The main idea is that an unsupervised deep model can be viewed as a hierarchy of factor-analysis [102], with the factor decomposition from lower layer serving as the input of deeper layer.

4.3 Deep CP Decomposition

4.3.1 Model Description

Let Y denote an incomplete K-order tensor. For different types of observation Y, a noise function f can be applied [97], depending on the data type being modeled, e.g. Gaussian for real valued data, or Bernoulli-logistic for binary valued. The goal is to infer the parameters of CP decomposition, $\boldsymbol{\lambda}$ and $\{U^{(k)}\}_{k=1}^{K}$, based on sparse observation Y. Assuming that the elements $y_i's$ of the observations are i.i.d, for continuous observations with Gaussian noise, the joint likelihood distribution of Y can be written as:

$$\mathrm{p}(Y|X) = \prod_{k=1}^{K} \prod_{i_k} \mathcal{N}(\mathbf{y_i}|\mathbf{x_i}, \tau_o^{-1}) \qquad (4.3)$$

where τ_o is the precision of the noise, and $\mathbf{i} = i_1, ...i_K$.

How to reduce the size of tensor rank so as to make the rich-feature based user representation scalable is one primary concern during model construction. Our low-rank construction is adopted from [7] and [72] using the multiplicative gamma process (MGP). Putting the prior on the super-diagonal elements of the core tensor $\boldsymbol{\lambda}$, the number of diagonal elements will increasingly shrink to zero. When it stabilizes, the number of elements remained can be inferred as the appropriate rank for dimensionality reduction.

4.3.2 Multi-layer Sparse Factorization

To enhance the feature representation, factor matrix $U^{(k)}$ for mode k can be further constructed through an unsupervised deep model in terms of a hierarchy of factor analysis. Assuming that the hierarchical tensor factorization model is performed for L layers, the original data Y is represented in terms of $N_k \times R$ tensor factor matrix $U^{(k)}$ as equation (4.1). In addition to assess the classification performance based on the factor loading and scores of the decomposition, it is of interest to examine the physical meaning of the associated factor elements. The input for each layer is the previous layer's factor loading matrix. The discarding of residue between the layers acts as noise filtering.

In the multi-layer stage, each factor matrix is further represented by a lower rank component $W^{(k)} \in \mathbb{R}_{N_k \times M}$ and $D^{(k)} \in \mathbb{R}_{M \times R}$, where M indicates the number of factors for this layer. The matrix E captures the idiosyncratic noise. $l = 1, 2, \ldots, L$ specifies how deep the network wishes to go to. The output of the $l - 1$ layer decomposition can be used as the input to the $l^t h$ layer (as shown in Figure 4.2).

The inference of the factor number for each layer is realized through a Beta-Bernoulli process [87] as $W^{(k)} = B^{(k)} \odot V^{(k)}$. In practice, the number M is initialized large, thus the element $b_{nm}^{(k)} \in \{0, 1\}$ of $B^{(k)}$ can indicate whether the m^{th} factor has been used or not.

To go to the next layer, we denote $\hat{M}$ as the number of factors that has nonzero indicator $B^{(k)}$ for at least one sample, and use these factor loadings as the entry of the next layer. Model fitting at the deeper layers are similar to the first layer. With the gradual deduction of factor numbers, the computational complexity decreases with layer increasing.

4.3.3 Probabilistic Hierarchical Tensor Model

Combining the two steps discussed in previous section, we now propose a hierarchical generative framework for a three-way tensor data whereby the aforementioned tensor construction and deep model fitting can be performed. The DCPF model is formed as follows (the graphic model in Figure 4.2):

$$\mathbf{y_i} \sim \mathcal{N}(\mathbf{x_i}, \tau_o^{-1})$$

$$x_\mathbf{i} = \sum_{r=1}^{R} \lambda_r \prod_{k=1}^{K} u_{i_k r}^{k}$$

$$\lambda_r \sim \mathcal{N}(0, \tau_r^{-1})$$

$$\tau_r = \prod_{t=1}^{r} \delta_t, \quad \delta_t \sim \mathcal{G}amma(a_r, 1) \quad a_r > 1 \tag{4.4}$$

$$u_r^k \sim \mathcal{N}(\mu^{(k)}, \sigma^{(k)})$$

$$U_l^{(k)} = B_l^{(k)} \odot V_l^{(k)} D_l^{(k)} + \mathbf{E}_l^{(k)}$$

$$b_{n_k m} \sim \mathcal{B}ernoulli(\pi_m), \quad \pi_k \sim \mathcal{B}eta(1/M, b)$$

$$v_{n_k m} \sim \mathcal{N}(0, \tau_{v_{n_k m}}^{-1}), \quad \tau_{v_{n_k m}} \sim \mathcal{G}amma(c_0, d_0)$$

$$d_{mr} \sim \mathcal{N}(0, \tau_{d_{mr}}^{-1}), \quad \tau_{d_{mr}} \sim \mathcal{G}amma(e_0, f_0)$$

$$\mathbf{E}^{(k)} \sim \mathcal{N}(0, \tau_\epsilon^{-1}), \quad \tau_\epsilon \sim \mathcal{G}amma(g, h)$$

The multiplicative gamma process prior is described in equation (4.4), and is placed on the precision of the Gaussian distribution for λ_r. To infer the appropriate rank R, we start at a reasonably large truncation level. When the value of component λ_r drops below a threshold, e.g. 0.001, it is discarded. With a probability $p(t) = exp(\beta_0 + \beta_1 t)$ at the t^{th} iteration (in practice β_0 and β_1 are chosen as 1 and 0.2

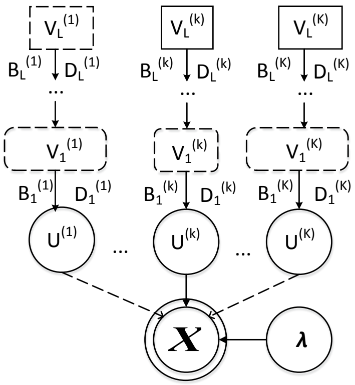


Figure 4.2: Graphic model: The observed W is decomposed into a diagonal core tensor $\boldsymbol{\lambda}$ and K-mode of matrices. Each factor matrix $U^(k)$ is further constructed through a L layer deep network .

respectively, so that adaptation occurs around every 10 iterations at the beginning of the Markov chain but decreases in frequency exponentially fast [7]), a sequence of uniform random numbers are generated. If the t^{th} value is smaller than $p(t)$, we discard the redundant entities. In this way, no parameter tuning is required for low-rank inference.

Considering the model, a straightforward Gibbs sampler for posterior computation can be formulated. Since it is fully conjugate, the sampler cycling steps are presented in Algorithm 4.1. Note that since for each layer of the deep network, the basic model is the same, the layer superscripts are omitted for clarity. The Beta-Bernoulli construction is used to infer the number of factors for layer-wise decomposition. In practice, the factor number M is truncated for inference of the subset that are actually needed. One could alternatively employ the Indian buffet process (IBP) in [31]. But a truncated version is proved efficient for satisfactory performance [12]. All the parameters with "ˆ" in the updating steps are the posterior expectation and precision for $\boldsymbol{\lambda}$ and $U^{(k)}$, which will be further elaborated in the following inference sub-section.

4.3.4 Inference

In this sub-section, we present in detail the Gibbs sampling derivations for inferring the latent variables in the multilayered tensor model. Since exact inference is intractable, we implement the posterior computation using Markov Chain Monte Carlo (MCMC). As shown by the model construction, the model is locally conjugate. For the automatic shrinkage of the tensor rank λ_r part, we adopt the method from [72]. The joint likelihood is presented as:

$$\begin{aligned}
&P(Y, X, \boldsymbol{\lambda}, U, B, V, D)\\
=&\prod_{i} \mathcal{N}(y_i|x_i, \tau_o^{-1})\\
\times&\prod_{r=1}^{R} \mathcal{N}(\lambda_r|0, \tau_r^{-1})\mathcal{G}a(\delta_r|a_r, 1)\prod_{k} \mathcal{N}(\mathbf{u}_r^{(k)}|\mu_r^{(k)}, \tau_\epsilon^{-1})\\
\times&\prod_{i_k}\prod_{m} \mathcal{B}ernoulli(b_{i_k m}^{(k)}|\pi_m^{(k)})Beta(\pi_m^{(k)}|a_0, b_0)\\
\times&\prod_{i_k} \mathcal{N}(\mathbf{v}_{i_k}^{(k)}|(B^{(k)} \odot V^{(k)})D^{(k)}, \tau_v^{-1})\mathcal{G}a(\tau_v|c_0, d_0)\\
\times&\prod_{m}\prod_{r} \mathcal{N}(d_{mr}|0, \tau_d^{-1})\mathcal{G}a(\tau_d|e_0, f_0)
\end{aligned} \tag{4.5}$$

In the proposed model, all conditional distributions are analytic. The choices for prior hyper-parameters are relatively standard in Bayesian analysis, hence no particular tuning is required. Updating equations for the latent parameters are provided in detail as follows.

- Update $\boldsymbol{\lambda}$

As illustrated in the sampling steps of Algorithm 4.1, the posterior mean and precision for $\boldsymbol{\lambda}$ are:

Algorithm 4.1 Gibbs sampler steps

```
Begin
Initialize $U^{(k)}$, $B^{(k)}$, $V^{(k)}$, $D^{(k)}$
for iteration $i$ do
    for each diagonal element $r$ of the core tensor $\Lambda$, which is independently drawn from a normal distribution do
        1. Sample its independent conditionally conjugate posteriors from $\lambda_r \sim \mathcal{N}(\hat{\mu_r}, \hat{\tau_r}^{-1})$
        2. Update $\delta_r \sim Ga(a_2 + \frac{1}{2}(R - r + 1), b_2 + \frac{1}{2}\sum_r \tau_r^2 \delta_r)$
    end for
    for each mode factor matrix $U^{(k)}$ do
        1. Update binary indicator matrix $B^{(k)}$, factor loading matrix $V^{(k)}$ and factor score matrix $D^{(k)}$
        2. Sample column vector conditionally conjugate posteriors from $U_{i_k}^{(k)} \sim \mathcal{N}(\hat{\mu}_{i_k}^{(k)}, \hat{\Sigma}_{i_k}^{(k)-1})$
        3. Update hyper-parameters
    end for
end for
```

$$\begin{aligned} \hat{\tau_r} &= \tau_r + \tau_\epsilon \sum_i (\prod_{k=1}^{K} U_{i_k r}^{(k)}) \\ \hat{\mu_r} &= \hat{\tau_r}^{-1} \tau_\epsilon \sum_i \prod_{k=1}^{K} U_{i_k r}^{(k)} (y_i - \sum_{r \neq r'} \lambda_{r'} \prod_{k=1}^{K} U_{i_k r'}) \end{aligned} \tag{4.6}$$

