

# Exploit Latent Dirichlet Allocation for One-Class Collaborative Filtering

Haijun Zhang, Zhoujun Li, Yan Chen, Xiaoming Zhang, Senzhang Wang

State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China

haijun\_cumtb@126.com, lijz@buaa.edu.cn, yanchensmile@gmail.com, yolixs@buaa.edu.cn, szwang@cse.buaa.edu.cn.

## ABSTRACT

Previous work studied one-class collaborative filtering (OCCF) problems including pointwise methods, pairwise methods, and content-based methods. The fundamental assumptions made on these approaches are roughly the same. They regard all missing values as negative. However, this is unreasonable since the missing values actually are the mixture of negative and positive examples. A user does not give a positive feedback on an item probably only because she/he is unaware of the item, but in fact, she/he is fond of it. Furthermore, content-based methods, e.g. collaborative topic regression (CTR), usually require textual content information of items. This cannot be satisfied in some cases. In this paper, we exploit latent Dirichlet allocation (LDA) model on OCCF problem. It assumes missing values unknown and only models the observed data, and it also does not need content information of items. In our model items are regarded as words and users are considered as documents and the user-item feedback matrix denotes the corpus. Experimental results show that our proposed method outperforms the previous methods on various ranking-oriented evaluation metrics.

## Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Information filtering- *Recommendation System*.

## General Terms

Algorithms.

## Keywords

One-class Collaborative Filtering; Latent Dirichlet Allocation; Topic Model

## 1. INTRODUCTION

The goal of recommendation system is to automatically suggest items to each user that she/he may find appealing. Traditional collaborative filtering approaches predict users' interests by mining user rating history data, and these rating data are multi-valued scores, which can be categorized as "multi-class" recommendation problem [1]. Many machine learning based algorithms are designed to predict user's interesting on these multi-valued data, among of which matrix factorization based algorithm achieved great success [2].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

CIKM '14, November 3-7, 2014, Shanghai, China.

Copyright 2014 ACM 978-1-4503-2598-1/14/11...\$15.00.

<http://dx.doi.org/10.1145/2661829.2661992>

However, in many applications, the collected data of user behaviors are in "one-class" form rather than multi-class form, e.g., "like" in Facebook, "bought" in Amazon, "collect" in Taobao and "follow" in Sina weibo. Such data are usually called implicit [3] or one-class [1, 4, 5] feedback. The one-class collaborative filtering (OCCF) problem is different from that of multi-valued rating prediction problem, since the former only contains positive feedback rather than both positive feedback and negative feedback, and the goal is item ranking instead of rating prediction. Traditional machine learning based algorithms cannot directly be used to tackle OCCF because of imbalanced data [6].

The important difficulty to tackle OCCF problem is over fitting, because only positive feedbacks are observed. In order to avoid over fitting previous methods, including pointwise methods [4], pairwise methods [1, 3] and content-based methods [7, 8], all assume missing data negative. However they introduced newer over fitting problem caused by too many negative data.

A good learning method is that it fits the observed data well and as well as avoids over fitting. In this paper, similar to [7, 8], we introduce topic model, i.e., latent Dirichlet allocation [9, 10] (LDA) to deal with this problem. Our model is different from the methods proposed in [7, 8] in two aspects: (1) we only model the observed data, and the latter assume missing data negative and strive to fit all the data. (2) our model does not need contents information of items. Compared with the pointwise [4] and pairwise methods [1, 3], the parameters learned by our model are probability distributions, which have the inherent characteristics that the probability is a positive number and the summation of probabilities is equals to 1. These characteristics made our model have excellent ability to avoid over fitting. Experimental results show that our proposed method outperforms the previous methods on various ranking-oriented evaluation metrics.

The rest of the paper is organized as follows. In the next section, we review previous works related to the OCCF problems. In Section 3, we propose our approach for OCCF problems. In Section 4, we empirically compare our method to state-of-the-art methods on three real world data sets. Finally, we conclude the paper and give some future works.

## 2. RELATED WORKS

In this section, we review the literature of a few state-of-the-art approaches proposed for OCCF problems. There are mainly three types of approaches, (1) pointwise methods, (2) pairwise methods, and (3) content-based methods.

