Neurocomputing 127 (2014) 124–134

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Cross domain recommendation based on multi-type media fusion

Shulong Tan [a,*], Jiajun Bu [a], Xuzhen Qin [a], Chun Chen [a], Deng Cai [b]

[a] Zhejiang Key Laboratory of Service Robot, College of Computer Science, Zhejiang University, Hangzhou 310027, China
[b] State Key Laboratory of CAD& CG, College of Computer Science, Zhejiang University, Hangzhou 310027, China

ARTICLE INFO

Article history:
Received 21 December 2012
Received in revised form
6 May 2013
Accepted 11 August 2013
Communicated by Y. Chang
Available online 18 October 2013

Keywords:
Recommender systems
Cross domain
Topic modeling
Latent dirichlet allocation
Transfer learning

ABSTRACT

Due to the scarcity of user interest information in the target domain, recommender systems generally suffer from the sparsity problem. To alleviate this limitation, one natural way is to transfer user interests in other domains to the target domain. However, objects in different domains may be in different media types, which make it very difficult to find the correlations between them. In this paper, we propose a Bayesian hierarchical approach based on Latent Dirichlet Allocation (LDA) to transfer user interests cross domains or media. We model documents (corresponding to media objects) from different domains and user interests in a common topic space, and learn topic distributions for documents and user interests together. Specifically, to learn the model, we combine multi-type media information: media descriptions, user-generated text data and ratings. With this model, recommendation can be done in multiple ways, via predicting ratings, comparing topic distributions of documents and user interests directly and so on. Experiments on two real world datasets demonstrate that our proposed method is effective in addressing the sparsity problem by transferring user interests cross domains.

1. Introduction

Recommender systems attempt to suggest items that target users are likely to be interested in. The most representative recommendation method is Collaborative Filtering (CF) which predicts the preference of a user by combining feedbacks of other users with similar interests. Even though CF methods achieve great successes in practical applications, there are still some problems which limit their performance. One main limitation is the well-known sparsity problem [1,2]. That is, when some users access limited items or some items are used by limited users, it is difficult to predict user interests and overfitting may happen easily.

To alleviate the sparsity problem, auxiliary data, such as users' explicit and implicit feedbacks in other domains, can be used. Fig. 1 shows an example. Assuming the scenario that users leave less preference information (e.g., ratings and comments) in books but much more in movies. It is difficult to recommend books for these users only based on their feedbacks on books, since their interest data in books is limited (i.e., the data is sparse). Fortunately, we can transfer user interests from movies to books. Intuitively, if users like the movie *Harry Potter*, they may also like the book *Harry Potter*. In another more meaningful example, if users watched many science fiction films, they may be interested in books in similar styles.

The key is how to transfer user interests cross domains, even with different media types. Some researchers propose transfer learning methods to solve this problem [3,4]. They assume that the rating matrices in different domains share similar cluster-level rating patterns and consider these patterns as potential candidates to be transferred from the auxiliary domain. Although these methods can alleviate the sparsity problem to some extent, there are still two major limitations. Firstly, they require data in both the auxiliary domain and the target domain to be standardized and structured. It means that the data in both domains are in the form of rating matrices, while in practice this requirement cannot be met sometimes. Secondly, these methods are hard to extend for exploiting other kinds of information, such as media content and user-generated text data.

In practical applications, various kinds of information can be utilized to transfer user interests. For instance, in E-commerce websites such as Amazon, recommender system designers may be interested in transferring user interests cross commodity categories (e.g., electronic products to books). In this example, users' comments can be used to mine user interests, description text can be used to build correlations between commodities from different categories, and so on. In the example of Fig. 1, the useful information includes the content of the media itself, the media description text, some kinds of meta data and user-generated information. To transfer user interests based on these types of information, a nature solution is to build correlations or similarities between objects from different domains. However it is difficult to get these correlations which are consistent with user

* Corresponding author. Tel.: +1 8058868154.
E-mail address: laos1984@gmail.com (S. Tan).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.08.034

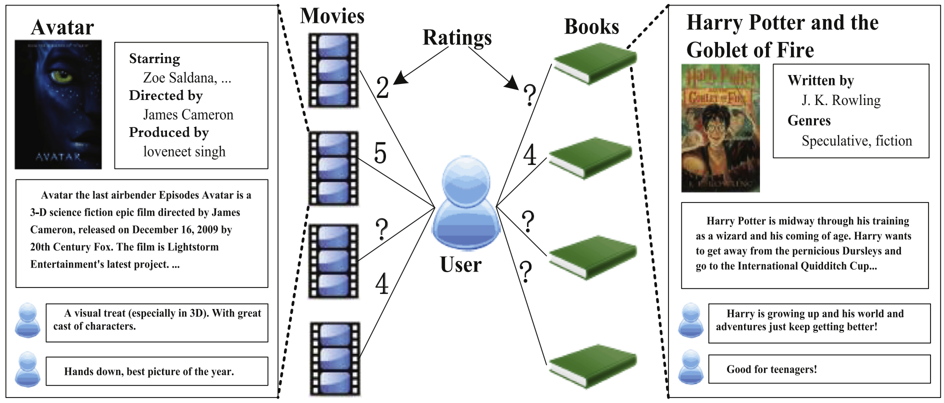


Fig. 1. An example for transferring user interests cross domains. We consider two domains here, movies and books.

interests, especially for objects in the form of different media types.

In this paper, we propose a Bayesian hierarchical approach based on Latent Dirichlet Allocation (LDA) for cross domain recommendation. We model documents (we consider all types of objects, such as movies and books, as documents) from different domains and user interests in a common topic space. The topic distributions for documents and user interests are learned simultaneously and the topic distribution of a particular document is built on document content as well as user interests. Then the correlations among different types of media can be constructed based on the topic distributions. Since we incorporate user interests in topic modeling, the correlations are forced to agree with user tastes. Specifically, a document corresponds to a media object and includes three parts of information: media content (or its description text), user-generated text data and ratings. We model media content in a similar way as the basic LDA model. But for the user-generated text, we choose topics either from the document topic or the user interest topic, because user generated text, such as tags, is related to both the document and the user interests. Finally, we model ratings based on the assumption: a user will like a document if there are common topics between the user interests and the document. Given two topics chosen from the document and the user respectively, a rating is drawn from a rating distribution. Different from the word distribution in basic LDA model which is in the form of a two order matrix, the rating distribution here is a three order tensor. Based on this model, we can suggest documents to users based on predicting ratings or by comparing topic distributions between documents and user interests directly. Experiments on two real world datasets demonstrate that our proposed method outperforms baseline methods in cross domain recommendation.

The rest of this paper is organized as follows. Section 2 reviews the related work. In Section 3, we represent the data used in our method and define notations. In Section 4, we introduce our proposed method on cross domain recommendation. Extensive experimental results are presented in Section 5. We conclude our paper in Section 6.

2. Related work

In this paper, we propose to transfer user interests for cross media recommendation based on LDA model. Our work is related to recommendation by transfer learning, cross media retrieval and recommendation by LDA model. In this section we provide a brief review of these works.

2.1. Cross domain recommendation

Recently, some researchers introduce transfer learning methods for cross domain recommendation. Phuong et al. propose to apply multi-task learning (a kind of transfer learning methods [5]) to collaborative filtering [6]. This work treats each user rating prediction as a classification problem and solves multiple classification problems together. It does not consider any auxiliary data and is very different from the problem we focus on. To utilize information in related relations or networks, [7,8] consider heterogeneous relations as auxiliary information sources and introduce methods based on the Collective Matrix Factorization framework. However, all the information used in their methods, either the target relation or auxiliary relations, is from a common domain, which is different from our purpose of transferring user interests cross media. As mentioned in Section 1, Li et al. consider the cluster-level rating patterns as potential candidates to be transferred from the auxiliary domain [3,4]. These methods require that the rating matrix in auxiliary data is dense which is often impossible in practice. Besides, Pan et al. try to discover the principle coordinates of both users and items in the auxiliary data matrices, then use these coordinates to adapt to the target data [2]. Zhang et al. propose to use probabilistic matrix factorization to model the ratings in each domain simultaneously. In this way, knowledge can be transferred cross domains adaptively [9]. Cao et al. introduce a nonparametric Bayesian framework to transfer knowledge for cross domain recommendation [10]. Recently, Pan et al. proposed to utilize unstandardized rating information in auxiliary domains: such as binary ratings, rating ranges and rating distributions [11,12]. These methods focus on similar problems as ours, but they only involve ratings and ignore other kinds of useful information. Our proposed method is a more flexible one which can exploit various types of information, ratings, media content, user-generated text and so on.