• Update $U^{(k)}$

For $1 \leqslant r \leqslant R$, $1 \leqslant k \leqslant K$, at the (r, k) tuple, all the other entities are regarded as non-variables, so x_i can be rewritten as:

$$x_i = \underbrace{(\lambda_r \prod_{k' \neq k, k'=1}^{K} u_{i_{k'} r}^{(k')})}_{} u_{i_k r}^{(k)} + \underbrace{\sum_{r' \neq r} \lambda_{r'} \prod_{k=1}^{K} u_{i_k r'}^{(k)}}_{} \tag{4.7}$$

Let the first parentheses part equals to $p_{i_k r}^{(k)}$ and the second part be $q_{i_k r}^{(k)}$. With Gaussian noise precision τ_ϵ, the prior of $u^{(k)}$ is $\mathcal{N}\left(\mu^{(k)}, \tau_\epsilon^{-1}\right)$, where $\mu^{(k)}$ equals to $(B^{(k)} \odot V^{(k)}) D^{(k)}$. Thus the conjugate posterior can be inferred as:

$$\mathbf{u}_{i_k}^{(k)} \sim \mathcal{N}\left(\hat{\mu}_{i_k}^{(k)}, \hat{\Sigma}_{i_k}\right) \tag{4.8}$$

with the posterior expectation and covariance as

$$\begin{aligned}\hat{\Sigma}_{i_k} &= (\tau_o \sum_{i_k} \mathbf{p}_{i_k}^{(k)2} + \tau_\epsilon^{-1}) \\ \hat{\mu}_{i_k}^{(k)} &= \hat{\Sigma}_{i_k}^{-1} \left(\tau_\epsilon \mu_{i_k}^{(k)} + \tau_o \sum_i (y_i - \mathbf{q}_{i_k}^{(k)}) \mathbf{p}_{i_k}^{(k)} \right)\end{aligned} \tag{4.9}$$

• Update $B_l^{(k)}$ and $V_l^{(k)}$

For each entity $b_{i_km}^{(k)}$ of $B_l^{(k)}$, we have

$$p(b_{i_km}^{(k)} = 1|-) = \widetilde{\pi_{i_km}^{(k)}} \tag{4.10}$$

where $\frac{\widetilde{\pi_{i_km}^{(k)}}}{1-\widetilde{\pi_{i_km}^{(k)}}} =$

$$\frac{\pi_{i_km}^{(k)}}{1-\pi_{i_km}^{(k)}} exp[-\frac{\tau_\epsilon}{2}(v_{i_km}^{(k)}{}^2 \mathbf{d}_m^{(k)} \mathbf{d}_m^{(k)^T} - 2v_{i_km}^{(k)} U_{-m}^{(k)} \mathbf{d}_m^{(k)^T})] \tag{4.11}$$

$U_{-m}^{(k)}$ here equals to $\mathbf{u}^{(k)} - \sum_m (b_{i_km}^{(k)} \odot V^{(k)}) D^{(k)}$ and $b_{i_km}^{(k)}$ is the most recent sample [12].

Taking the advantage of conjugate property, the posterior mean and covariance for the factor loading element $v_{i_km}^{(k)}$ can be derived as:

$$\Sigma_v = 1 \oslash (\tau_v + \tau_\epsilon \mathbf{d}_m^{(k)} \mathbf{d}_m^{(k)^T} b_{i_km}^{(k)}) \tag{4.12}$$

$$\mu_v = \tau_\epsilon b_{i_km}^{(k)} \Sigma_v \odot (U_{-m}^{(k)} \mathbf{d}^{(k)} + \mathbf{d}_m^{(k)} \mathbf{d}_m^{(k)^T} v^{(k)}) \tag{4.13}$$

$\odot$ and $\oslash$ are the element-wise product and division operator. For multi-layer implementation, after sampling $\mathbf{v}$, it is used as the input to the next layer. Thus, there is a residue filtering between each layer. The rest of the Gibbs draws are continued as follows.

- Update $D^{(k)}$

Similar to $V^{(k)}$, the Gaussian conjugate posterior parameters of $D^{(k)}$ can be expressed as:

$$\Sigma_d = 1 \oslash (\tau_d + \tau_\epsilon \sum_{i_k=1}^{n_k} b_{i_km}^{(k)} {v_{i_km}^{(k)}}^2) \tag{4.14}$$

$$\mu_v = \Sigma_v \odot (\sum_{i_k=1}^{n_k} b_{i_km}^{(k)} \tau_\epsilon (U_{-m}^{(k)} v_{i_km}^{(k)} + d_{i_km}^{(k)} {v_{i_km}^{(k)}}^2)) \tag{4.15}$$

- Update $\pi_m^{(k)}$

$$\pi_m^{(k)} \sim Beta(\hat{a}, \hat{b}) \tag{4.16}$$

where $\hat{a} = a_0 + \sum_{i_k=1}^{n_k} b_{i_km}^{(k)}$ and $\hat{b} = b_0 + n_k - \sum_{i_k=1}^{n_k} b_{i_km}^{(k)}$.

4.4 Measure of Computational Creativity

In dynamic situations, surprise is an important indicator of the belief changing significance [4]. It can be also referred as a quantization for creativity measure. According to Bayesian theorem, for an observer with prior distribution $P(X)$, the collection of data D leads to a re-evaluation of beliefs $P(X|D)$, the posterior distribution. Surprise is to define the distance between the prior and posterior distributions, specifically as:

$$S(D, X) = d[P(X), P(X|D)] \tag{4.17}$$

where d is a distance measure function. For example, if we use the relative entropy or Kullback-Liebler [49] divergence, which is a common practice for many well-know dissimilarity measures, the surprise S can be calculated as:

$$S(D, X) = logP(D) - \int_X P(X) logP(D|X) \, \mathrm{d}X \tag{4.18}$$

Since we can regard the novel tag generation process as a tensor completion problem, based on the posterior distribution of factor matrix $U^{(1)}$, the posterior distribution for co-occurring tags t_1, t_2 for user u can be written as:

$$p(y_{i_1=\{t_1,t_2\},i_2=u,i_3}) = \prod_{i_3=1}^{n_3} \prod_{i_1} \mathcal{N}(y_{\mathbf{i}} | \hat{x}_{\mathbf{i}}, \hat{\tau}_e^{-1}) \tag{4.19}$$

where $\hat{x}_{\mathbf{i}} = \hat{p}^{(k)}_{i_k r} U^{(k)}_{i_k r} + \hat{q}^{(k)}_{i_k r}$, $\hat{p}_{i_k r}$ and $\hat{q}^{(k)}_{i_k r}$ are the posterior values calculated from the conditional posterior of $U^{(k)}_{i_k r}$.

4.5 Probabilistic Ranking with Bayesian Surprise

Most existing tensor-based recommendation approaches rank the recommendations based on the values of reconstructed tensor. For example, Zheng et al. [108] utilize the linear add-up of the i^{th} row of each mode factor matrix. The higher weight it is, the more relevant user i is related to the topic. Since the goal of our task is not only providing accurate but also creative recommendations, instead of solely using the final reconstructed tensor from the model, we propose to rank the result of tensor modeling incorporating the Bayesian surprise value for generating the top-N list of candidate items to the users.

Since the generated factor matrices discover the latent groups of associations among tags, users and videos, we can utilize them for recommendation [108]. For example, the i^{th} row of user factor matrix $U^{(2)}$ provides an additive combination of cluster components for user i. The higher weight it is, the more relevant user i is related to the topic, and similarly for the j^{th} row of video factor matrix $U^{(3)}$. Thus, groups can be recommended according to the linear add-up of their corresponding factor weights. The score matrix S for user-video pairs can be defined as:

$$S = \sum_{r=1}^{R} \lambda_r u_r^{(2)} u_r^{(3)T} \tag{4.20}$$

where $S \in \mathbb{R}_{N_u \times N_v}$, N_u, N_v are the distinct number of active users and videos. For top-N recommendation, we pick N videos with highest scores for each user. Similarly, score matrices can also be constructed for video-tag and user-tag pairs, and can be leveraged for tag recommendation and completion.

Using the reconstructed tensor $\hat{\mathcal{Y}}$, for each user u, two candidate lists can be created: (1) a list of items that the user might be interested in based on the posterior add-up values of user and item factor matrices; and (2) a list of tag preference pairs based on the normalized maximum values of Bayesian surprise metric. Assume the number of candidate item is n, and the number of tag pairs is J. J can be adjusted depending on how much compromise on creativity the system wishes to make. The score for ranking user-item pair is calculated as follows:

$$\mathcal{S}core_{u,v} = w_{u,v} \sum_{i=1}^{N_j \in v} \frac{S_{u,i}}{\sum_j^J S_{u,j}} \tag{4.21}$$

where $\mathbf{w} = \sum_{r=1}^{R} \lambda_r \mathbf{u}_r^{(1)} \mathbf{u}_r^{(2)T}$ is the weight matrix obtained from reconstructed tensor, N_j is the number of tag pairs belonging to video v, in the top-J surprising list to user u. $\mathbf{S}$ is the surprise degree matrix of users to tag pairs. Each row vector of the final score matrix is then ranked in descending order, indicating the top-N weighted items as recommendation candidates.

4.6 Experiments

We perform experiments on both synthetic toy data and large-scale real-word dataset to verify the performance of the DCPF model on expressing high-level semantic features of user behavior, and the effect of deep structure exploration through multilayer tensor construction.

Table 4.1: Synthetic data MSE comparison

	3-D data (R=8)	4-D data (R=10)
Bayesian CP	0.2431 ± 0.0247	0.0922 ± 0.0207
DCPF	0.2502 ± 0.0055	0.0459 ± 0.0014
2-layer DCPF	0.2490 ± 0.0006	0.0412 ± 0.0011

4.6.1 Toy Example

The first example we considered is a toy problem, in which 3-D and 4-D cases are tested to verify the tensor completion performance of DCPF. The 3-D synthetic data is of size $15 \times 14 \times 13$ with 50% non-zero values. The 4-D synthetic data is of size $20 \times 20 \times 20 \times 20$ with 1000 non-zero values. Table 4.1 compares the results using a baseline method Bayesian CP (BCP), i.e. a fully Bayesian version of the standard probabilistic CP decomposition [95]. The inferred rank using our method is 8 and 10 for the two synthetic datasets. Since BCP has to specify the rank, it was run with ranks ranging from $3 - 10$, and the rank that generates smallest mean squared error (MSE) is chosen for comparison.

We also construct a three-way $100 \times 100 \times 100$ tensor, with sparseness control (missing percentage) of $50\% - 90\%$. We compare the reconstruction errors (MSE) in Table 4.2. Similarly, Bayesian CP was run with rank ranging from $10 - 90$ using the best results. In all the cases, both one layer and 2-layer DCPF provide competitive performances comparing to the state-of-art BCP. From varying the percentage of missing values, we can infer that a multi-layer filtering of the factor matrix will prevent the degrading of the reconstruction performance especially when the data has higher sparseness percentage.