### 2.1 Pointwise Methods

Pointwise methods take implicit feedback as absolute preference scores. For example, an observed user-item pair is interpreted as a positive feedback and is assigned with a high absolute rating

score, e.g., 1, and the unobserved user-item pair is seen as a negative feedback and is assigned with rating score 0. Then, machine learning based methods, e.g. weighted low-rank approximations (WLRA) [4], are designed to fit the rating score matrix. Because most of the data are negative, in order to correct the bias there are two kinds of techniques are adopted, (1) the weight of the negative data is assigned with a small value relative to it of the positive data, and (2) the negative data are sampled with a small probability. The limitation of these methods is that taking unobserved user-item pairs as negative feedback may introduce errors.

## 2.2 Pairwise Methods

Pairwise methods achieved great success in OCCF problems, which mainly include Bayesian personalized ranking based matrix factorization (BPR) [3] and group Bayesian personalized ranking (GBPR) [1]. BPR supposes that a user  $u$  prefer an item  $i$  to an item  $j$  if the user-item pair  $(u, i)$  is observed, and  $(u, j)$  is not observed, and then take pairs of items as basic units and maximize the likelihood of pairwise preferences over observed items and unobserved items. Different from BPR, GBPR proposes a stronger constraint which can be expressed by Eq.(1) and Eq. (2):

$$\text{if } u \in G_i \text{ and } u \notin G_j \text{ then } r_{G_i,i} > r_{u,j} \quad (1)$$

$$r_{G_i,i} = \frac{\sum_{u \in G_i} r_{u,i}}{|G_i|} \quad (2)$$

Where  $G_i$  is the group of users who give a positive feedback to item  $i$ , and  $r_{u,j}$  denotes the preference value of user  $u$  on item  $j$ , and  $r_{G_i,i}$  denotes the group preference value of group  $G_i$  on item  $i$  and it is the average preference value of all users in group  $G_i$  on item  $i$  which is expressed by Eq.(2).

GBPR use sigmoid function  $f(r_{G_i,i} - r_{u,j}) = \frac{1}{1 + \exp(-r_{G_i,i} + r_{u,j})}$  to approximate the probability  $\Pr(r_{G_i,i} > r_{u,j})$ , and  $r_{u,j}$  is factorized by a product of two vectors which is expressed by Eq(3).

$$r_{u,j} = q_u \cdot s_j^T \quad (3)$$

Where  $q_u$  is the latent vector of user  $u$  and  $s_j$  is the latent vector of item  $j$ . These latent vectors are learned by maximize the likelihood of pairwise preferences.

Pairwise methods [1, 3] outperforms pointwise methods [4], but both of them have roughly the same fundamental assumptions that all missing data should be negative. Because according to their optimization goal, if a user-item pair  $(u, j)$  is unobserved, the smaller the preference of user  $u$  on item  $j$  is the easier the optimization goal can be realized. Matrix factorization technique is adopted in these methods to access to the optimization goal, and meanwhile it utilizes the interaction between latent vectors to bound the value of the preference.

## 2.3 Content-based Methods

In this OCCF scenario, items have textual contents. Topic model is used in this scenario to model the corpus and the user-item rating scores at the same time. Where an item is seen as a

document and all the items' contents constitute a corpus. Meanwhile, all missing data are assumed negative and are assigned with a small value, e.g. 0, and positive feedback are assigned with value 1 [7, 8]. The models proposed in [7] and [8] are called collaborative topic regression (CTR) and collaborative topic regression with social matrix factorization (CTR-SMF) respectively. Figure 1 shows the CTR model. CTR represents users with topic interests and assumes that items are generated by a topic model. CTR additionally includes a latent variable  $\varepsilon_j$  which offsets the topic proportions  $\theta_j$  when modeling the user ratings. This offset variable can capture the item preference of a particular user based on their ratings. Assume there are  $K$  topics, and let  $\beta = \beta_{1:K}$ . The generative process of CTR is as follows.

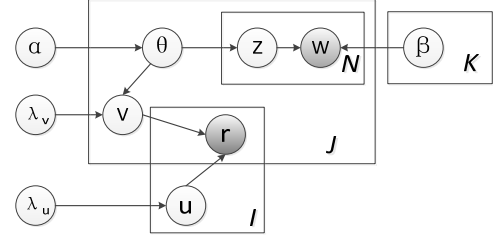


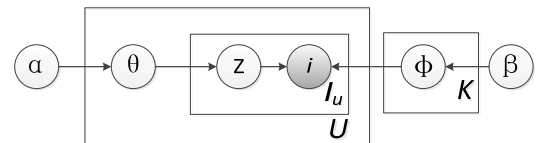
Figure 1. The graphical model for the CTR model.