Besides, Tang et al. propose a novel generative model to recommend research collaborators in different domains to facilitate cross domain collaborations [13]. This work is similar to the friend recommendation task in social networks but for "friends" in different research domains, which is very different from our task. We focus on resource recommendation in this paper.

2.2. Cross media retrieval

Cross media retrieval is committed to searching and processing documents containing different types of media in an integrated way. The key problem of cross media retrieval is to find correlations between the considered media types. There are already some outstanding papers in this direction. Jeon et al. propose to annotate and retrieve images automatically by cross media relevance models. Specifically, they assume that regions in an image can be described using a small vocabulary of blobs. Then they use probabilistic models to predict the probability of generating a word given the blobs in an image [14]. Pan et al. develop a graph based method to get correlations cross media based on given similarity functions for each type of media [15]. To annotate images, Monday et al. present three algorithms based on the Probabilistic Latent Semantic Analysis (PLSA) model. Under the PLSA assumptions, an image is modeled as a mixture of latent aspects that generates both image features and text captions. Then this paper introduces three ways to learn the mixture of aspects [16]. Recently, Yang et al. present a framework for cross-media retrieval, where the query example and the retrieved result(s) can be of different media types. They first explore the semantic correlations by utilizing media content and co-occurrence information. Then they propose a new ranking algorithm, namely ranking with Local Regression and Global Alignment, to rank the multimedia data [17]. Rasiwasia et al. investigate the method to jointly model text and image components of multimedia documents. They first get features of text and images respectively, and then learn correlations between text and images by canonical correlation analysis [18]. Although these methods achieve successes in some applications, but none of them can be used for cross media recommendation.

2.3. Recommendation by LDA based models

In these years, numbers of research has been done in trying to model the collaborative filtering process using complex probabilistic models, such as PLSA and LDA [19–21]. Marlin proposes a generative latent variable model for rating-based collaborative filtering called the User Rating Profile model (URP), which is similar with the LDA model. This method produces complete user rating profiles, an assignment of one rating to each item for each user, and represents each user as a mixture of user attitudes [20]. Krestel et al. introduce a tag recommendation approach based on LDA. For a resource, they consider its complete tag set from all users as its word tokens and elicit latent topics based on these words [21]. Our proposed method is also based on LDA, however it is a unified framework which can model various types of information besides ratings and can be used for cross media recommendation.

Recently, Wang and Blei develop an algorithm for scientific article recommendation by combining traditional collaborative filtering and LDA [22]. Their method can utilize binary ratings and paper contents but cannot use multi-scale ratings and user generated text information as ours. Besides, their model is not proposed for cross domain recommendation.

3. Data representation and notions

As represented in Fig. 1, the data comes from multiple domains, relating to different media types. To simplify the description, we only consider the situation of two domains here (different media types or the same media type with different categories). The document collections (i.e., items) from these domains are denoted by D_1 and D_2. For a document collection D from a particular domain, $D=\{d_1,d_2,\ldots,d_{|D|}\}$, where d_i is the ith document in the collection. To simplify the notation, we represent the number of items in D as D too, in the following narrative. In a document d, there are three types of information, document descriptions, user-generated text data and ratings. Document descriptions include media content, associated text, some kinds of meta data and so on. The examples of user-generated text data are comments by users and bookmarks in the form of tags. Ratings are explicit user interest information. In this paper, we only consider the case with one type of document description data and one type of user-generated text data. Actually, our proposed model is a general one which can be extended easily to deal with multiple kinds of data mentioned above simultaneously. We use a collection of word tokens to represent description data in a document d, $\mathbf{w}_d=\{w_d^1,w_d^2,\ldots,w_d^N\}$, where N is the number of word tokens in this description. The word tokens here can be text words, image features and so on. The number of word tokens in the dictionary of description data is V_w. The user-generated text data (e.g., comments) is denoted as $\mathbf{v}_d=\{\mathbf{v}_{d,1},\mathbf{v}_{d,2},\ldots,\mathbf{v}_{d,U}\}$ and $\mathbf{v}_{d,u}=\{v_{d,u}^1,v_{d,u}^2,\ \ldots,v_{d,u}^M\}$, where U is the number of users who leave information in this document and M is the number of word tokens in the entry of user-generated data. The number of word tokens in the dictionary of user-generated data is V_v. The rating set for a document d is denoted as $\mathbf{r}_d=\{r_{d,1},r_{d,2},\ldots,r_{d,U}\}$, where U is the number of users who leave ratings in this document. The dictionary of rating values is $\{1,2,\ldots,V_r\}$ (e.g, { 1,…,5}). We summarize notations for our method in Table 1.

4. Transfer user interests cross media

In this section, we introduce the proposed *user interests considered cross media LDA model (cmLDA)*. Before defining the model, we present the basic LDA model and some of its extensions first. Based on the model definition, we then introduce the inference method and parameter estimation by Gibbs Sampling. Recommendation by our model is discussed in the end.

4.1. LDA model and its extensions

The proposed topic model in this paper is based on Latent Dirichlet Allocation (LDA) [23] which is a powerful generative model for collections of discrete data, such as text words. LDA aims at discovering the topics that generate the documents in a corpus, as well as the topic proportion for each document. As shown in Fig. 2(a), the topic proportion θ_d for a particular document d follows a Dirichlet distribution with parameter α. Given θ_d, a

Table 1
Notations for our proposed method.

Symbols	Descriptions
D_1, D_2	A document collection in each domain
$\mathbf{w}_d$, $\mathbf{v}_d$, $\mathbf{r}_d$	The document description data, the user-generated text data and the set of ratings of document d
$\mathbf{v}_{d,u}$	The user-generated text data of user u in d
N, M	The number of word tokens in $\mathbf{w}_d$ or $\mathbf{v}_{d,u}$
U, K	The number of users and topics considered
V_w,V_v, V_r	The number of word tokens in dictionaries
w_d^i, $v_{d,u}^j$, $r_{d,u}$	A word token or a rating
θ_d, μ_u	The topic distribution of document d or user u
$\lambda_{d,u}$	The topic type decision distribution for user u's generated data in d
z_d^i, $\mathbf{z}$	A topic association or the association set for document description data
$y_{d,u}^j$, $\mathbf{y}$	A topic type association or the association set for user generated text data
$s_{d,u}^j$, $\mathbf{s}$, $\mathbf{s}'$, $\mathbf{s}''$	A topic association or association sets for user generated text data, $\mathbf{s}'$ and $\mathbf{s}''$ are topic association sets with document topics or user topics, $\mathbf{s}=\{\mathbf{s}',\mathbf{s}''\}$
$b_{d,u}$, $\mathbf{b}$	A document topic association or the set for rating data
$c_{d,u}$, $\mathbf{c}$	A user topic association or the set for rating data
$\phi_{w;k}$, $\phi_{v;k}$, $\phi_{r;k_1,k_2}$	The word distribution or rating value distribution of given topics
α, γ, β	The fixed parameters of symmetric Dirichlet priors

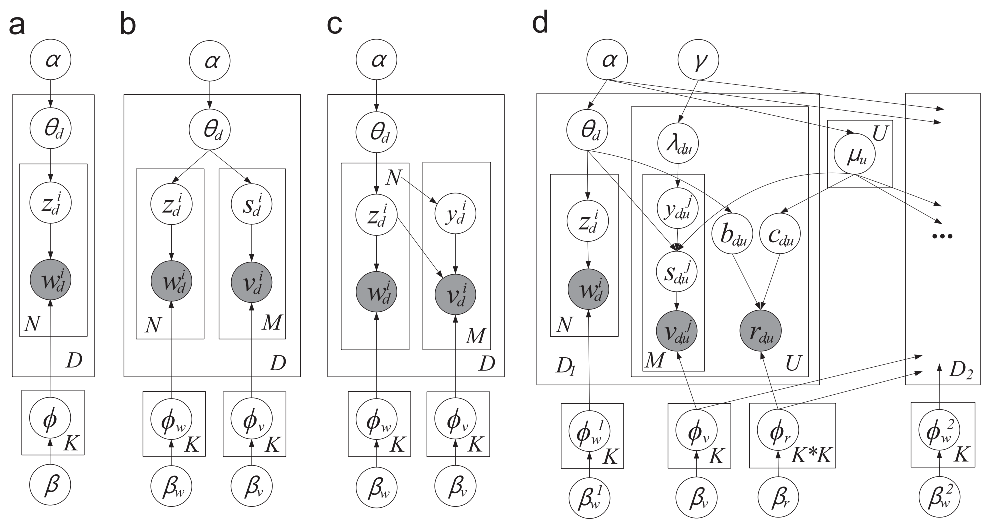


Fig. 2. (a) The basic LDA model; (b) the multi-modal LDA (mmLDA); (c) the correspondence LDA (cLDA); (d) the proposed cmLDA model.

particular topic z_d^i is drawn from a multinomial distribution. In turn, a word w_d^i is from a multinomial distribution ϕ with Dirichlet prior β conditioned on the topic z_d^i. ϕ is a $K \times V$ (K is the number of topics considered in the model and V is the dimension of the vocabulary) matrix each row of which denotes the word distribution of a topic. Different from Probabilistic Latent Semantic Indexing (PLSI) model [24], LDA assumes that each word in both the observed and unseen documents is generated by a randomly chosen topic. This advantage can handle some problems in recommendation, such as the new item cold start problem [1].