4.6.2 On Real Valued Large-scale Tensor

We use records in three consecutive weeks at the same time slot from Tencent QQ browser. Each record consists of video id, user id, and four categories of word tags:

Table 4.2: Reconstruction error comparison of different data sparse percentages (lower the better)

	90%	80%	70%	60%	50%
Bayesian CP	0.4137	0.4123	0.4120	0.4093	0.3993
DCPF	0.4104	0.4100	0.3989	0.3959	0.3957
2-layer DCPF	0.3951	0.3865	0.3811	0.3697	0.3542

(1) type (e.g. action, mystery); (2) region (e.g. US, main land China); (3) director and (4) actor, based on which we construct a three-way tag× user × video tensor. There are $4,071,811$ samples in total with $137,465$ unique users, and $9,393$ unique videos they clicked. We focus on warm-start test for current application, which requires both active users and candidate videos occurred before. So the training and testing subsets are generated as follows.

We select a subset of users who have over 10 historical records, which generates a sample collection of $335,783$ records with $42,652$ distinct word tags. For each category of tags, the vocabulary size is 2,941, 103, 7762 and 31,906 respectively. This is also inferred to as in-matrix prediction [93]. A 5-fold cross-validation is used for evaluation. To guarantee that all the users and items have already appeared in the training sets, for records that have been viewed by each user for over five times, we distributed at least one of them into each fold. For records that have appeared less than five times ($246,642$ in total), we always put them into the training fold. The dimensionality of the testing tensor is $1421 \times 4013 \times 231$.

The Gibbs sampling algorithm in sub-section 4.3.4 is implemented for the proposed DCPF model. In our experiments, 1000 iterations for burn-in were performed, and every 50 samples were collected for another 1500 iterations.

4.6.3 Factor Examination

To visualize how the multi-layer implementation of DCPF actually influences the feature representation, we present four sample factors of tags and top factor of videos discovered using both single layer and 2-layer DCPF in this sub-section. We examine $U^{(1)}$ and $U^{(3)}$, which represents the latent factors of tags and videos respectively. The top weighted factors from each factor matrix are selected with eight highest score items each (tag or video), for topic inference (Figure 4.3 and Figure 4.4).

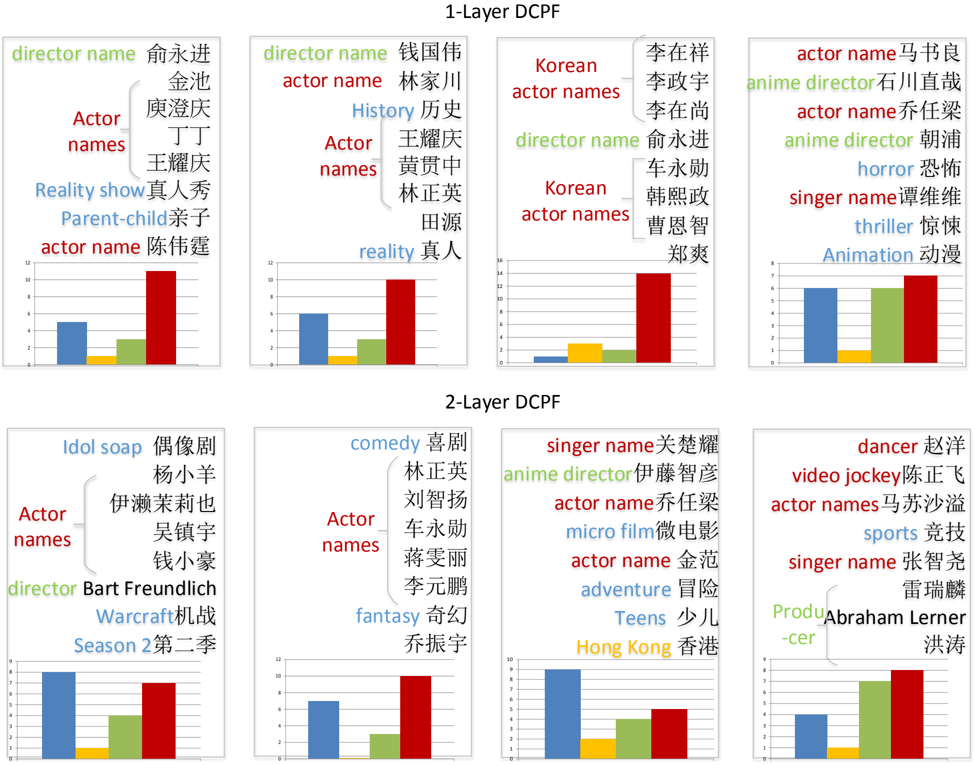
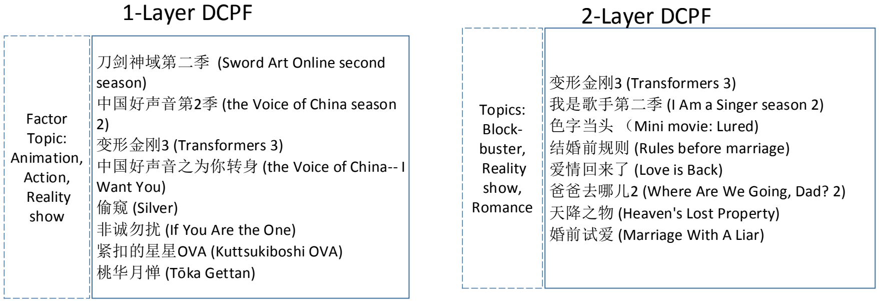


Figure 4.3: Four top weighted factors learned from single layer and 2-layer DCPF. Each block shows two aggregated statistics of the factor: the eight most representative tags and a histogram of the distribution of 20 top weighted tags from 4 categories. The Columns in each of the histogram correspond to type, region, director and actor from left to right.

As shown in Figure 4.3, semantic topics using single layer DCPF are primarily consisted of tags from the actor category. This phenomena illustrates that on one hand, certain types of users indeed choose their watching list according to favor of particular actors, on the other hand, this could be also due to the high occurrences of actors and reduplicative annotation. When going deeper, a 2-layer factorization filters out noises and abundances. Thus, factor weights for the other three categories of tags are better explored. Although the group of tags for each factor seems more irregular on the surface, the user preferences are actually better represented, since they are naturally and comprehensively mixed. Figure 4.4 depicts the top factor inferred from 1-layer and 2-layer using factor matrix alone video direction. Single layer DCPF tends to cluster videos in the same form, even same series together (e.g. the Voice of China), but the topics covered are more general. On the other hand, with the deep filtering of layers, DCPF are more diverse on the video form (mini film, regular film, tv shows etc.), but more focused in topics. As shown in Figure 4.4 (b), five out of eight videos are in the romance genre, which illustrates the benefit of deep structure. The improved prediction accuracy presented in previous sub-section also justifies this conclusion.



1-Layer DCPF

Factor Topic: Animation, Action, Reality show

刀剑神域第二季 (Sword Art Online second season)
中国好声音第2季 (the Voice of China season 2)
变形金刚3 (Transformers 3)
中国好声音之为你转身 (the Voice of China-- I Want You)
偷窥 (Silver)
非诚勿扰 (If You Are the One)
紧扣的星星OVA (Kuttsukiboshi OVA)
桃华月惮 (Tōka Gettan)

2-Layer DCPF

Topics: Block-buster, Reality show, Romance

变形金刚3 (Transformers 3)
我是歌手第二季 (I Am a Singer season 2)
色字当头 （Mini movie: Lured)
结婚前规则 (Rules before marriage)
爱情回来了 (Love is Back)
爸爸去哪儿2 (Where Are We Going, Dad? 2)
天降之物 (Heaven's Lost Property)
婚前试爱 (Marriage With A Liar)

(a) Top factor inferred from 1-layer DCPF (b) Top factor inferred from 2-layer DCPF

Figure 4.4: Factor learned from single layer and 2-layer DCPF. Each block consists a combination of eight videos with highest weights that jointly expressed the factor topic.

4.6.4 Tag Completion Evaluation

Recall that our goal is to automatically generate serendipitous tag combinations for users, it is important to analyze the tag completion performance part of the proposed method first. If we regard each type of tags as semantic words from different domains, this problem can also be regarded as multi-task clustering. We would like to evaluate how well the system ranks the correct items for each user based on single category and multiple categories of tags. Two metrics are employed for this: (1) the Mean Reciprocal Rank (MRR) and (2) Precision at 1 (P@1). MRR computes the inverse rank of the correct item and averages the score across the whole data. P@1 computes the percentage of times the ground truth item is ranked as the top one.

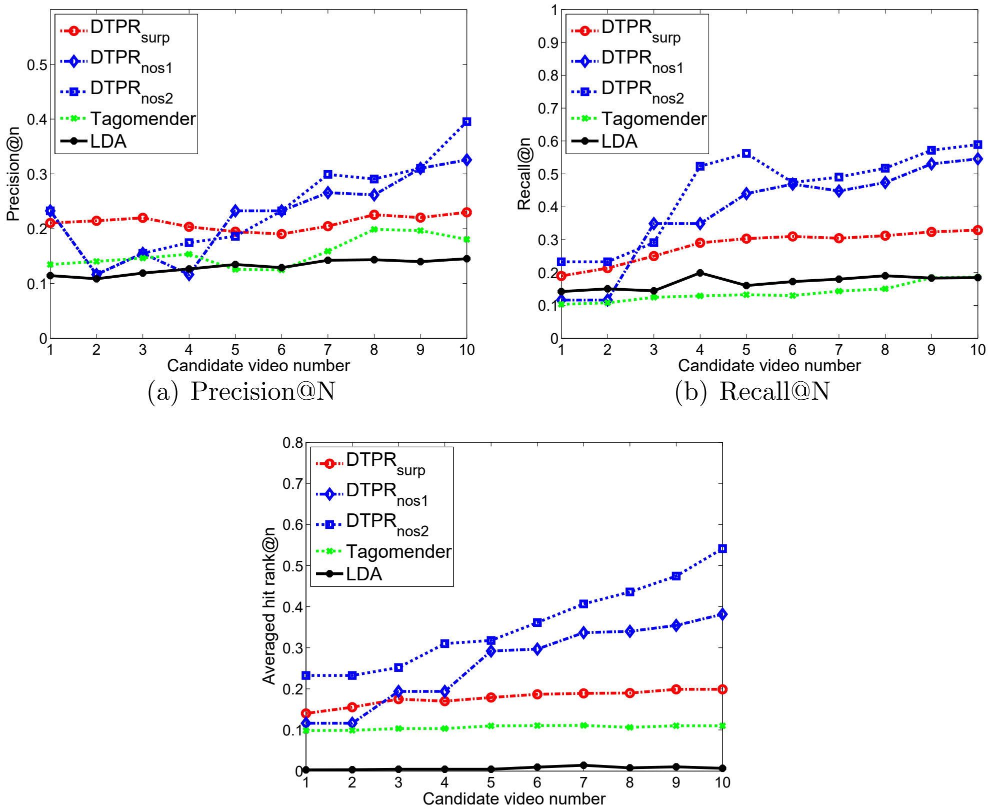
From Table 4.3, although having high frequency of occurrence, tags from category "Region" have the lowest MRR and P@1 due to its low differentiable vocabulary. For single layer DTPR, comparing to use purely actor tags, combining information from all these four categories increases the MRR by 1.7% (from 0.3662 to 0.3726). The 2-layer DTPR, in comparison, increases the MRR from 0.3744 to 0.4110, which is a 9% relative increase. Comparing to single layer DTPR, the increase in P@1 is also more obvious (36%) comparing to 1-layer implementation. This encouraging results indicate that a deep decomposition utilizing information from multiple aspects for factor matrix can better capture the semantic representation of user behavior.