1. For each user  $i$ , draw user latent vector  $u_i \sim N(0, \lambda_u^{-1} I_K)$ .
2. For each item  $j$ ,
  - (a) Draw topic proportions  $\theta_j \sim \text{Dirichlet}(\alpha)$ .
  - (b) Draw item latent offset  $\varepsilon_j \sim N(0, \lambda_v^{-1} I_K)$  and set the item latent vector as  $v_j = \varepsilon_j + \theta_j$ .
  - (c) For each word  $w_{jn}$ ,
    - i. Draw topic assignment  $z_{jn} \sim \text{Mult}(\theta_j)$ .
    - ii. Draw word  $w_{jn} \sim \text{Mult}(\beta_{z_{jn}})$ .
3. For each user-item pair  $(i, j)$ , draw the rating  $r_{ij} \sim N(u_i^T v_j, c_{ij}^{-1})$ .

Where  $c_{ij}$  is the precision parameter for  $r_{ij}$  which need be determined in advance by cross validation, and it represents the confidence for rating  $r_{ij}$ . For positive examples (i.e.  $r_{ij} = 1$ ),  $c_{ij}$  is set with a big value relative to negative examples (i.e.  $r_{ij} = 0$ ).

CTR-SMF model extended CTR model, and it can be used to address the recommendation problem when item's content, rating records and social network information are all known. However, in recommendation scenario, items usually have no textual contents, and these two methods assume missing data negative, which may not hold water, so, these two methods have limitations.

## 3. OUR METHOD



**Figure 2. LDA model for OCCF.**

In this paper, similar to the articles [7,8], we also use topic model to tackle OCCF problem, but in our model, a user is seen as a document, and an item is regarded as a word, and a positive feedback of user  $u$  for item  $i$  denotes an occurrence of word  $i$  in document  $u$ . We use latent Dirichlet allocation (LDA) [9] to model OCCF problem, which is represented by Figure 2.

Our method has two stages. **Step 1**, LDA model for OCCF was used to learn the matrices  $\theta$  and  $\phi$ , i.e., topic distributions of users and item distributions of topics respectively. The generative process is as follows.

1. For each user  $u$ , Draw topic proportions  $\theta_u \sim \text{Dirichlet}(\alpha)$ .
2. For each topic  $k$ , draw item proportions  $\phi_k \sim \text{Dirichlet}(\beta)$ .
3. For each observed user-item pair  $(u, i)$ ,
  - (a) Draw topic assignment  $z_{ui} \sim \text{Mult}(\theta_u)$ .
  - (b) Draw item  $i \sim \text{Mult}(\phi_{z_{ui}})$ .

We use Gibbs sampling method to estimate the parameters  $\theta$  and  $\phi$ , for more information please refer to [11]. Here, the topic number  $K$  and the super parameters  $\alpha$  and  $\beta$  need to be tuned.

**Step 2**, compute the preference of user  $u$  on item  $i$  by Eq. (4).

$$r_{ui} = \theta_{u,\cdot} \cdot \phi_{\cdot,i}^T = \sum_{k=1}^K \theta_{u,k} \phi_{k,i} \quad (4)$$

Where  $\theta_{u,k}$  is the distribution of user  $u$  on topic  $k$ , and  $\phi_{k,i}$  is the distribution of topic  $k$  on item  $i$ , and they are all positive numbers. From Eq.(4), we can see that only  $\theta_{u,k}$  and  $\phi_{k,i}$  are all bigger

numbers,  $r_{ui}$  becomes bigger. This conforms to actuality. For example, if user  $u$  very likes action movie, and movie  $i$  has many action scenes, we can concluded that user  $u$  has a bigger probability like this movie. Conversely, if user  $u$  hates to see action movie or movie  $i$  has no action scenes, he will not like this movie. Here, the genre of action denotes a topic. Thus, using Eq.(4) to measure the preference of user  $u$  on item  $i$  is reasonable other than the Euclidean distance between the two vectors. When a target user's preferences on all items are figured out, recommendation item list can be ranked.

We call this proposed method LDA-OCCF, which has three better characteristics than previous works: (1) Compared with the previous methods, in LDA-OCCF, missing ratings are assumed unknown other than negative; (2) Compared with CTR or CTR-SMF model, LDA-OCCF does not need contents information of items; (3) Compared with pointwise or pairwise methods, the parameters learned in LDA-OCCF are probability distributions, which are positive numbers and the summation of them is equal to 1. These strong constraints help LDA-OCCF avoid over fitting.