In order to extend the basic LDA model to describe the joint distributions of image and caption words in multimedia documents, some authors propose new methods based on LDA. These methods assume the existence of a small set of shared latent variables that are the common causes of correlations between the two modalities (i.e., image and caption). The most popular methods are multi-modal LDA (mmLDA) [25] and correspondence LDA (cLDA) [26]. The two extensions differ in choices of latent variables being shared between images and texts. By assuming the topic proportion variable θ is the common factor which generates both types of word tokens, mmLDA forces the topic distribution to be the same for image and caption modalities as seen in Fig. 2(b). Different from mmLDA, cLDA assumes that only a subset of topics that generate image features are used in generating the caption words. In cLDA, the image modality is designed as the primary one and generated first. Conditioned on the topics used in image, caption word are then generated, as represented in Fig. 2 (c). Since there are no dominant measures which work well for all data, we simply draw the idea of mmLDA for our model to handle multi-modal data in a document (e.g., document description data and user-generated data). As can be seen, all these extensions focus on multi-modal data in the same domain. It is very different from our model which is committed to explore correlations among documents from different domains and transfer user interests cross domain or media.

4.2. *User interests considered cross media LDA model (cmLDA)*

To transfer user interests for cross media recommendation, in this paper we design an *user interests considered cross media LDA model* (*cmLDA*). We first assume that documents (corresponding to media objects) from different domains and users in the online community share a common topic space. This assumption is very realistic for general popular online communities since most of items in these communities are literary and entertainment works [4]. In this topic space, documents are represented as various "categories", for example, the movie *Avatar* may belong to categories about "science fiction" or "3D movies". While users are characterized by interest topics, such as a user likes science fiction movies or books. Designing with this assumption, we force documents from different domains and user interests to be modeled in the same topic space. As can be seen in Fig. 2(d), we model topic distributions θ for documents from different domains and the topic distributions μ for user interests to be drawn from the same Dirichlet distribution with the parameter α. Because we embed documents with different media types in a common topic space (i.e., with the same mean topic proportion), the cross media correlations can be defined by comparing their topic distributions. The correlations are built by considering not only document content, but also user interests. For example, if two documents have similar content and/or accessed by common user groups, we believe that they are in similar topics or closely related. In this way, we can transfer user interests cross media.

In a particular document, we consider multi-modal data as presented in Fig. 2(d). Because multi-modal data is very common in online social communities as mentioned in Section 1. Formally, we use three types of information here: document description data (e.g., media features or description text), user-generated text data (e.g., comments or tags) and ratings. For each word token in document description data, given θ_d, a particular topic z_d^i is drawn from a multinomial distribution which is the same with the basic LDA model. But for user-generated text data, we generate each entry of them from two topic distributions, a document topic distribution θ_d and a user interest topic distribution μ_u. Since user-generated data contains not only document topic information but also user interests. Specifically, for each word $v_{d,u}^j$ in the short text (e.g., a comment), a binary association type $y_{d,u}^j$ drawn from a type decision distribution $\lambda_{d,u}$, is required for controlling whether to choose the topic $s_{d,u}^j$ from θ_d or μ_u. If $y_{d,u}^j = 0$, the topic of the word will be drawn from the document topic from θ_d. Otherwise ($y_{d,u}^j = 1$), the topic of the word will be drawn from the user interest topic from μ_u. For each rating $r_{d,u}$, we first draw a document topic $b_{d,u}$ from θ_d and a user interest topic $c_{d,u}$ from μ_u. Then different from document description data and user-generated text data which generate an observed word from a $K \times V$ word distribution matrix ϕ_w or ϕ_v, we generate a rating from a $K \times K \times V$ rating distribution tensor ϕ_r. By modeling these three types of information, document topic distributions are learned considering user interests as mentioned above.

To summarize, we have the following data generation process for cmLDA model:

1. Choose a topic distribution $\theta_d \sim Dirichlet(\alpha)$, for each document d in each domain.
2. Choose an interest topic distribution $\mu_u \sim Dirichlet(\alpha)$, for each user u.
3. Choose a word distribution $\phi^j_{w;k} \sim Dirichlet(\beta^j_w)$ for each topic k for document description in each domain, $j \in \{1,2\}$.
4. Choose a word distribution $\phi_{v;k} \sim Dirichlet(\beta_v)$ for each topic k in user-generated text data.
5. Choose a rating distribution $\phi_{r;k_1,k_2} \sim Dirichlet(\beta_r)$ for topics k_1 from a document and k_2 from a user in rating data (ϕ_r is a three order tensor).
6. For each word $w_d{}^i$ in document description data, $i \in \{1,\ldots,N\}$ and $d \in \{1,\ldots,D_1+D_2\}$:
   (a) Choose a topic $z^i_d \sim Multinomial(\theta_d)$.
   (b) Choose a word $w^i_d \sim Multinomial(\phi^j_{w;z^i_d})$, $j \in \{1,2\}$.
7. For each user u's self-generated text data in each document d, $u \in \{1,\ldots,U\}$ and $d \in \{1,\ldots,D_1+D_2\}$:
   (a) Choose a type decision distribution $\lambda_{d,u} \sim Dirichlet(\gamma)$ for u's self-generated text data.
   (b) For each word $v^j_{d,u}$ in u's self-generated text data, $j \in \{1,\ldots,M\}$:
      i. Choose a type $y^j_{d,u} \sim Bernoulli(\lambda_{d,u})$.
      ii. If $y^j_{d,u}=0$, choose a topic $s^j_{d,u} \sim Multinomial(\theta_d)$.
      iii. Else (i.e., $y^j_{d,u}=1$), choose a topic $s^j_{d,u} \sim Multinomial(\mu_u)$.
      iv. Choose a word $v^j_{d,u} \sim Multinomial(\phi_{v;s^j_{d,u}})$.
8. For each rating $r_{d,u}$, $u \in \{1,\ldots,U\}$ and $d \in \{1,\ldots,D_1+D_2\}$:
   (a) Choose a topic $b_{d,u} \sim Multinomial(\theta_d)$.
   (b) Choose a topic $c_{d,u} \sim Multinomial(\mu_u)$.
   (c) Choose a rating $r_{d,u} \sim Multinomial(\phi_{r;b_{d,u},c_{d,u}})$.

Given the hyperparameters α, γ, $\beta_w{}^1$, $\beta_w{}^2$, β_v and β_r, the joint distribution of observed variables in cmLDA is obtained by

$$\begin{aligned}\mathcal{L} &= P(\mathbf{y},\mathbf{z},\mathbf{s},\mathbf{b},\mathbf{c},\mathbf{w},\mathbf{v},\mathbf{r}|\alpha,\gamma,\beta^1_w,\beta^2_w,\beta_v,\beta_r)\\ &= P(\mathbf{y}|\gamma)P(\mathbf{z}_1,\mathbf{z}_2,\mathbf{s}',\mathbf{b}|\mathbf{y}=0;\alpha)P(\mathbf{s}'',\mathbf{c}|\mathbf{y}=1;\alpha)P(\mathbf{w}_1|\mathbf{z}_1;\beta^1_w)\\ &\quad P(\mathbf{w}_2|\mathbf{z}_2;\beta^2_w)P(\mathbf{v}|\mathbf{s}',\mathbf{s}'';\beta_v)P(\mathbf{r}|\mathbf{b},\mathbf{c};\beta_r). \end{aligned} \tag{1}$$

Note that we consider the most complex situation here: document description data in two domains are in two different types and neither in text words (e.g., in image features and acoustic features). In the most simple case, all document description data in both domains are in text words, which means we use text to describe objects (i.e., documents) instead of their media content. In this easier situation, we only need two βs (one for text and one for ratings) but not four like in Fig. 2(d).