Table 4.3: Results for 1-layer and 2-layer DTPR on different categories of tags.

	1-layer DTPR		2-layer DTPR	
	MRR	P@1	MRR	P@1
I	0.1867	0.0093	0.1974	0.0099
II	0.0009	0.0013	0.0009	0.0012
III	0.2900	0.2500	0.2990	0.2541
IV	0.3662	0.2233	0.3744	0.2350
ALL	0.3726	0.2249	0.4110	0.3197

4.6.5 Bayesian Surprise Assessment

For a subset of 3-way tensor, since Bayesian surprise is user dependent, the surprise value for each pair-wise combination of tags is presented in Figure 4.6. To make the figure clear, only the first $4,000$ pairs from a total $22,155$ are shown. A higher value



(a) Precision@N

(b) Recall@N

(c) Averaged hit rate

Figure 4.5: Video recommendation performance comparison for single layer DTPR incorporated with Bayesian surprise (DTPR$_{surp}$), 1-layer (DTPR$_{nos1}$) and 2-layer (DTPR$_{nos2}$) DTPR without Bayesian surprise, Latent Dirichlet Allocation [10], and Tag matching method [82], with varying candidates number n ranging from 1-10.

indicates a more diverse posterior belief. In other words, if the semantic themes from the pair of tags co-occur in the same video, a higher Bayesian surprise would indicate that the user regards it as more creative. As indicated from the figure, the response of users to these $4,000$ tag pairs can generally be grouped into five folds. Within each fold, the surprise degree of each user varies according to the past video preference. The clustering of tag pairs also illustrates that the user preference has its probabilistic topic features. Hence, comparing to traditional serendipity measurements, which are based on fixed distance of observed records, the posterior likelihoods are more suitable for creativity evaluation, as they incorporate the uncertainty and the cluster characteristics of user preference.

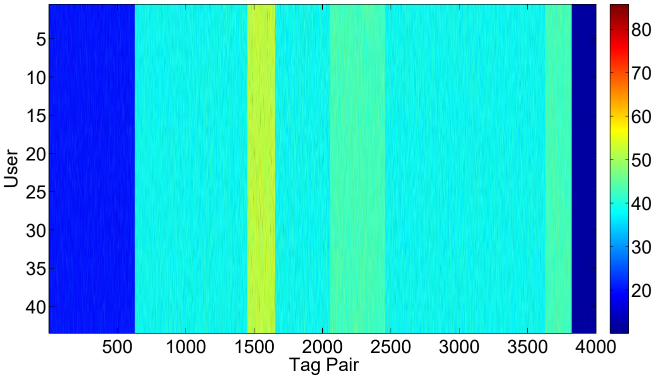


Figure 4.6: The surprise value for each user in the subset if the two tags occur together. X-axis indicates 4000 distinct tag combinations. Y-axis marks the 43 users

We also compare the recommendation performances with and without incorporating Bayesian surprise, using precision, recall and averaged hit rate. The pair number J is fixed at 5. As shown in Figure 4.5, we compare the proposed model in single layer and 2-layer version with and without incorporating Bayesian surprise for item ranking. The increase of pool number will enhance the probability to recommend the favored video. Thus, the three metrics have the trend of gradually growing. Although single layer and 2-layer DTPR have similar precision for predicting, a multi-layer implementation allows an obvious higher hit rate, which indicates that it can pick the correct choice of users at an earlier stage of recommendation. Through incorporating Bayesian surprise, the performance does degrade due to the additional uncertainty it introduces to the item ranking. However, it still outperforms both LDA and Tagommender [82] method. Considering the surprise it brings to the users, which will in return enhance user experience, this kind of recommendation has its merits in practice.

4.6.6 Computational Cost

To assess the scalability of DCPF, we again use the synthetic dataset generated in sub-section 4.6.1 with tensor size 100 × 100 × 100 for 50 iterations. The sparseness percentages ranges from $50\% - 90\%$, specifically, $100,000 - 500,000$ entries are observed. The core tensor rank stays fixed at 50. As shown in Figure 4.7, DCPF scales almost linearly with the number of training samples.

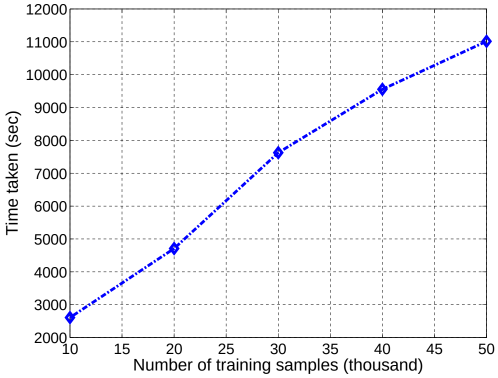


Figure 4.7: Linear scalability of DCPF for 50 iterations

4.7 Discussion

Although comparing to the DPIS methods proposed in the previous chapter, DCPF can provide higher performance in precision and recall, its prediction coverage rate is lower than DPIS. This is attributed to the imputation nature of tensor reconstruction, as the original existent links between user and item would still have high weights. However this also limits the diversity of the recommendation list. That is why we want to introduce the measure of computational creativity into the ranking process.

4.8 Summary

Given the increasing growth in large-scale multi-relational data, we study the computational creativity problem in video recommendation domain. To effectively learn user-item-tag correlations, we utilize deep Bayesian tensor model, which provides an effective way for joint analysis of user and video features. Through a scalable framework for tensor decomposition and completion, and through introducing Bayesian surprise into probabilistic ranking, we are able to recommend personalized items taking creativity as a consideration. Our model can perform fully conjugate Bayesian inference via Gibbs sampling inference. The quantization index, Bayesian surprise, for computational creativity is assessed.

This chapter is expected to provide a guidance for constructing tensor-based recommender systems, which encourages more diverse recommendation while keeps satisfactory prediction accuracy. Currently the creativity is valuated based on generating new combination of existing items. Creative construction for data from previously unexplored domain based on current knowledge are also appealing for future targets.

This model was first put forward in [54]. Preliminary results on the Bayesian surprise based recommendation were published in [56].

Chapter 5

Deep Convolutional Factorization based Transfer Learning

5.1 Introduction

There have been significant advances in deep models for a wide variety of machine learning tasks and applications. However, many of these improvements in performance are attributed to large amount of labeled training data. For example, recognition and classification algorithms relying on high capacity convolutional neural network (CNN) models require millions of supervised images for initial training [109]. For new tasks lacking of labeled data, how to utilize the appropriate representation from labeled data at hand becomes an important task. For example, if we only had ten labeled images of furniture against hundreds of other unlabeled ones, classifier directly training on these images alone would easily suffer from overfitting, and hence with poor classification performance. However, if the source domain has much more labeled furniture images, or labeled images that have similar distinguishable features, we may transfer the canonical information inherited in the source tasks to the sparse label domain during the learning stage. This is related to supervised or semi-supervised transfer learning problem [73]. Since our aim is to develop methods for transferring knowledge from related tasks in the source domain towards new tasks in a separate target domain, it is imperative to discover good ways of effective transfer across related learning problems.

Learning a classifier considering the differences between source and target distributions is also known as domain adaptation. It is also a hard problem to find the relatedness between the tasks and the domains. Previous approaches to domain adaptation suggest to build the mappings between the source and the target domains, so that the features and classifier learned during the training stage of source data can be applied to the target domain [28]. Unlike most previous papers on domain adaptation, which work with fixed feature representations, we focus on combining domain adaptation and deep multi-task feature learning, so as to understand the output of each layer, as well as to develop problem-independent feature extraction methods to solve different but related tasks. This is made possible by the initial success of applying deep representation learning on multi-task learning aspect [12], where the model is trained simultaneously on images from different classes. There are also recent researches on network-to-network (holistic) transfer [27, 90], where a subset of features obtained from the source deep networks are properly shared for target tasks.

In order to find which and how tasks are related, we should first know what classes are in each domain, i.e. having an effective model for each domain. However, as pointed out by Tzeng et al. [90], to adopt an effective transferring, the domains are expected to be related and sharing similar features, which creates a cyclic dependency. Our focus is hence to embed domain adaptation into the layer-wise task learning of representations, so that the final classification criteria, both discriminative and invariant to the domain shifts, are met based on features along each layer of the deep hierarchies. Hence, we focus on learning features that combine discriminative task characteristics and domain invariance. After constructing a hierarchical network in a convolutional factorization way, the characteristics of each layer are made related to

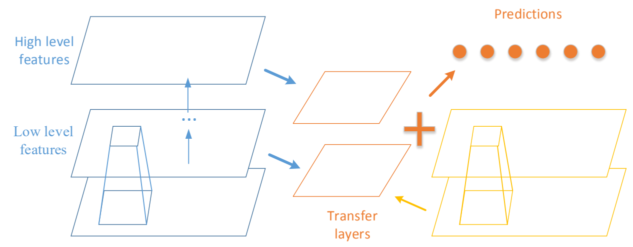


Figure 5.1: The proposed method DCFTL for asymmetric knowledge transferring from a source hierarchical deep factorization to a target domain.

[51, 12]. Specifically, the deeper-layer dictionary elements are in an intuitively visual form while the shallower-layer dictionaries are more general. If the two domains are quite similar in nature, i.e. large amounts of data in the same category, we would expect that they share quite specific features from the deeper layers. Otherwise, if the two domains are related but mostly distinct, only the shallow layer features, such as edges and lines in image analysis will be shared. This deep convolutional factorization based knowledge transfer, named as DCFTL, takes an asymmetrical form by projecting the instances onto the latent source manifold from source task to target task only. And the factorization results from each layer provide physically interpretable dictionary elements. As presented in Figure 5.1, the whole framework of domain feature representation and transfer learning is graphical model based, making it benefit from the advantages of traditional graphical models, including seamless merge of inference and learning, the ability to handle missing data, and clear interpretability of conditional independence [40]. The deep structure (in blue color) is our source domain, while the transfer parts (in orange color) can only be the first layer canonical information or holistic knowledge from all the layers. The predictions are made based on a combination of the canonical information from the target domain (in yellow color), as well as the transfer one.

5.2 Related Work

There have been various approaches proposed in recent years to solve the domain adaptation problem. In most cases, the similarity between domains is measured by the distance between the source and target subspace representation [20]. In the symmetric approach, identical processing mechanism is adopted in both domains. Representative mechanism includes projection of both source and target tasks onto shared space [53]. For the supervised and semi-supervised adaptation scenarios, when only a limited amount of labeled data is available in the target domain, some approaches focus on constructing a target classifier regularized against the source classifier [24]. Raina et al. [73] proposed a *self-taught learning* setting, in which the label spaces between the source and target domains are likely to be different. This implies that the side information of the source domain cannot be used directly leading to the unavailability of source labels.