## 4. EXPERIMENTS

### 4.1 Datasets

We use three real-world datasets in our empirical studies, including MovieLens100K<sup>1</sup>, MovieLens1M<sup>1</sup>, and a subset of

Netflix<sup>2</sup>. MovieLens100K contains 100, 000 ratings assigned by 943 users on 1, 682 movies, MovieLens1M contains 1, 000, 209 ratings assigned by 6, 040 users on 3, 952 movies. Similar to [1], we sample 155,872 ratings larger than 3 as the observed positive feedbacks (to simulate the one-class feedback) from the Netflix dataset, and the ratings contain 5039 users and 5018 movies. We call this subset of Netflix dataset Netflix5K5K. We use “item” to denote movie. For MovieLens100K, MovieLens1M, we also take a pre-processing step, which only keeps the ratings larger than 3. Then, we randomly divide each dataset into 5 equal parts for five-fold cross validation, i.e., for each dataset, 4 parts of the data (i.e., 80%) is used to train the model and the rest data is used to test the model, and the reported result is the average value over 5 independent runs. The final datasets used in the experiments are shown in Table 1.

**Table 1. Datasets used in the experiments**

Datasets	Users	Items	Ratings
MovieLens100K	942	1447	55,375
MovieLens1M	6038	3533	575,281
Netflix5K5K	5039	5018	155,872

### 4.2 Evaluation Metrics

Because users usually only check a few top-ranked items, we use top-k evaluation metrics to study the recommendation performance, including top-k results of precision, recall, and MAP. For each evaluation metric, we first calculate the performance for each user from the test data, and then obtain the averaged performance over all users.

Precision is the proportion of top recommendation results that are relevant. Recall is the proportion of all relevant results included in the top results. They can be computed by Eq. (5) and (6) respectively.

$$\text{Prec} @ N = \frac{r}{N} \quad (5)$$

$$\text{Rec} @ N = \frac{r}{|T|} \quad (6)$$

Where  $r$  is the number of relevant items in top  $N$  recommendation list of a user  $u$  which means these  $r$  items are preferred by user  $u$  in test set, and  $|T|$  is the number of items that user  $u$  preferred in test set.

MAP (Mean Average Precision) is widely used in information retrieval for evaluating the ranked documents over a set of queries. We use it in this paper to assess the overall performance based on precisions at different recall levels on a test set. It computes the mean of average precision (AP) over all users in the test set. AP for user  $u$  is computed at the point of each of the preferred items in the ranked list:

$$AP_u = \frac{\sum_{i=1}^N \text{prec}(i) \times \text{pref}(i)}{\# \text{ of preferred items}} \quad (7)$$

where  $i$  is the position in the rank list,  $N$  is the number of recommended items which are the items that user  $u$  did not express positive feedback in training set,  $\text{prec}(i)$  is the precision (fractions of recommended items that are preferred by user  $u$  in

<sup>1</sup> <http://grouplens.org/datasets/movielens/>

<sup>2</sup> <http://www.netflixprize.com/>

test set) of a cut-off rank list from 1 to  $i$ , and  $pref(i)$  is a binary indicator returning 1 if the  $i$ -th item is preferred by user  $u$  or 0 otherwise. The MAP is the mean of  $AP_u$  over all users.

### 4.3 Baselines and Parameter Settings

We use three popular baseline algorithms in our experiments, PopRank, BPR [3], and GBPR [1]. PopRank is a basic algorithm in one-class collaborative filtering, which ranks the items according to their popularity in the training data. BPR [3] is a seminal work for OCCF problem and is also a very strong baseline, which is shown to be much better than two well-known pointwise methods, i.e., iMF [12] and OCCF [4]. Note that the implementation of this baseline is done with the publicly available software MyMediaLite [13]. GBPR [1] is recently proposed method, which outperforms BPR.

The parameters in BPR and GBPR are tuned according to the papers [3] and [1] respectively. In our method LDA-OCCF, for all experiments, the topic number  $K$  is tried with  $\{10, 20, 50\}$ , the parameter  $\alpha$  is set to 0.5 and  $\beta$  is set to 0.02, and Gibbs sampling times are chosen from  $\{1000, 10000, 100000\}$ .

### 4.4 Experimental Results

The recommendation performance of LDA-OCCF and other baselines are shown in Table 2, from which we can have the following observations.