4.3. Gibbs sampling

Similar to the basic LDA model, exact inference for our model is intractable. There are some approximate inference methods, such as variational inference [23], expectation propagation [27] and Gibbs Sampling [28,29]. We use Gibbs Sampling here, since it has been proved effective as avoiding local optima and easy to extend. Gibbs Sampling is also called collapsed Gibbs Sampling, which means θs, λs, μs and ϕs will be integrated out. Specifically, we sample a topic assignment for each word or rating based on the observations and topic assignments for other words and ratings. The details about Gibbs sampling for LDA-like probabilistic models can be found in [7].

The sampling methods of $\mathbf{z}_1$ and $\mathbf{z}_2$ (corresponding to *domain 1* and *domain 2*) for document description data are as follows:

$$\begin{aligned}&P(z^i_{1;d}=k|\mathbf{z}_1^{-(d,i)},\mathbf{y},\mathbf{z}_2,\mathbf{s},\mathbf{b},\mathbf{c},\mathbf{w},\mathbf{v},\mathbf{r};\alpha,\gamma,\beta^1_w,\beta^2_w,\beta_v,\beta_r)\\ &\quad\times\propto \frac{\Omega^{1,k}_{(\cdot),v}+\beta^1_w-1}{\Omega^{1,k}_{(\cdot),(\cdot)}+V^1_w\cdot\beta^1_w-1}\times(\Omega^{1,k}_{d,(\cdot)}+\Phi'^{k}_{d,(\cdot)}+\Psi^{k,(\cdot)}_{d,(\cdot),(\cdot)}+\alpha-1),\end{aligned} \tag{2}$$

$$\begin{aligned}&P(z^i_{2;d}=k|\mathbf{z}_2^{-(d,i)},\mathbf{y},\mathbf{z}_1,\mathbf{s},\mathbf{b},\mathbf{c},\mathbf{w},\mathbf{v},\mathbf{r};\alpha,\gamma,\beta^1_w,\beta^2_w,\beta_v,\beta_r)\\ &\quad\times\propto \frac{\Omega^{2,k}_{(\cdot),v}+\beta^2_w-1}{\Omega^{2,k}_{(\cdot),(\cdot)}+V^2_w\cdot\beta^2_w-1}\times(\Omega^{2,k}_{d,(\cdot)}+\Phi'^{k}_{d,(\cdot)}+\Psi^{k,(\cdot)}_{d,(\cdot),(\cdot)}+\alpha-1),\end{aligned} \tag{3}$$

where v is the token in the vocabulary that has the same symbol with token i. $\Omega^{1,k}_{d,v}$ is the number of word tokens in the dth document (a document in the domain 1) with the same word symbol (the vth word in the vocabulary) assigned to the kth topic. $\Omega^{2,k}_{d,v}$ is similar but for the domain 2. $\Phi'^{k}_{d,v}$ is the number of word tokens in the user generated text data of dth document with the vth word assigned to the kth topic. Moreover the topic k is drawn from the document topic distribution θ but not the user topic distribution μ. Conversely, $\Phi''^{k}_{u,v}$ which will be used below, is for the uth user's generated text data and k is drawn from the user topic distribution μ but not the document topic distribution θ. $\Psi^{k_1,k_2}_{d,u,v}$ is the number of rating value tokens from user u to document d with the rating value v assigned to the k_1th document topic from θ and the k_2th user topic from μ. $\Psi^{k_1,k_2}_{d,u,v}=0$ or 1.

The sampling process of $\mathbf{s}$ for user-generated data is different from the process of $\mathbf{z}$. Each topic assignment in $\mathbf{s}$ is sampled from $2K$ topics (K topics from θ and K topics from μ) rather than K topics.

$$\begin{aligned}&P(s^j_{d,u}=k,y^j_{d,u}=0\Big|\mathbf{s}^{-(d,u,j)},\mathbf{y}^{-(d,u,j)},\mathbf{z}_1,\mathbf{z}_2,\mathbf{b},\mathbf{c},\mathbf{w},\mathbf{v},\mathbf{r};\alpha,\gamma,\beta^1_w,\beta^2_w,\beta_v,\beta_r)\\ &\quad\times\propto\frac{\Phi'^{k}_{(\cdot),v}+\Phi''^{k}_{(\cdot),v}+\beta_v-1}{\Phi'^{k}_{(\cdot),(\cdot)}+\Phi''^{k}_{(\cdot),(\cdot)}+V_v\cdot\beta_v-1}\times\frac{\Omega^{(\cdot),k}_{d,(\cdot)}+\Phi'^{k}_{d,(\cdot)}+\Psi^{k,(\cdot)}_{d,(\cdot),(\cdot)}+\alpha-1}{\Omega^{(\cdot),(\cdot)}_{d,(\cdot)}+\Phi'^{(\cdot)}_{d,(\cdot)}+\Psi^{(\cdot),(\cdot)}_{d,(\cdot),(\cdot)}+K\cdot\alpha-1}\\ &\quad\times(\Delta_{d,u,0}+\gamma-1),\end{aligned} \tag{4}$$

$$\begin{aligned}&P(s^j_{d,u}=k,y^j_{d,u}=1\Big|\mathbf{s}^{-(d,u,j)},\mathbf{y}^{-(d,u,j)},\mathbf{z}_1,\mathbf{z}_2,\mathbf{b},\mathbf{c},\mathbf{w},\mathbf{v},\mathbf{r};\alpha,\gamma,\beta^1_w,\beta^2_w,\beta_v,\beta_r)\\ &\quad\times\propto\frac{\Phi'^{k}_{(\cdot),v}+\Phi''^{k}_{(\cdot),v}+\beta_v-1}{\Phi'^{k}_{(\cdot),(\cdot)}+\Phi''^{k}_{(\cdot),(\cdot)}+V_v\cdot\beta_v-1}\times\frac{\Phi''^{k}_{u,(\cdot)}+\Psi^{(\cdot),k}_{(\cdot),u,(\cdot)}+\alpha-1}{\Phi''^{(\cdot)}_{u,(\cdot)}+\Psi^{(\cdot),(\cdot)}_{(\cdot),u,(\cdot)}+K\cdot\alpha-1}\\ &\quad\times(\Delta_{d,u,1}+\gamma-1),\end{aligned} \tag{5}$$

where $\Delta_{d,u,0}$ is the number of word tokens in the entry of user generated text data (user u's text data in document d) which are drawn from document topics θ_d, and $\Delta_{d,u,1}$ is the number of word tokens which are drawn from user topics μ_u

Finally, $\mathbf{b}$ and $\mathbf{c}$ are sampled from $K*K$ variables as follows:

$$\begin{aligned}&P(b_{d,u}=k_1,c_{d,u}=k_2\Big|\mathbf{b}^{-(d,u)},\mathbf{c}^{-(d,u)},\mathbf{z}_1,\mathbf{z}_2,\mathbf{s},\mathbf{y},\mathbf{w},\mathbf{v},\mathbf{r};\alpha,\gamma,\beta^1_w,\beta^2_w,\beta_v,\beta_r)\\ &\quad\times\propto\frac{\Psi^{k_1,k_2}_{(\cdot),(\cdot),v}+\beta_r-1}{\Psi^{k_1,k_2}_{(\cdot),(\cdot),(\cdot)}+V_r\cdot\beta_r-1}\times(\Omega^{(\cdot),k_1}_{d,(\cdot)}+\Phi'^{k_1}_{d,(\cdot)}+\Psi^{k_1,(\cdot)}_{d,(\cdot),(\cdot)}+\alpha-1)\\ &\quad\times(\Phi''^{k_2}_{u,(\cdot)}+\Psi^{(\cdot),k_2}_{(\cdot),u,(\cdot)}+\alpha-1).\end{aligned} \tag{6}$$

Note that our model can receive multi-scale ratings, such as the 5-star ratings. Different rating values have different meanings, such as 1–5 stars may mean “Poor”, “Below Average”, “Average”, “Above Average” and “Excellent” respectively. If a user rates an item 5 stars, we could reasonably conclude that they may be in similar topics. On the contrary, 1 star may lead us believe that their topics are very different. To make the model “understand” meanings of different rating values, we force the above sampling process (i.e., Eq. (6)) to repeat until $b_{d,u}=c_{d,u}$ for ratings bigger than the median (e.g., 3) and $b_{d,u}\neq c_{d,u}$ for ratings smaller than the median, in each iteration. For ratings equal to the median, the sampling process is performed only once in each iteration.