There has also been a great amount of recent work on deep representation of data, which hierarchically organize multiple nonlinear transformations with the goal of yielding more abstract and ultimately more useful representations. Deep learning methods such as deep belief networks, sparse coding-based methods, convolutional networks, and deep Boltzmann machines have already been successfully applied to a variety of tasks in pattern recognition, computer vision, audio processing, natural language processing, and information retrieval. Using deep representation effectively reduces the effect of domain shifts [94] and also enhances the learning of invariant representation [14]. However, training these networks requires labels for each instance, so it is not applicable for unsupervised or semi-supervised settings.

Glorot et al. [29] proposed to learn robust feature representations with stacked denoising auto-encoders (SDAE) [92] for domain adaptation with deep learning structure. SDAE is a feed-forward neural network for learning representations and reconstructing the input data. Since usually the input data are randomly corrupted by noise, it aims to undo the effect through finding representative structures. SDAEs can be stacked into deep learning architectures with the outputs of their intermediate layers being used as input features for Support Vector Machine (SVM) classification. Several extended models based on SDAE are proposed for domain adaptation. Marginalized stacked denoising auto-encoder (mSDA) [13] is a variant of SDAE proposed, which has been shown to be more effective and efficient. Its linear denoising step is followed by a non-linear step, which is just a hyperbolic tangent function.

Recently, some new CNN based architectures are proposed for multi-task learning and domain adaptation. Tzeng et al. combine a domain confusion and softmax cross-entropy losses to train the network with the target data [90]. But this late fusion strategy optimizes the loss function based heavily upon the ultimate tuning results of CNN. Kandemir introduces a two-layer feed-forward Gaussian process based Bayesian model for asymmetric transfer learning [41], that jointly learns separate discriminative functions from the source and target features to the labels, with variational approximation employed. It adopts an early confusion of the source and target layer, which facilitates the adaptation step. However, how the deeper layer of the source domain would influence the knowledge is not discussed.

5.3 Model Description

5.3.1 Problem Formulation

Considering the transfer learning scenario, observations are from two domains, i.e. *source* domain s and *target* domain t. Let $X_d \in \mathbb{R}^{N_d \times P}$, where P is the feature dimension, $d = \{s, t\}$ with N_d instances. $\mathcal{Y} \in \mathbb{R}^{N_d \times C_d}$ is the output label space, with C_d a finite number of categories possessed by each domain. From the *source* domain, data $X_s = \{\mathbf{x}_1, \cdots, \mathbf{x}_{N_s}\}$ are sampled, while from the *target* domain data

$X_t = \{\mathbf{x}_1, \cdots, \mathbf{x}_{N_t}\}$ with only partial or no labels sampled. The feature distributions from the two domains can be different. Our goal is to learn a feature representation, given samples from *source* domain and *target* domain, and a classifier based on the representations learned to predict the labels of data from the *target* domain.

To learn a joint model, each instance in both domains is expanded in terms of dictionaries, which are defined by compact canonical elements. Moreover, the target domain is assisted with the dictionary loading information transferred from the source domain. We consider the transfer learning setup, where each sample in X_d is expanded in terms of dictionaries defined by compact canonical elements $D_d \in \mathbb{R}^{n_d \times K}$, with $n_d \ll N_d$. The dictionary elements are designed to capture the local structure of input data. The spatial shifts of the dictionaries convolve with the weight matrix W, as in a typical convolutional mode [51]. The n_d^{th} sample of X_d can be represented as:

$$X_{n_d} = \sum_{k=1}^{K} \mathbf{w}_{n_d k} * \mathbf{d}_k + \epsilon \tag{5.1}$$

where $*$ is the convolutional operator, $\mathbf{d}_k$ is the row vector of dictionary and ϵ is the residual. Viewed as a special case of factorization, the elements of matrix W can also be written in a $\{w_{n_d ki}\}_{i \in \mathcal{S}}$ format, where $\mathcal{S}$ corresponds to all possible shifts. A "max-pooling" step is applied to W, after which the sample weights are stacked as input to the next layer. Because of the max-pooling step, the basic computational complexity decreases with increasing hierarchies, as the number of spatial locations decreases.

5.3.2 Approach

Inspired by the motivations and promising results of self-taught learning and deep learning in domain adaptation, a Bayesian deep factorization based model to address the shift of distribution in two domains is proposed with dependencies of parameters shown in Figure 5.2. As a complete graphical model based deep transfer learning framework, this method takes the advantages of merging learning and inference as well as interpretability for conditional dependencies. For the source domain, the observation is reconstructed as

$$\prod_{n_s=1}^{N_s} \mathcal{N}(\mathbf{X}_{n_s}; \sum_{k=1}^{K_1} \sum_{i \in \mathcal{S}} w_{n_s ki}^{(s)} \mathbf{d}_{ki}^{(s)}, \gamma^{-1} \mathbf{I}_P) \mathcal{G}(\gamma; a_1, b_1). \tag{5.2}$$

The priors for source canonical weights $\mathbf{w}^{(s)}$ and dictionaries $\mathbf{d}^{(s)}$ are formed as Normal-Gamma distributions (equation 5.3), where the hyper-parameters of Gamma distribution $\{c_1, d_1, e_1, f_1\}$are set to enable large α and β for sparsity imposing on weight matrix.

$$\begin{aligned} &\mathcal{N}(\mathbf{w}^{(s)}; 0, \alpha^{-1}) \mathcal{G}(\alpha; c_1, d_1), \\ &\mathcal{N}(\mathbf{d}^{(s)}; 0, \beta^{-1}) \mathcal{G}(\beta; e_1, f_1). \end{aligned} \tag{5.3}$$

The transfer between the two domains is realized through a latent layer-wise transfer space $\mathbf{V}$, which also incorporates the projection of the target instances probabilistically. The two representations from the layers of source and target domains are blended into one representation through Bernoulli weight averaging with the mixture hyper-parameterized by a conjugate Beta distribution. For both domains, the output layer takes the representation as features for label prediction. The latent space $\mathbf{V}$ for transfer is set as

$$\mathcal{N}(\mathbf{V}; \mathbf{W}^{(\mathbf{s})}, \eta^{-1} \mathbf{I}). \tag{5.4}$$

The likelihood for target domain observations is

$$\mathcal{N}(\mathbf{X}_{n_t}; \sum_{k=1}^{K_1} \sum_{i \in \mathcal{S}} [(1 - b_{n_t ki}) w_{n_t ki}^{(t)} + b_{n_t ki} \mathbf{v}_{ki}] \mathbf{d}_{ki}^{(t)}, \lambda^{-1} \mathbf{I}_P), \tag{5.5}$$

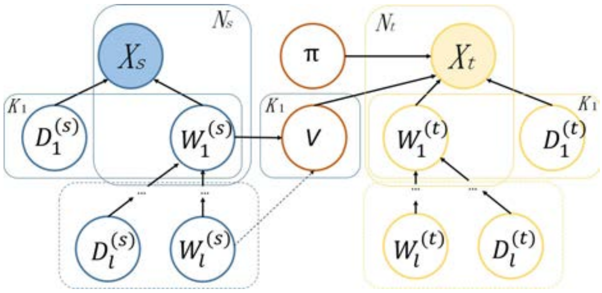


Figure 5.2: The graphic representation of the proposed model. The colors correspond to different parts in Figure 5.1. The shaded nodes are observations. For clarity, all the hyper-parameters are omitted.

and the likelihood for Beta-Bernoulli combination of transfer and target domain dictionary weight is

$$\prod_{n_t=1}^{N_t}\prod_{k=1}^{K_1}\mathcal{B}er(b_{n_tki};\pi_k)\mathcal{B}eta(\pi_k;\frac{t_1}{K_1},\frac{t_2(K_1-1)}{K_1}). \tag{5.6}$$

As illustrated in the dashed square of Figure 5.2, each factor loading matrix can be further divided into L layers, and one can choose which layers to transfer. Using W_{l-1} as the input to the l^{th} layer, model fitting is performed analogous to layer-1. This enables flexible layer-wise transfer, as on one hand, we can conduct holistic transfer, that is only transferring the information from the deepest layer without projection back to the first layer; on the other hand, we can project each layer separately onto the latent transfer space. This will be further discussed in the following algorithmic section.

5.4 Learning Algorithms and Inference

5.4.1 Layer-wise Transfer Algorithms

By defining the number of layers L of deep structure and the factor number K_l of each layer, one can recursively apply convolutional factorization on both the source and target domain data, and map each layer to generate different granularity of

Algorithm 5.1 Source Layer-1 to Target Layer-1 DCFTL

```
Input: source sample and label collection $\{X_s, \mathcal{Y}\}$, target sample $X_t$ (label optional), number of layers $L$, number of factors for each layer $K_l$, and max pooling ratio
Begin
Initialize $D^{(s)}$, $W^{(s)}$, $V$, $B$, $D^{(t)}$, $W^{(t)}$
for iteration $i$ do
  for source domain do
    1. Sample factor loading matrix $W^{(s)}$ conditionally conjugate posteriors according to (5.7).
    2. Update layer-1 dictionaries $D^{(s)}$.
  end for
  for transfer latent space do
    Update transfer weight matrix $V$ according to (5.10).
  end for
  for transfer to target domain do
    1. Update the proportions Bernoulli $B$ and domain specific weight matrix $W^{(t)}$ according to (5.13) simultaneously.
    2. Sample conditionally conjugate posteriors of $D^{(t)}$ according to (5.16).
  end for
end for
Do feature augmentation and train a classifier on target domain.
```

feature transformations.

To illustrate that the framework can be flexibly generalized to multi-layer wise transfer, we adjust DCFTL in the following three settings. The first is direct transfer from layer-1 dictionaries to the target data. Instead of considering the probability of weights combination, only latent transfer space $\mathbf{V}$ is used as prediction feature. In other words, $b_{n_t ki}$ is set to 1 with probability equal to one. The second is layer-1 to layer-1 transfer, which is used as an illustrative derivation in the inference section. The third is layer-2 to layer-1 transfer, that is for the source domain, the second layer dictionaries are first projected back on to the first layer, and then transfer to the latent space. We name them 'DCFTLl1-t', 'DCFTLl1-l1', 'DCFTLl2-l1' respectively.

Noted that different from previous low-rank transfer learning methods [84], we treat the transferred latent space from source domain as part of the dictionaries that employed to reconstruct the whole data in the target domain. For 'DCFTLl1-l1', the overall algorithm is summarized in Algorithm 5.1.

This algorithm can be easily generalized to deeper layer or shallower layer transfer. For example, when conducting layer-2 to layer-1 transfer (‘DCFTLl2-l1’), after applying max-pooling, each shift of $W_{n_d k}$ is mapped to a $\hat{W}_{n_d k}$ with the m^{th} value corresponding to the largest-magnitude component within the m^{th} region [11]. $\hat{W}_{n_d k}$ is then used as input to the next layer factorization. Since the priors for this step can be identical to those of the first layer, the inference can be easily derived. In the following subsection, we use ‘DCFTLl1-l1’ as an illustration of the inference.