1. Both BPR, GBPR and LDA-OCCF are much better than the PopRank algorithm, which means that personalized recommendation method is better than most popularity based recommendation method.
2. LDA-OCCF further outperforms BPR and GBPR on all evaluation metrics on all three datasets, which shows the effectiveness of our method, and validates its better characteristics described in section 3.

On the other hand, LDA-OCCF only samples observed data, so its computation complexity is lower than BPR and GBPR if all of them take the same sampling times.

**Table 2. Compared Results with Baselines**

Dataset	Algorithm	Prec@5	Rec@5	MAP
MovieLens 100K	PopRank	0.1418	0.0650	0.1081
	BPR	0.2135	0.1308	0.1798
	GBPR	0.2565	0.1412	0.2106
	LDA-OCCF	<b>0.2642</b>	<b>0.1481</b>	<b>0.2119</b>
MovieLens 1M	PopRank	0.1518	0.0485	0.0905
	BPR	0.2602	0.1034	0.1752
	GBPR	0.2696	0.1055	0.1815
	LDA-OCCF	<b>0.2897</b>	<b>0.1091</b>	<b>0.1877</b>
Netflix 5K5K	PopRank	0.0101	0.0507	0.0432
	BPR	0.1150	0.05968	0.1009
	GBPR	0.1213	0.05183	0.0987
	LDA-OCCF	<b>0.1451</b>	<b>0.0708</b>	<b>0.1170</b>

## 5. CONCLUSIONS

In this paper, we study the one-class collaborative filtering problem and design a topic model based algorithm LDA-OCCF. Different from previous proposed methods, it avoided the

unreasonable assumption that missing data are negative. Due to the strong constraint that the learned probabilities are positive and their summation is equal to 1, LDA-OCCF alleviates the over fitting problem. Meanwhile, different from previous topic model based recommendation methods, it does not need content information of items. Experimental results show that, this proposed method outperforms the compared algorithms. In the future, we will further study how to integrate more information such as social structure into the algorithm to improve the performance further.

## 6. ACKNOWLEDGMENTS

We greatly appreciate Weike Pan, i.e., the author of paper [1], for his codes of algorithm GBPR, which makes us able to evaluate the algorithm more efficiently and more fairly. This work is supported by NSFC (Nos.61170189, 61370126, 61202239), the Research Fund for the Doctoral Program of Higher Education (No. 20111102130003), the Fund of the Stage Key Laboratory of Software Development Environment (No. SKLSDE-2013ZX-19), Microsoft Research Asia Fund (No. FY14-RES-OPP-105), the Fund of Beijing Social Science [No.14JGC103], and the Statistics Research Project of National Bureau [No. 2013LY055]. The third author thanks the Innovation Foundation of Beihang University for Ph.D. Graduates (YWF-13-T-YJSY- 024).

## 7. REFERENCES

- [1] Pan W, Chen L. GBPR: group preference based Bayesian personalized ranking for one-class collaborative filtering[C]. AAAI 2013: 2691-2697.
- [2] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30-37.
- [3] Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian personalized ranking from implicit feedback[C]. AUAI 2009: 452-461.
- [4] Pan R, Zhou Y, Cao B, et al. One-class collaborative filtering[C]. ICDM 2008, 502-511.
- [5] Li Y, Hu J, Zhai C X, et al. Improving one-class collaborative filtering by incorporating rich user information[C]. CIKM 2010: 959-968.
- [6] He H, Garcia E A. Learning from imbalanced data[J]. TKDE 2009, 21(9): 1263-1284.
- [7] Wang C, Blei D M. Collaborative topic modeling for recommending scientific articles[C]. KDD 2011: 448-456.
- [8] Purushotham S, Liu Y, Kuo C C J. Collaborative topic regression with social matrix factorization for recommendation systems[J]. arXiv preprint arXiv:1206.4684, 2012.
- [9] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. the Journal of machine Learning research, 2003, 3: 993-1022.
- [10] Chen Y, Yin X, Li Z, et al. A LDA-based approach to promoting ranking diversity for genomics information retrieval[J]. BMC genomics, 2012, 13(Suppl 3): S2.
- [11] Heinrich G. Parameter estimation for text analysis[R]. Technical report, 2005.
- [12] Hu Y, Koren Y, Volinsky C. Collaborative filtering for implicit feedback datasets[C]. ICDM'08: 263-272.
- [13] Gantner Z, Rendle S, Freudenthaler C, et al. MyMediaLite: A free recommender system library[C]. RecSys 2011: 305-308.