4.4. Parameter estimation

Given the sampled topics **z**, **y**, **s**, **b** **c**, as well as the inputs: α, β_w, β_v, β_r, **w**, **w** and **r**, we can estimate model parameters λs, θs, μs and ϕs:

$$\lambda_{d,u,0} = \frac{\Delta_{d,u,0}+\gamma}{\Delta_{d,u,(\cdot)}+2\cdot\gamma}; \lambda_{d,u,1} = \frac{\Delta_{d,u,1}+\gamma}{\Delta_{d,u,(\cdot)}+2\cdot\gamma}, \tag{7}$$

$$\theta_{d,k} = \frac{\Omega_{d,(\cdot)}^{(\cdot),k}+\Phi_{d,(\cdot)}^{\prime k}+\Psi_{d,(\cdot),(\cdot)}^{k,(\cdot)}+\alpha}{\Omega_{d,(\cdot)}^{(\cdot),(\cdot)}+\Phi_{d,(\cdot)}^{\prime(\cdot)}+\Psi_{d,(\cdot),(\cdot)}^{(\cdot),(\cdot)}+K\cdot\alpha}, \tag{8}$$

$$\mu_{u,k} = \frac{\Phi_{u,(\cdot)}^{\prime\prime k}+\Psi_{(\cdot),u,(\cdot)}^{(\cdot),k}+\alpha}{\Phi_{u,(\cdot)}^{\prime\prime(\cdot)}+\Psi_{(\cdot),u,(\cdot)}^{(\cdot),(\cdot)}+K\cdot\alpha}, \tag{9}$$

$$\phi_{w;k,v}^1 = \frac{\Omega_{(\cdot),v}^{1,k}+\beta_w^1}{\Omega_{(\cdot),(\cdot)}^{1,k}+V_w^1\cdot\beta_w^1}; \quad \phi_{w;k,v}^2 = \frac{\Omega_{(\cdot),v}^{2,k}+\beta_w^2}{\Omega_{(\cdot),(\cdot)}^{2,k}+V_w^2\cdot\beta_w^2}, \tag{10}$$

$$\phi_{v;k,v} = \frac{\Phi_{(\cdot),v}^{\prime k}+\Phi_{(\cdot),v}^{\prime\prime k}+\beta_v}{\Phi_{(\cdot),(\cdot)}^{\prime k}+\Phi_{(\cdot),(\cdot)}^{\prime\prime k}+V_v\cdot\beta_v}, \tag{11}$$

$$\phi_{r;k_1,k_2,v} = \frac{\Psi_{(\cdot),(\cdot),v}^{k_1,k_2}+\beta_r}{\Psi_{(\cdot),(\cdot),(\cdot)}^{k_1,k_2}+V_r\cdot\beta_r}. \tag{12}$$

4.5. Recommendation by cmLDA

In this paper, we focus on the top-N recommendation task and consider recommendation as a ranking problem. Since the top-N recommendation task is more practical in real commercial systems, than the rating prediction task [30]. With our proposed model, the top-N recommendation task can be done in following three ways:

1. Based on θs, μs and ϕ_r, predict the probability of getting each rating for each user u and each document d: $P(r_{d,u} = i|\mu_u, \theta_d, \phi_r)$.

$$P(r_{d,u} = i|\mu_u, \theta_d, \phi_r) = \frac{\sum_{k_1=1}^{K}\sum_{k_2=1}^{K}\theta_{d,k_1}\mu_{u,k_2}\phi_{r;k_1,k_2,i}}{\sum_{j=1}^{V_r}\sum_{k_1=1}^{K}\sum_{k_2=1}^{K}\theta_{d,k_1}\mu_{u,k_2}\phi_{r;k_1,k_2,j}}. \tag{13}$$

   Then for each user, rank items by their probability summaries over ratings bigger than a rating threshold $\hat{r}$ (e.g., the average of all training ratings):

$$\tilde{P}(d,u) = \sum_{i>\hat{r}} P(r_{d,u} = i|\mu_u, \theta_d, \phi_r). \tag{14}$$

   We denote this method as *cmLDA-ProbRate* in short.
2. Based on Eq. (13), rank items by predicting their rating values, which is denoted as *cmLDA-PredRate*:

$$\tilde{r}(d,u) = \sum_{i=1}^{V_r} P(r_{d,u} = i|\mu_u, \theta_d, \phi_r)\times i. \tag{15}$$

3. Recommendation based on comparing topic distributions between documents and user interests by the cosine similarity (denoted as *cmLDA-TopicSim*).

$$sim(d,u) = \frac{\sum_{k=1}^{K}\theta_{d,k}\times\mu_{u,k}}{\sqrt{\sum_{k=1}^{K}(\theta_{d,k})^2}\times\sqrt{\sum_{k=1}^{K}(\mu_{u,k})^2}}. \tag{16}$$

Keep in mind that we have two set of documents from two different domains, even in the form of different types of media. To transfer user interests and recommend documents cross media, we learn the parameters for all documents and all user interests simultaneously. For a new document from either domain, we can build its topic distribution θ based on ϕ_w. This method can be used for solving the new item cold start problem.

Besides, to simplify the description, we only consider two different domains above. Actually, our proposed algorithm is a general framework which can be extended easily for multiple domains. In experiments, we will use datasets with more than two domains. If one domain is considered as the target domain and all other domains will be used as auxiliary domains.

5. Experiments

In this section, we investigate the use of our proposed algorithm for cross media recommendation. Before presenting the experiment results, we introduce the experiment settings first.

5.1. Datasets

To evaluate our algorithm, we downloaded two real work datasets. The first dataset is about movies in different categories from MovieLens.[1] This dataset was collected by GroupLens[2] and there are 19 movies categories in the original set. We choose five of these categories with most ratings for our experiments: *Drama*, *Comedy*, *Thriller*, *Romance* and *Action*. Except ratings, there are some bookmarks for each movie in this dataset. Since there is no movie description information in the data, we downloaded movie description text from *Wikipedia*.[3]

The second dataset is collected from Amazon.com[4] used in [31], which contains 6.7 million products. We use a subset of this data which corresponds to three commodity types due to their most popularity: *movie DVD*, *music CD* and *book*. For each movie DVD, music CD or book, we collected its description text, user-generated comments and ratings.

Tables 2 and 3 show some basic statistics about our datasets. All users in these datasets have ratings in all domains. Note that these two datasets are very different in rating ratio (i.e., sparsity). The performance stability of our algorithm can be evaluated on these datasets.

5.2. Data processing and experiment settings

Since it is difficult to get media content data, such as movies themselves, we use the associated text as the description data instead of the media content. For the MovieLens dataset, we use movie summary text and movie meta data (e.g., the director and actors) as the description data, and treat bookmarks in form of tags as the user-generated text data. For the Amazon dataset, we combine the commodity summary text as the description data, and treat comments as the user-generated text data.

[1] http://www.movielens.org.
[2] http://www.grouplens.org.
[3] http://www.wikipedia.org.
[4] http://www.amazon.com.

Table 2
Dataset from Movielens.

Movie type	*Drama*	*Comedy*	*Thriller*	*Romance*	*Action*	Total
No. items	2398	2318	1500	1348	1112	8676
No. users	3267	3267	3267	3267	3267	3267
No. ratings	4,76,202	6,54,569	4,56,802	5,20,118	4,87,085	2,594,776
Rating ratio (%)	6.1	8.6	9.3	11.8	13.4	9.2

Table 3
Dataset from Amazon.

Item type	*Movie DVD*	*Music CD*	*Book*	Total
No. items	1918	1596	1526	5040
No. users	1844	1844	1844	1844
No. ratings	39,216	23,668	12,144	75,028
Rating ratio (%)	1.11	0.80	0.43	0.81

To build the dictionaries, we first remove all stop words from text data and all words appearing less than five times. To improve the quality of the dictionary, we then use the *Porter Stemmer* algorithm to reduce each word to its basic root or stem (e.g., *books* to *book*). Note that we keep multiple dictionaries (for description data in each domain and user-generated data) in our experiments through all description data and user-generated data are in text words.