5.4.2 Bayesian Inference

As the posterior density of DCFTL is intractable, approximate inference is needed. We conduct Gibbs sampling for the posterior update. Based on the full likelihood of the proposed model, all conditional distributions used to draw samples are analytic, and at each iteration, we can draw the samples from conditional distributions. Here for clarity, we take a layer-1 transfer inference as an example to illustrate the sampling steps.

Sampling $w^{(s)}_{n_s ki}$: We start with the source domain canonical information inference. Since the generative distributions are conjugate, the posterior distribution of the factor loading matrix is multivariate normal distribution, represented as:

$$P(w^{(s)}_{n_s ki}|-) = \mathcal{N}(\mu_{n_s ki}, \Sigma_{n_s ki}) \tag{5.7}$$

where

$$\Sigma_{n_s ki} = (\gamma \mathbf{d}^{(s)T}_{ki} \mathbf{d}^{(s)}_{ki} + \alpha_{n_s ki} + \eta)^{-1}, \tag{5.8}$$

$$\mu_{n_s ki} = \Sigma_{n_s ki}(\gamma X^{(s)T}_{n_s ki} \mathbf{d}^{(s)}_{ki} + \eta \mathbf{v}_k). \tag{5.9}$$

Here, $X_{n_s ki}$ is the most recent sample. The dictionary sampling for source domain is similar to the inference in [12]. As it does not involve with the transfer part, the inference for each layer is identical.

Sampling $\mathbf{v}_{ki}$: Since the transfer space is based on a transformation of the source factor loading, the increase or decrease of convolutional factorization layers would not affect its format. Its posterior distribution can be represented as:

$$P(\mathbf{v}_{ki}|-) = \mathcal{N}(\zeta_{ki}, \boldsymbol{\Psi}_{ki}), \tag{5.10}$$

where

$$\boldsymbol{\Psi}_{ki} = (\lambda \mathbf{b}_{ki}^T \mathbf{b}_{ki} \mathbf{d}_{ki}^T \mathbf{d}_{ki} + \eta)^{-1}, \tag{5.11}$$

$$\zeta_{ki} = \boldsymbol{\Psi}_{ki}(\lambda \mathbf{b}_{ki} X_{n_tki}^T \mathbf{d}_{ki}). \tag{5.12}$$

Sampling $w_{n_tki}^{(t)}$: For the direct source layer-1 to target transfer, $\mathbf{b}_{ki} == 1$, the distribution expression of $w_{n_ski}^{(t)}$ is identical to the case that deep convolutional factorization directly applies on the target domain, with the factor loading matrix replaced by the latent transfer matrix.

$$P(w_{n_tki}^{(t)}|-) = \mathcal{N}(\vartheta_{n_tki}, \Omega_{n_tki}) \tag{5.13}$$

where

$$\Omega_{n_tki} = ((1 - \mathbf{b}_{ki})^T (1 - \mathbf{b}_{ki}) \mathbf{d}_{ki}^T \mathbf{d}_{ki} + \tau)^{-1}, \tag{5.14}$$

$$\vartheta_{n_tki} = \Omega_{n_tki}(2\lambda(1 - \mathbf{b}_{ki})(X_{n_tki}^{(t)T} \mathbf{d}_{ki} - \mathbf{b}_{ki}^T \mathbf{v}_{ki} \mathbf{d}_{ki}^{(t)T} \mathbf{d}_{ki}^{(t)}) \tag{5.15}$$

Sampling $\mathbf{d}_{ki}^{(t)}$: If $\mathbf{b}_k == \mathbf{0}$, which indicates that no transfer learning is considered, then the expression is identical to deep learning on the target data.

$$P(\mathbf{d}_k^{(t)}|-) = \mathcal{N}(\xi_k, \boldsymbol{\Phi}_k) \tag{5.16}$$

where

$$\boldsymbol{\Phi}_k = (\sum_{n_t=1}^{N_t} \lambda \parallel 1 - \mathbf{b}_k \parallel_2^2 \parallel \mathbf{w}_{n_tk}^{(t)} \parallel_2^2 + \lambda \mathbf{b}_k^T \mathbf{b}_k \parallel \mathbf{v}_k \parallel_2^2 + \beta \tag{5.17}$$

$$+ \sum_{n_t=1}^{N_t} 2\lambda(1 - \mathbf{b}_k)^T \mathbf{b}_k \parallel \mathbf{w}_{n_tk}^{(t)} \parallel)^{-1},$$

$$\xi_k = \boldsymbol{\Phi}_k(\sum_{n_t=1}^{N_t} \sum_{i \in \mathcal{S}} \lambda(1 - \mathbf{b}_k) w_{n_tki}^{(t)} X_{n_tki} - \sum_{n_t=1}^{N_t} \sum_{i \in \mathcal{S}} \lambda \mathbf{b}_k \mathbf{v}_{ki}^T X_{n_tki}) \tag{5.18}$$

5.5 Experiments

We perform evaluation of the proposed approach on several popular image datasets, including large-scale datasets of small images popular with deep learning testing, i.e. ‘MNIST’ [99], CIFAR-100 [47] and the ‘Office’ datasets [79], which is used as standard benchmark for visual domain adaptation challenge.

The baseline model for our comparison is the source-only model, which is trained without target-domain data. The learned two layers of features are directly used as training data for classifiers on the target domain. The basic convolutional factorization model is trained on the target domain with class labels revealed. This approach serves as an lower bound, assuming that target data are abundant and the shift between the domains is not significant. The classifier is designed similar to Lee et al.'s work [51], in which we perform classification based on layer-one coefficients, or on both layer-1 and layer-2 using a standard SVM [25]. Another comparison model is based on target-only learning, where the dictionaries learned solely from the target observations are used as features for classifier training.

5.5.1 Parameter Settings

The hyper-parameters parameters needed to set is limited and can be set in a standard way [88]. Specifically, for the Gamma-Normal distributions for factorization and transfer $a_1 = b_1 = f_1 = 10^{-6}$, $c_1 = e_1 = 1$ and $d_1 = 10^{-3}$. For the Beta-Bernoulli part, $t_1 = t_2 = 1$. The dictionary sizes for the first and second layer are 4 and 8, with a max pooling ratio 3. For the Gibbs sampler, 500 burn-in iterations are used, with 300 collection samples (the results vary little after 100 iterations).

5.5.2 MNIST Digits - Alphabet

Our first experiment deals with the MNIST dataset, where we use either the digits or alphabet dataset as the source domain, while the other one as the target domain. This experiment can be regarded as a case where two domains share great similarities, while the tasks themselves are distinct. Standard five-fold training-test splits are considered, with $60,000$ MNIST handwritten number samples for training and $10,000$ for testing. For the testing images we consider unsupervised adaptation with no labels provided.

Classification results are presented in Table 5.1. For source CNN, all the dictionary weights learned from source domain are used as training samples, while all the samples in the target domain are used for testing. For target CNN, the samples in the target domain are split into five folders, with four for training and one for testing. "DCFTLl1-t" indicates direct transfer source layer-1 features to target data; "DCFTLl1-l1" indicates transfer source layer-1 features to target layer-1 dictionaries; while "DCFTLl2-l1" indicates first projecting the layer-2 features back to layer-1 in the source domain, and then transfer the layer-1 features to the first layer in the target domain.

We can see that the performance of directly transfer source layer-1 dictionaries to target data is not satisfactory, while a layer-1 to layer-1 transfer can reach similar accuracy comparing to layer-2 to layer-1 transfer. Also we can notice that when treating as source or target, the transfer is not equally difficult. When Digit data is used as source information, layer-1 to layer-1 transfer provides better representation comparing to layer-2 to layer-1 transfer. On the other hand, when Alphabet collection is used as source dataset, the situation is reversed. This could be contributed to the more dynamic dictionary elements that alphabet images provide.

Table 5.1: Classification accuracies for MNIST classifications for different source and target domains

source target	Digit Alphabet	Alphabet Digit
Source CNN layer 1+2	.4465	.3883
Target CNN layer 1+2	.4819	.4012
DCFTLl1-t	.4567	.3889
DCFTLl1-l1	.4827	.4032
DCFTLl2-l1	.4810	.4043

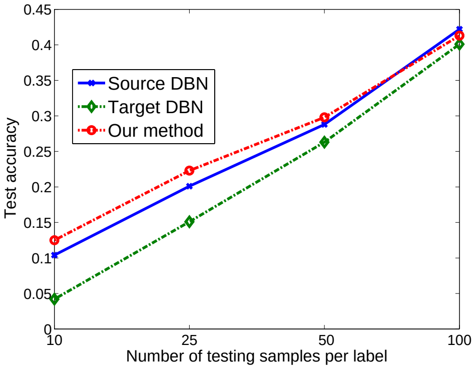


Figure 5.3: Test set classification accuracy for different number of samples in the target domain

5.5.3 CIFAR-100 with Few Examples per Class

The CIFAR-100 dataset consists of 32 × 32 color images categorized by 100 classes, with each class further divided into 20 groups of 5 each. We use the fine labels of this dataset to demonstrate the utility of transfer learning when the labeled samples are few. For the 600 samples of each class, we randomly select 500 for training, and create 5 subsets of the remaining 100 samples by randomly choosing 10, 25, 50, 100 samples per class, and test the models on each subset.

The test performance of these models is compared in Figure 5.3. We observe that when the number of samples for testing is small, the proposed model already provides improvement over the baseline. The improvement diminishes as the available sample size increases.

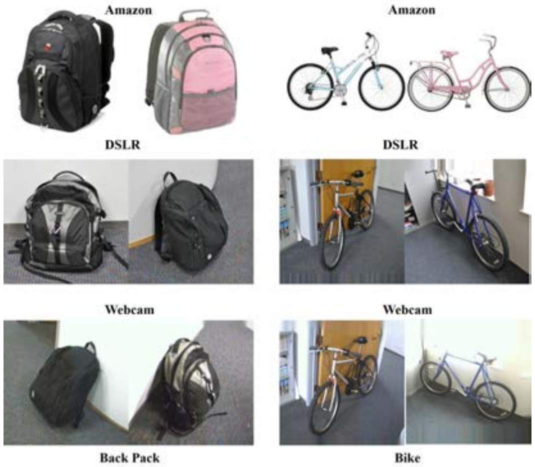


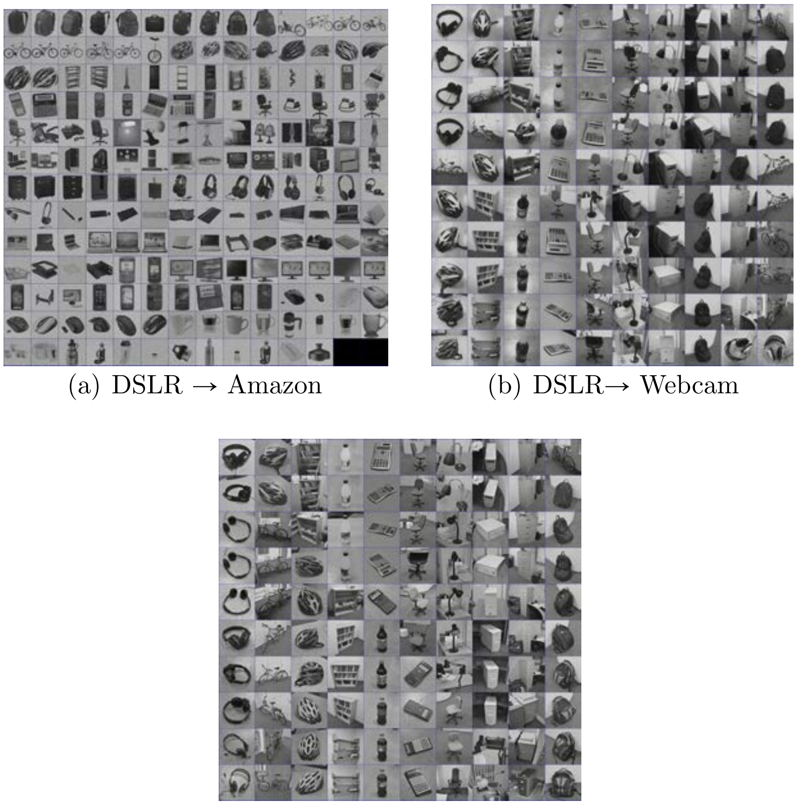
Figure 5.4: Example of the three domains labeled by "back pack" and "bike".