The relationship between parameters α, βs and the topic number K is a mutual one. In experiments of this paper, we fix these parameters as: $\alpha=2$, $\gamma=0.5$, $\beta_w=\beta_v=0.01$, $\beta_r=0.1$ and $K=20$ heuristically. The details about parameter settings for LDA model can be found in [28].

5.3. Compared methods

We compare our algorithm with four other methods. The first one is an user-based Collaborative Filtering (**CF**) method [32]. Given a target user u, let $\hat{r}_{u,d}$ be a predicted ranking score of user u for document d:

$$\tilde{r}_{u,d}=\overline{r}_u+\frac{\sum_{j=1}^{k}(r_{j,d}-\overline{r}_j)s_{u,j}}{\sum_{j=1}^{k}s_{u,j}}, \tag{17}$$

where $\overline{r}_u$ is the mean ratings of user u, $s_{u,j}$ is the similarity weight between users u and j. k is the number of nearest neighbors of user u. We employ the cosine-based approach [33] to compute the similarities between users. Based on the obtained similarities, we use the significance weighting method proposed in [34] to improve the recommendation performance. Specifically, if the number of co-used items between two users, denoted by n, is less than a threshold number N, then we multiply their similarity by n/N. In our experiment, we tune the value of N and the number of nearest neighbors k to achieve the best performance. Since all domains share common users, we combine all items in different domains (auxiliary domains and the target domain) together and input ratings on all items to CF.

The second compared methods is based on a the-state-of-the-art model-based recommendation algorithm: **NPCA** proposed in [35]. Similar to the above baseline, we expand NPCA to utilize all ratings in different domains.

Our recommendation method exploits rating information and item content information. So we choose **CTR** in [22] as the third baseline which can also utilize ratings and text content. As mentioned in Section 2.3, CTR can only use binary ratings. To adjust our data to CTR, we preprocess multi-scale ratings as follows: If the rating value is not smaller than the median, it will be rewritten as 1. Otherwise, it will be ignored (i.e., rewritten as 0). Besides, CTR cannot use user generated text information (e.g., comments) naturally as our proposed model. We combine all user generated text on an item and its description text as an entire content document for this item.

The last baseline is a recommendation method based on transfer learning: **RMGM** introduced in [4], which is proposed for cross domain recommendation.

5.4. Evaluation

To evaluate the performance of our algorithm and other compared algorithms, we randomly select part of ratings (80%) in the target domain for training and the others for testing. We also remove user-generated text (comments and bookmarks) corresponding to ratings in the test data. Note that every domain in a dataset can be considered as the target domain and meanwhile the other domains are treated as its auxiliary domains. All information in auxiliary domains is used for training. For example, if the first domain (i.e., Drama) in the Movielens dataset is treated as the target domain, 80% information of this domain and all information in the other four domains will be used in the training step.

For evaluating the performance of recommendation, common metrics are error metrics, such as RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error). However most commercial systems only push top-N results to users and it was verified that error metrics do not really measure top-N recommendation performance [30]. Recently, researchers tend to choose ranking-based metrics to evaluate top-N performance directly [32,30,36]. So we use ranking-based metrics: Precision, Recall, Mean Average Precision (MAP) and Normalized Discount Cumulative Gain (NDCG) to measure the performance of different recommendation algorithms in this work. Precision is defined as the number of correctly recommended items divided by the total number of recommended items. Recall is defined as the number of correctly recommended items divided by the total number of items which should be recommended (i.e., those actually used by the target user). Average Precision (AP) is the average of precisions computed at the point of each correctly recommended item in the recommendation list:

$$\mathtt{AP}=\frac{\sum_i^N \mathtt{Precision}@i*\mathtt{corr}_i}{\mathtt{Number\ of\ correctly\ recommended\ items}}, \tag{18}$$

where Precision $@i$ is the precision at ranking position i, N is the number of recommended items ($N=20$ in this paper), and corr $_i=1$ if the item at position i is correctly recommended, otherwise corr $_i=0$. MAP is the mean of average precision scores over all users. NDCG at position n is defined as

$$\mathtt{NDCG}@n=\frac{1}{\mathtt{IDCG}}\times\sum_{i=1}^{n}\frac{2^{r_i}-1}{\log_2(i+1)}, \tag{19}$$

where r_i is the relevance rating of item at rank i. In our case, r_i is the rating value (e.g., 1, 2, …, 5). IDCG is chosen so that the perfect ranking has a NDCG value of 1.

5.5. Experiment results

We use all evaluation metrics mentioned in Section 5.4 to measure the performance of each recommendation algorithm on two datasets. Table 4 and Fig. 3(a)–(e) show MAP, Precision, Recall and NDCG on the Movielens dataset for all five algorithms. It is evident that our proposed algorithm significantly outperforms other recommendation methods in most cases, especially in top

Table 4
Comparison of recommendation algorithms in terms of MAP, Precision (P) and Recall (R) on the Movielens dataset. (Bold typeset indicates the best performance and statistical significant at $p < 0.001$ compared to the second best.).

Target domain	Method	MAP	P@5	P@15	R@5	R@15
Drama	CF	0.2159	0.1087	0.1208	0.0222	0.0734
	NPCA	0.3407	0.2059	0.1442	0.0388	0.0803
	CTR	0.3386	0.2260	0.2531	0.0472	0.1633
	RMGM	0.3166	0.1502	0.0833	0.0302	0.0470
	cmLDA	**0.4663**	**0.3505**	**0.2679**	**0.0777**	0.1623
Comedy	CF	0.2274	0.1235	0.1374	0.0183	0.0607
	NPCA	0.3434	0.2229	0.1625	0.0309	0.0657
	CTR	0.3705	0.2669	0.2702	0.0412	**0.1317**
	RMGM	0.2694	0.1552	0.0808	0.0235	0.0361
	cmLDA	**0.4555**	**0.3349**	**0.2805**	**0.0496**	0.1155
Thriller	CF	0.1914	0.0953	0.1066	0.0258	0.0863
	NPCA	0.2865	0.1601	0.1142	0.0398	0.0857
	CTR	0.3185	0.2084	0.2275	0.0601	0.1437
	RMGM	0.1400	0.0566	0.0519	0.0143	0.0379
	cmLDA	**0.4142**	**0.3063**	**0.2440**	**0.0711**	0.1412
Romance	CF	0.2364	0.1355	0.1484	0.0314	0.1053
	NPCA	0.3228	0.1963	0.1413	0.0425	0.0928
	CTR	0.3252	0.2277	0.2345	0.0538	0.1604
	RMGM	0.3161	0.1917	0.1235	0.0465	0.0880
	cmLDA	**0.5222**	**0.4442**	**0.3064**	**0.0791**	**0.1809**
Action	CF	0.3240	0.2248	0.2231	0.0440	0.1308
	NPCA	0.3912	0.2722	0.1983	0.0493	0.1059
	CTR	0.4055	0.2949	0.3242	0.0611	**0.2108**
	RMGM	0.3475	0.2367	0.1641	0.0456	0.0954
	cmLDA	**0.5347**	**0.4375**	0.3282	**0.0826**	0.1887