5.5.4 Office Dataset

The benchmark office dataset contains 10 categories taken in four different conditions, corresponding to three domains: Amazon, dslr, and webcam. For each of the six domain shifts, we conduct cross-validation for five train/test splits, which are generated by sampling examples from the full set of images per domain. In the source domain, we follow the standard protocol for this dataset and generate splits by sampling 10 examples per category for each of the three domains. In each case, we train on the source dataset and test on a different target domain dataset, considering the shifts between domains (as shown in Figure 5.4). In Table 5.2, we

Table 5.2: Classification accuracies for Office dataset classifications. A: Amazon; W: Webcam; D:DSLR

source	A	A	W	W	D	D
target	W	D	A	D	A	W
Source DBN Layer-1	.5342	.5726	.3794	.8696	.5115	.9130
Source DBN Layer-1+2	.5625	.6112	.3987	.9024	.5401	**.9304**
Target DBN Layer-1	.8137	.8106	.4998	.8248	.5970	.8141
Target DBN Layer-1+2	**.8266**	.8176	.6001	.8292	.6170	.8232
DCFTLl1-t	.6522	.5566	.3268	.8600	.4755	.7301
DCFTLl1-l1	.8140	.8095	.5018	**.9097**	.5970	.8141
DCFTLl2-l1	.8143	**.8196**	.5005	.9060	**.6175**	.8159



(a) DSLR → Amazon

(b) DSLR→ Webcam

(c) Webcam → DSLR

Figure 5.5: Samples of reconstructed images from each domain

present the accuracies and compare our method in three settings to the state-of-the-art deep belief networks (DBN). We find that for most transfer situations, our method outperforms the baseline. We can notice that the gap between domains significantly affects the performance. As the images in the webcam dataset are more diverse and have enriched background, direct source DBN based method performs best since through max-pooling, it directly filters out the noise information. While in the opposite direction, our method performs best in a Webcam to dslr adaptation.

Another point worth noticing is the influence of which layer to transfer. As shown in the fourth and fifth columns of Table 5.2, transfer from layer-1 to layer-1 outperforms layer-2 to layer-1 transfer in both cases where webcam is the source domain. This could be contributed to the disturbing background of the images in this condition, as when going deeper, the background features would confuse the classifier.

5.6 Summary

Targeting the scenario of lacking labeled data, we propose a deep convolutional factorization based transfer learning method, which aims to seek most shared discriminative features within source data to facilitate the unseen target learning. It captures the layer-wise feature similarity, and transfers the knowledge from the source domain to the target one utilizing the layer-wise factor weights. The graphic representation mechanism for both deep and transfer learning encourages interpretability of conditional dependencies and a flexible generalization of layer-wise implementation. Series of benchmark experiments on image classification prove that the method can provide competitive performance comparing to the state-of-art baselines.

Starting from a graphical representation perspective, this work is expected to discover how the layer-wise similarity can be utilized into a layer-wise transferring between the domains. This would hopefully put some insights on the question such as how many layers is enough for deep domain adaptation. The work is under preparation for submitting [57].

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, four novel hierarchical Bayesian models for multi-relational sparse data applications on latent feature learning have been introduced, with the following contributions:

1. Ai-hLDA incorporates rich auxiliary information to arrange shorter length documents along paths of a tree for taxonomy building. It takes advantage of the hierarchical nature, and jointly models word and auxiliary information as a generative process. With flexible usage of the supplemental/auxiliary tagging observations and a nested Chinese Restaurant Process prior, hierarchical topic patterns can be discovered for documents with short length.

2. Targeting the scarceness issue of item and its features, a unified Dirichlet mixture probit model for information scarcity (DPIS) has been developed. It learns topics over scarce information by directly modeling the generation of records. Specifically, the clicking history of a user is assumed to consist of a mixture of topics, and each record sample as a whole is assigned a specific topic with certain probability, from which its corresponding tags are generated. In the generative model, the words describing each item are drawn from the vocabulary of the record. The topic proportions, a multinomial vector, are also used as an *item vector* to describe whether a user

viewing or not viewing the item. Although DPIS does not model the user-level generative process, the link between active users and video pools can be learned through a probit model, with a Laplacian prior enforced on the sparse parameters. Hence, our work differs from standard approaches, as the *item vectors* now serve two roles: explaining both the words that tag the video record; and capturing the collaborative component. In this way, the overall model can not only cluster the semantic topics of the video records based on aggregated item features for active users, but can also combine the observed features and generate topics for newly synthesized data. Here, we employ a collapsed Gibbs sampling inference to find the posterior solution to the topic discovery and the probit parameters, which is convenient to implement.

3. Exploring multi-relational data from a tensor perspective, a fully conjugate deep probabilistic approach for tensor decomposition is proposed. Based on the Canonical PARAFAC (CP) decomposition, the model is capable of clustering the three-way data along each direction simultaneously. To find a more compact representation in the latent space of each mode, a multi-layer factorization is imposed on the mode factor matrix to incorporate nonlinear mapping. Instead of relying on ad-hoc or cross-validating parameter selection, the rank of the core tensor and the factor number for each layer of the deep network can all be automatically determined. As a fully conjugate Bayesian model, efficient Gibbs sampling inference is facilitated, with an improving performance for tensor reconstruction and prediction accuracy.

4. To solve the low diversity issue of tensor-based recommender system, a metric for computational creativity quantization is put forward. Through incorporating *Bayesian surprise* in the probabilistic ranking, a compromise between accuracy and serendipity is made during the recommendation stage, making the candidate item more diverse.

5. We also consider the scenarios where labeled data is lacked when there are multiple sources of data. Deep convolutional factorization based domain adaptation

method provides a semi-supervised way for transfer learning. It captures the layer-wise feature similarity, and transfers the knowledge from the source domain to the target one by utilizing the layer-wise factor weights.

The papers, published, in press or submitted, are the partial outputs of my PhD study. The work on ai-hLDA is currently under review by a journal[55]. Preliminary results of DPIS model were published in [59]. More intensive discussions of this model are currently under review [60]. The DCPF model was first put forward in [54]. Results on the Bayesian surprise based video recommendation were published in [56]. A more detailed elaboration on the framework and experiment results is under preparation for submission [58]. The work on DCFTL is also under preparation for submission [57].

6.2 Future Work

Probabilistic graphic model for latent feature learning is a research area with outstanding existing work as well as promising academic values, thus there exists a number of problems waiting for being solved. Several future work are described here:

- First, combining deep learning and transfer learning, we can better handle the cold-start problem. This is especially useful for a new domain, where historical data is scarce. This concerns about how the two domains are related, and which part of the inner structures can be transferred from the previous domain.

- Second, we consider the study of personalization and creativity is of high potential. We proposed the Bayesian surprise index as a quantitative measure, but currently the way of incorporating it into the probability ranking is naive. How to leverage this measure more effectively, while maintaining satisfactory accuracy is a direction worth considering.

- Third, deep generative models, such as Deep Belief Networks (DBN), variational autoencoders (VAE) and generative adversarial networks (GAN), are gaining more and more attention. Our study on modifying these models for transfer learning as well as inner structure understanding is still at a preliminary stage. In the future, we plan to continue the understanding of the mechanism of the layer-wise model, and applying them to the prevalent semi-supervised and unsupervised situations.

As will be required by applications, more priors and models should be developed and we believe that the ideas and various derivations presented here can be used as blueprints for future research on exploration of richer priors, more efficient inference and more scalable for big data.

Bibliography

[1] McCallum, Andrew, Xuerui Wang, and Corrada-Emmanuel, Andrés. Topic and role discovery in social networks with experiments on Enron and academic email. *Journal of Artificial Intelligence Research*, pages 249–272, 2007.

[2] Asim Ansari, Skander Essegaier, and Rajeev Kohli. Internet recommendation systems. *Journal of Marketing Research*, 37(3):363–375, 2000.

[3] Ramnath Balasubramanyan, Bhavana Dalvi, and William W Cohen. *From topic models to semi-supervised learning: Biasing mixed-membership models to exploit topic-indicative features in entity clustering*, pages 628–642. Springer, 2013.

[4] Pierre Baldi and Laurent Itti. Of bits and wows: A Bayesian theory of surprise with applications to attention. *Neural Networks*, 23(5):649–666, 2010.

[5] Justin Basilico and Thomas Hofmann. Unifying collaborative and content-based filtering. In *Proceedings of the 21st International Conference on Machine Learning*, page 9. ACM, 2004.

[6] Yoshua Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

[7] Anirban Bhattacharya and David B Dunson. Sparse Bayesian infinite factor models. *Biometrika*, 98(2):291–306, 2011.

[8] David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[9] David M Blei, Thomas L Griffiths, and Michael I Jordan. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM (JACM)*, 57(2):7, 2010.

[10] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet allocation. *the Journal of Machine Learning Research*, 3:993–1022, 2003.

[11] Y-Lan Boureau, Francis R. Bach, Yann LeCun, and Jean Ponce. Learning mid-level features for recognition. In *2010 IEEE Conference on Computer Vision*

and Pattern Recognition (CVPR), pages 2559–2566. IEEE Computer Society, 2010.

[12] Bo Chen, Gungor Polatkan, Guillermo Sapiro, David Blei, David Dunson, and Lawrence Carin. Deep learning with hierarchical convolutional factor analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 2013.

[13] Minmin Chen, Kilian Q. Weinberger, Fei Sha, and Yoshua Bengio. Marginalized denoising auto-encoders for nonlinear representations. In *ICML*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1476–1484. JMLR.org, 2014.

[14] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 539–546. IEEE, 2005.

[15] Wei Chu and Zoubin Ghahramani. Probabilistic models for incomplete multidimensional arrays. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, pages 89–96, 2009.

[16] Shui-Lung Chuang and Lee-Feng Chien. Taxonomy generation for text segments: A practical web-based approach. *ACM Transactions on Information Systems (TOIS)*, 23(4):363–396, 2005.

[17] Andrzej Cichocki, Rafal Zdunek, Anh Phan, and Shun-ichi Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation.* John Wiley & Sons, 2009.