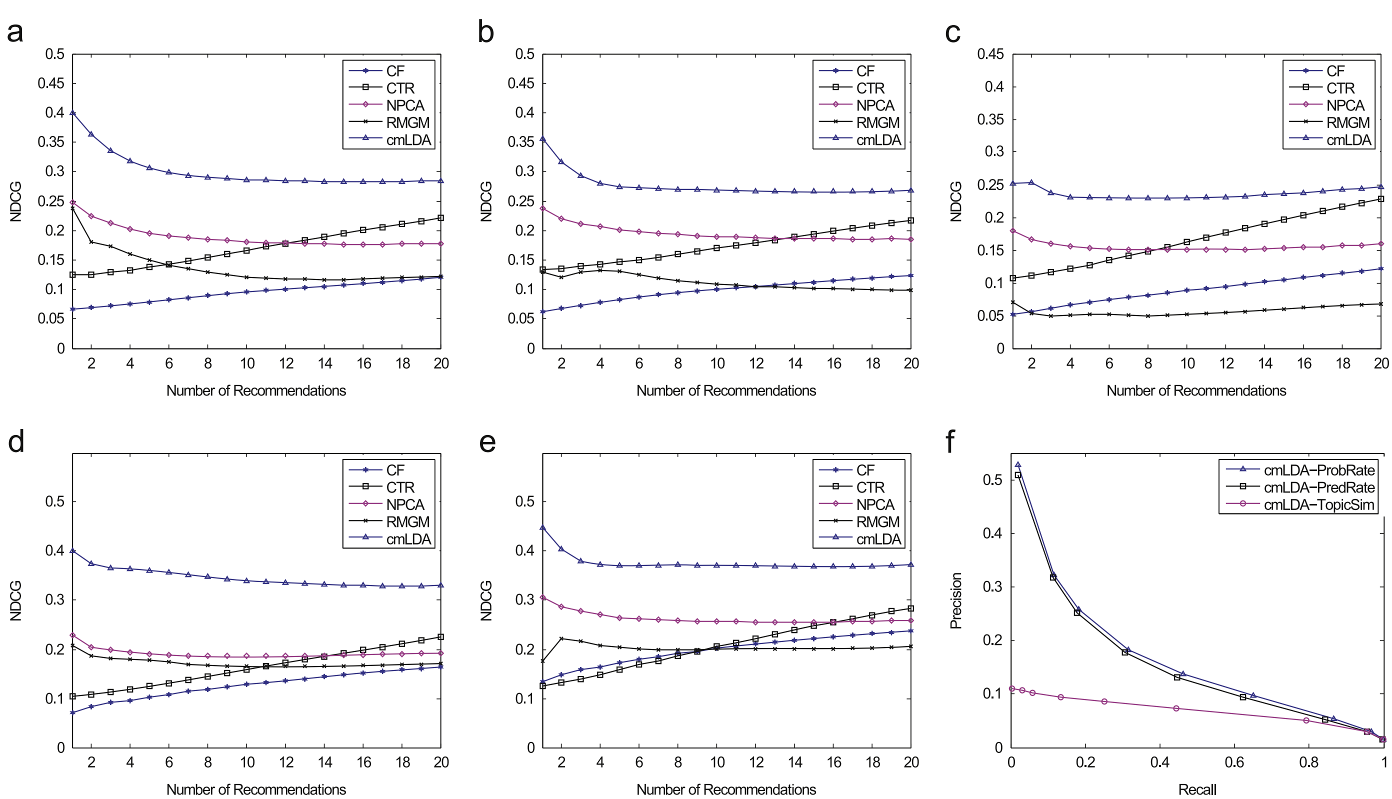


Fig. 3. (a)–(e) NDCG on five domains in the Movielens dataset, (f) Recall–Precision curves for three cmLDA based recommendation methods. It is the average over all five domains in the Movielens dataset. (a) NDCG on Drama, (b) NDCG on Comedy, (c) NDCG on Thriller, (d) NDCG on Romance, (e) NDCG on Action and (f) Recall-Precision curves.

rank positions. Compared with the three baselines (CF, RMGM and NPCA) which are only based on ratings, our method works better in every cases. The reason is that our method can exploit multi-type media information, such as item descriptions and user-generated text. These kinds of information are very useful in item topic learning and user profile building. The baseline CTR uses all information in the dataset, but it works worse than the proposed method in most cases, especially in top 10 rank positions. Top 5 or 10 recommendation results are more important in practical recommender systems since these systems generally only show 5 or 10 results to users. Reasons of the superiority of our method over CTR mainly lie in: (1) our method utilizes different kinds of text information in a reasonable way but CTR just combines these kinds of information as the item description text simply. (2) our method can use multi-scale ratings directly as mentioned in the end of Section 4.3, however CTR transfers multi-scale ratings to 0-1 relevance. Moreover, we find that through RMGM works well on the rating prediction error metrics (i.e., MAE and RMSE), it works badly in the ranking-based evaluation metrics (e.g., MAP and NDCG). This is consistent with previous studies: rating prediction error metrics do not really measure top-N ranking performance [30].

We also compare different recommendation methods based on cmLDA (mentioned in Section 4.5): cmLDA-ProbRate, cmLDA-PredRate and cmLDA-TopicSim on the Movielens dataset. Fig. 3(f) shows the recall-precision curves for these three methods. It is the average over all five domains. As can be seen, cmLDA-ProbRate and cmLDA-PredRate work well and similarly, but cmLDA-TopicSim works much worse. We use cmLDA-ProbRate in all other comparing experiments.

The corresponding results of the Amazon dataset are reported in Table 5 and Fig. 4. This dataset is highly sparse as shown in Table 3 and the rating ratio of this dataset is about one-tenth of that of the Movielens dataset. From Table 5 and Fig. 4, we can find that only our method and CF method output meaningful results. All other methods suffer from the sparsity problem and work badly. Moreover, our method works much better than the second best method CF. CF works better than other baselines in this case probably because it only uses local data (i.e., neighbors' ratings) but not global data when it computes the item list for each user. These results demonstrate the feasibility and stability of our proposed method on sparse data.

6. Conclusion

To alleviate the sparsity problem in recommender systems, we introduce a probabilistic collaborative filtering algorithm based on Latent Dirichlet Allocation model for cross domain or cross media recommendation. We first assume that documents (i.e., items) from different domains and user interests share a common topic space. Then topic distributions for documents and user interests

Table 5
Comparison of recommendation algorithms in terms of MAP, Precision (P) and Recall (R) on the Amazon dataset. (Bold typeset indicates the best performance and statistical significant at $p < 0.001$ compared to the second best.).

Target Domain	Method	MAP	P@5	P@15	R@5	R@15
Movie DVD	CF	0.0251	0.0079	0.0081	0.0078	0.0264
	NPCA	0.0095	0.0034	0.0036	0.0014	0.0071
	CTR	0.0087	0.0024	0.0031	0.0012	0.0101
	RMGM	0.0061	0.0015	0.0019	0.0016	0.0069
	cmLDA	**0.0532**	**0.0188**	**0.0141**	**0.0203**	**0.0397**
Music CD	CF	0.0313	0.0093	0.0098	0.0139	0.0469
	NPCA	0.0075	0.0024	0.0028	0.0033	0.0126
	CTR	0.0076	0.0024	0.0026	0.0031	0.0083
	RMGM	0.0065	0.0025	0.0021	0.0042	0.0097
	cmLDA	**0.0507**	**0.0193**	**0.0141**	**0.0267**	**0.0560**
Book	CF	0.0210	0.0060	0.0050	0.0173	0.0339
	NPCA	0.0054	0.0014	0.0020	0.0066	0.0150
	CTR	0.0070	0.0026	0.0017	0.0065	0.0105
	RMGM	0.0115	0.0036	0.0019	0.0074	0.0122
	cmLDA	**0.0337**	**0.0114**	**0.0084**	**0.0288**	**0.0610**

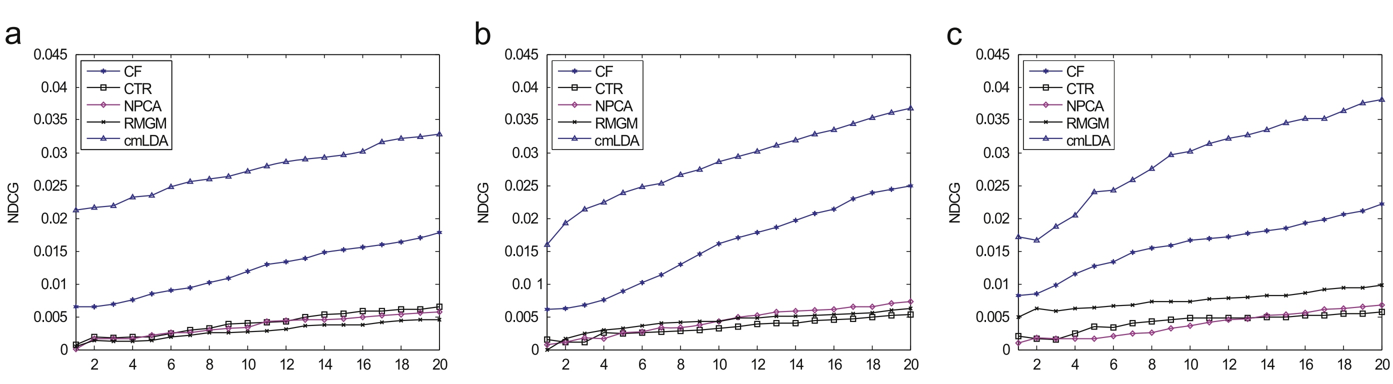


Fig. 4. NDCG on three domains in the Amazon dataset. (a) NDCG on Movie DVD, (b) NDCG on Music CD, and (c) NDCG on Book.

are learned simultaneously. In this way, cross media recommendation can be done by comparing topic distributions because of the common topic space assumption. Specifically, we consider each document contains three parts of information, document description data, user-generated text data and ratings. The document topic distributions are built by considering user interests besides document content. The experiments on datasets collected from MovieLens and Amazon have demonstrated the effectiveness of our proposed algorithm.