[18] Simon Colton, Geraint A Wiggins, et al. Computational creativity: the final frontier? In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 21–26, 2012.

[19] Andreas Damianou. *Deep Gaussian processes and variational propagation of uncertainty.* PhD thesis, University of Sheffield, 2015.

[20] Hal Daume III and Daniel Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, pages 101–126, 2006.

[21] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. The YouTube video recommendation system. In *Proceedings of the 4th ACM Conference on Recommender Systems*, pages 293–296. ACM, 2010.

[22] Jana Diesner, Terrill L Frantz, and Kathleen M Carley. Communication networks from the Enron email corpus “It’s always about the people. Enron is no

different". *Computational & Mathematical Organization Theory*, 11(3):201–228, 2005.

[23] Xiaowen Ding, Bing Liu, and Philip S Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 231–240. ACM, 2008.

[24] Lixin Duan, Dong Xu, and Ivor W Tsang. Domain adaptation from multiple sources: A domain-dependent regularization approach. *IEEE Transactions on Neural Networks and Learning Systems*, 23(3):504–518, 2012.

[25] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: a library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[26] Zhe Gan, Changyou Chen, Ricardo Henao, David Carlson, and Lawrence Carin. Scalable deep poisson factor analysis for topic modeling. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1823–1832, 2015.

[27] Yaroslav Ganin and Victor S. Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1180–1189, 2015.

[28] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-Adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

[29] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning*, pages 513–520. Omnipress, 2011.

[30] Miha Grčar, Dunja Mladenič, Blaž Fortuna, and Marko Grobelnik. *Data sparsity issues in the collaborative filtering framework*. Springer, 2006.

[31] Thomas Griffiths and Zoubin Ghahramani. Infinite latent feature models and the indian buffet process. In *Advances in Neural Information Processing Systems 18 (NIPS)*, pages 475–482. Gatsby Unit, 2005.

[32] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[33] Jiawei Han and Micheline Kamber. *Data mining: Concepts and techniques.* Morgan Kaufmann, 2000.

[34] Richard A Harshman. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 1970.

[35] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[36] Qirong Ho, Jacob Eisenstein, and Eric P Xing. Document hierarchies from text and links. In *Proceedings of the 21st International Conference on World Wide Web*, pages 739–748. ACM, 2012.

[37] Liangjie Hong and Brian D Davison. Empirical study of topic modeling in twitter. In *Proceedings of the 1st Workshop on Social Media Analytics*, pages 80–88. ACM, 2010.

[38] Changwei Hu, Eunsu Ryu, David Carlson, Yingjian Wang, and Lawrence Carin. Latent Gaussian models for topic modeling. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, pages 393–401, 2014.

[39] Noor Ifada. A tag-based personalized item recommendation system using tensor modeling and topic model approaches. In *Proceedings of the 37th international ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1280–1280. ACM, 2014.

[40] Michael I. Jordan. *Learning in Graphical Models*, volume 89. Springer Science & Business Media, 1998.

[41] Melih Kandemir. Asymmetric transfer learning with deep gaussian processes. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 730–738, 2015.

[42] Paul B Kantor, Lior Rokach, Francesco Ricci, and Bracha Shapira. *Recommender systems handbook*. Springer, 2011.

[43] Younghoon Kim, Yoonjae Park, and Kyuseok Shim. DIGTOBI: a recommendation system for digg articles using probabilistic modeling. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 691–702. International World Wide Web Conferences Steering Committee, 2013.

[44] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[45] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[46] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009.

[47] Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009.

[48] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[49] Solomon Kullback. *Information theory and statistics*. Courier Corporation, 1997.

[50] Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international Conference on Machine learning*, pages 473–480. ACM, 2007.

[51] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 609–616. ACM, 2009.

[52] Li-Jia Li, Chong Wang, Yongwhan Lim, and David M. Blei. Building and using a semantivisual image hierarchy. In *The 23rd IEEE Conference on Computer Vision and Pattern Recognition*, pages 3336–3343, 2010.

[53] Wen Li, Lixin Duan, Dong Xu, and Ivor W Tsang. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1134–1148, 2014.

[54] Wei Lu and Fu-lai Chung. Deep bayesian tensor for recommender system. In *Proceedings of the ECMLPKDD 2015 Doctoral Consortium*, pages 135–144, 2015.

[55] Wei Lu and Fu-lai Chung. Auxiliary information based hierarchical topic model: Mining biased short text. (under review by *Information Sciences*), 2016.

[56] Wei Lu and Fu-lai Chung. Computational creativity based video recommendation. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 793–796, New York, NY, USA, 2016. ACM.

[57] Wei Lu and Fu-lai Chung. A deep graphical model for layered knowledge transfer. (under preparation), 2016.

[58] Wei Lu and Fu-lai Chung. Deep tensor recommender system with computational creativity based probabilistic ranking. (under preparation), 2016.

[59] Wei Lu, Fu-lai Chung, and Kunfeng Lai. Scarce feature topic mining for video recommendation. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, New York, NY, USA, 2016 forthcoming. ACM.

[60] Wei Lu, Fu-lai Chung, Kunfeng Lai, and Liang Zhang. Topic mining for information scarcity. (under review by *Neural Networks*), 2016.

[61] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 165–172. ACM, 2013.

[62] E Meeds and Z Ghahramani. Modeling dyadic data with binary latent factors. In *Advances in Neural Information Processing Systems (NIPS)*, pages 977–984, 2006.

[63] Rosen-Zvi, Michal, Chaitanya Chemudugunta, Thomas Griffiths, Padhraic S-myth, and Mark Steyvers. Learning author-topic models from text corpora. *ACM Transactions on Information Systems (TOIS)*, 28(1):4, 2010.

[64] Morten Mørup. Applications of tensor (multiway array) factorizations and decompositions in data mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):24–40, 2011.

[65] Maks Ovsjanikov and Ye Chen. *Topic modeling for personalized recommendation of volatile items*, pages 483–498. Springer, 2010.

[66] John Paisley, Chong Wang, David M Blei, and Michael I Jordan. Nested hierarchical Dirichlet processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):256–270, 2015.

[67] Sinno Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[68] AJ Perotte, F Wood, and N Elhadad. Hierarchically supervised latent Dirichlet allocation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2609–2617, 2011.

[69] Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th International Conference on World Wide Web*, pages 91–100. ACM, 2008.

[70] Florian Pinel and Lav R Varshney. Computational creativity for culinary recipes. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, pages 439–442. ACM, 2014.

[71] Piyush Rai, Yingjian Wang, and Lawrence Carin. Leveraging features and networks for probabilistic tensor decomposition. In *The 29th AAAI Conference on Artificial Intelligence*, pages 2942–2948, 2015.

[72] Piyush Rai, Yingjian Wang, Shengbo Guo, Gary Chen, David Dunson, and Lawrence Carin. Scalable Bayesian low-rank decomposition of incomplete multiway tensors. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1800–1808, 2014.

[73] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th International Conference on Machine Learning*, pages 759–766. ACM, 2007.

[74] Daniel Ramage, Susan T Dumais, and Daniel J Liebling. Characterizing microblogs with topic models. *The 4th International AAAI Conference on Weblogs and Social Media*, 10:1–1, 2010.

[75] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 248–256. Association for Computational Linguistics, 2009.

[76] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Introduction to recommender systems handbook*. Springer, 2011.

[77] Francesco Ricci, Lior Rokach, and Bracha Shapira. Recommender systems: introduction and challenges. In *Recommender Systems Handbook*, pages 1–34. Springer, 2015.

[78] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 487–494. AUAI Press, 2004.

[79] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. *Adapting visual category models to new domains*, pages 213–226. Springer, 2010.

[80] Ruslan Salakhutdinov. Learning deep generative models. *Annual Review of Statistics and Its Application*, 2:361–385, 2015.

[81] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine Learning*, pages 791–798. ACM, 2007.

[82] Shilad Sen, Jesse Vig, and John Riedl. Tagommenders: connecting users to items through tags. In *Proceedings of the 18th International Conference on World Wide Web*, pages 671–680. ACM, 2009.

[83] Hanhuai Shan and Arindam Banerjee. Mixed-membership naive Bayes models. *Data Mining and Knowledge Discovery*, 23(1):1–62, 2011.

[84] Ming Shao, Dmitry Kit, and Yun Fu. Generalized transfer subspace learning through low-rank constraint. *International Journal of Computer Vision*, 109(1-2):74–93, 2014.

[85] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009.

[86] Yee Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 2012.

[87] Romain Thibaux and Michael I Jordan. Hierarchical Beta processes and the Indian buffet process. In *International Conference on Artificial Intelligence and Statistics*, pages 564–571, 2007.

[88] Michael E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

[89] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[90] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4068–4076, 2015.

[91] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1096–1103. ACM, 2008.

[92] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11:3371–3408, 2010.

[93] Chong Wang and David M Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 448–456. ACM, 2011.

[94] Tong Xiao, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang. Learning deep feature representations with domain guided dropout for person re-identification. *arXiv preprint arXiv:1604.07528*, 2016.

[95] Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff G Schneider, and Jaime G Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of the SIAM International Conference on Data Mining*, pages 211–222. SIAM, 2010.

[96] Zenglin Xu, Feng Yan, et al. Infinite tucker decomposition: Nonparametric Bayesian models for multiway data analysis. *arXiv preprint arXiv:1108.6296*, 2011.

[97] Zenglin Xu, Feng Yan, and Yuan Qi. Bayesian nonparametric models for multiway data analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):475–487, 2015.

[98] Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. A biterm topic model for short texts. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1445–1456. International World Wide Web Conferences Steering Committee, 2013.

[99] LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[100] Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen. Challenging the long tail recommendation. *Proceedings of the VLDB Endowment*, 5(9):896–907, 2012.

[101] Jianhua Yin and Jianyong Wang. A Dirichlet multinomial mixture model-based approach for short text clustering. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 233–242. ACM, 2014.

[102] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Robert Fergus. Deconvolutional networks. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2528–2535. IEEE, 2010.

[103] Jia Zeng. A topic modeling toolbox using belief propagation. *The Journal of Machine Learning Research*, 13(1):2233–2236, 2012.

[104] Xi Zhang, Jian Cheng, Ting Yuan, Biao Niu, and Hanqing Lu. Toprec: domain-specific recommendation through community topic mining in social network. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1501–1510. International World Wide Web Conferences Steering Committee, 2013.

[105] XianXing Zhang and Lawrence Carin. Joint modeling of a matrix with associated text via latent binary features. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1556–1564, 2012.

[106] XianXing Zhang, David B. Dunson, and Lawrence Carin. Hierarchical topic modeling for analysis of time-evolving personal choices. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1395–1403, 2011.

[107] Wayne Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. *Comparing twitter and traditional media using topic models*, pages 338–349. Springer, 2011.

[108] Nan Zheng, Qiudan Li, Shengcai Liao, and Leiming Zhang. Flickr group recommendation based on tensor decomposition. In *Proceedings of the 33rd international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 737–738. ACM, 2010.

[109] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems*, pages 487–495, 2014.