Acknowledgments

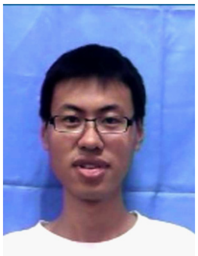
This work is partially supported by the National Basic Research Program of China (973 Program) under Grant No. 2013CB336500, National Natural Science Foundation of China (Grant No. 61173186, 61173185), National High Technology Research and Development Program of China (863 Program) under Grant No. 2013AA040601.

References

[1] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 734–749.
[2] W. Pan, E. Xiang, N.N. Liu, Q. Yang, Transfer learning in collaborative filtering for sparsity reduction, in: Proceedings of the 24th Conference on Artificial Intelligence (AAAI), Atlanta, Georgia, USA, 2010.
[3] B. Li, Q. Yang, X. Xue, Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction, in: Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, 2009.
[4] B. Li, Q. Yang, X. Xue, Transfer learning for collaborative filtering via a rating-matrix generative model, in: Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, Canada, 2009.
[5] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2009) 1345–1359.
[6] N.D. Phuong, T. M. Phuong, Collaborative filtering by multi-task learning, in: IEEE International Conference on Research, Innovation and Vision for the Future (RIVF), Hochiminh City, Vietnam, 2008.
[7] A.P. Singh, G.J. Gordon, Relational learning via collective matrix factorization, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, 2008.
[8] E.W. Xiang, N.N. Liu, S.J. Pan, Q. Yang, Knowledge transfer among heterogeneous information networks, in: IEEE International Conference on Data Mining Workshops, Miami, Florida, 2009.
[9] Y. Zhang, B. Cao, D.-Y. Yeung, Multi-domain collaborative filtering, in: Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI), Catalina Island, California, 2010.
[10] B. Cao, N.N. Liu, Q. Yang, Transfer learning for collective link prediction in multiple heterogenous domains, in: Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, 2010.
[11] W. Pan, N.N. Liu, E. Xiang, Q. Yang, Transfer learning to predict missing ratings via heterogeneous user feedbacks, in: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Barcelona, Spain, 2011.
[12] W. Pan, E. Xiang, Q. Yang, Transfer learning in collaborative filtering with uncertain ratings, in: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, Toronto, Canada, 2012.
[13] J. Tang, S. Wu, J. Sun, H. Su, Cross-domain collaboration recommendation, in: Proceedings of the 18th ACM SIGKDD international conference on Knowledge Discovery and Data Mining, Beijing, China, 2012.
[14] J. Jeon, V. Lavrenko, R. Manmatha, Automatic image annotation and retrieval using cross-media relevance models, in: Proceedings of the 26th ACM SIGIR Conference on Research and Development in Information Retrieval, Toronto, Canada, 2003.
[15] J.-Y. Pan, H.-J. Yang, C. Faloutsos, P. Duygulu, Automatic multimedia cross-modal correlation discovery, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, 2004.
[16] F. Monay, D. Gatica-Perez, Modeling semantic aspects for cross-media image indexing, IEEE Trans. Pattern Anal. Mach. Intell. 29 (10) (2007) 1802–1817.
[17] Y. Yang, D. Xu, F. Nie, J. Luo, Y. Zhuang, Ranking with local regression and global alignment for cross media retrieval, in: Proceedings of the 17th ACM International Conference on Multimedia, Beijing, China, 2009.
[18] N. Rasiwasia, J.C. Pereira, E.C. Gabriel Doyle, G.R. Lanckriet, R. Levy, N. Vasconcelos, A new approach to cross-modal multimedia retrieval, in: Proc. the 18th ACM International Conference on Multimedia, Florence, Italy, 2010.
[19] T. Hofmann, Collaborative filtering via gaussian probabilistic latent semantic analysis, in: Proceedings of the 26th ACM SIGIR conference on Research and Development in Informaion Retrieval, Toronto, Canada, 2003.
[20] B. Marlin, Modeling user rating profiles for collaborative filtering, in: Advances in Neural Information Processing Systems 16, Cambridge, MA, 2004.
[21] R. Krestel, P. Fankhauser, W. Nejdl, Latent dirichlet allocation for tag recommendation, in: Proceedings of the Third ACM Conference on Recommender Systems, New York, 2009.
[22] C. Wang, D.M. Blei, Collaborative topic modeling for recommending scientific articles, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, California, 2011.
[23] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[24] T. Hofmann, Probabilistic latent semantic indexing, in: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, USA, 1999.
[25] K. Barnard, P. Duygulu, F. Forsyth, N. de Freitas, D.M. Blei, M.I. Jordan, Matching words and pictures, J. Mach. Learn. Res. 3 (2003) 1107–1135.
[26] D.M. Blei, M.I. Jordan, Modeling annotated data, in: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, Toronto, Canada, 2003.
[27] T. Minka, J. Lafferty, Expectation–propagation for the generative aspect model, in: Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence, San Francisco, CA, 2002.
[28] T.L. Griffiths, M. Steyvers, Finding scientific topics, Natl. Acad. Sci. USA 101 (2004) 5228–5235.
[29] G. Heinrich, Parameter Estimation for Text Analysis, Technical Report, Fraunhofer IGD and University of Leipzig, Germany, 2009.
[30] P. Cremonesi, Y. Koren, R. Turrin, Performance of recommender algorithms on top-n recommendation tasks, in: Proceedings of the Fourth ACM Conference on Recommender Systems, Barcelona, Spain, 2010.
[31] N. Jindal, B. Liu, Opinion spam and analysis, in: Proceedings of the 1st International Conference on Web Search and web Data Mining, Stanford University, California, 2008.
[32] I. Konstas, V. Stathopoulos, J.M. Jose, On social networks and collaborative recommendation, in: Proceedings of the 32nd ACM SIGIR Conference on Research and Development in Information Retrieval, Boston, MA, 2009.
[33] J. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, San Francisco, CA, 1998.
[34] J.L. Herlocker, J.A. Konstan, A. Borchers, J. Riedl, An algorithmic framework for performing collaborative filtering, in: Proceedings of the 22nd ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, CA, 1999.
[35] K. Yu, S. Zhu, J. Lafferty, Y. Gong, Fast nonparametric matrix factorization for large-scale collaborative filtering, in: Proceedings of the 32nd ACM SIGIR Conference on Research and Development in Informaion Retrieval, Boston, 2009.
[36] J. Bu, S. Tan, C. Chen, C. Wang, H. Wu, X. He, Music recommendation by unified hypergraph: combining social media information and music content, in: Proceedings of the 18th ACM International Conference on Multimedia, Florence, Italy, 2010.

Shulong Tan received the BS degree in Software Engineering from Zhejiang University, China, in 2008. He is currently a PhD candidate in College of Computer Science, Zhejiang University, under the supervision of Prof. Chun Chen. His research interests include social network mining, recommender systems and text mining.

Jiajun Bu received the BS and Ph.D. degrees in Computer Science from Zhejiang University, China, in 1995 and 2000, respectively. He is a professor in College of Computer Science, Zhejiang University. His research interests include embedded system, data mining, information retrieval and mobile database.



Xuzhen Qin received his BS degree in Computer Science and Technology from Xiamen University, China, in 2011. He is currently working toward a master degree in College of Computer Science, Zhejiang University. His research interest includes recommender systems and data mining.

Chun Chen received the BS degree in Mathematics from Xiamen University, China, in 1981, and his MS and Ph.D. degrees in Computer Science from Zhejiang University, China, in 1984 and 1990, respectively. He is a professor in College of Computer Science, Zhejiang University. His research interests include information retrieval, data mining, computer vision, computer graphics and embedded technology.

Deng Cai is a Professor in the State Key Lab of CAD&CG, College of Computer Science at Zhejiang University, China. He received the Ph.D. degree in computer science from University of Illinois at Urbana Champaign in 2009. Before that, he received his Bachelor's degree and a Master's degree from Tsinghua University in 2000 and 2003 respectively, both in automation. His research interests include machine learning, data mining and information retrieval.