

Dual Role Model for Question Recommendation in Community Question Answering

Fei Xu^{1,2}, Zongcheng Ji^{1,2}, Bin Wang^{1,3}

¹Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

²Graduate University of Chinese Academy of Sciences, Beijing, China

²{feixu1966, jizongcheng}@gmail.com

³wangbin@ict.ac.cn

ABSTRACT

Question recommendation that automatically recommends a new question to suitable users to answer is an appealing and challenging problem in the research area of Community Question Answering (CQA). Unlike in general recommender systems where a user has only a single role, each user in CQA can play two different roles (dual roles) simultaneously: as an asker and as an answerer. To the best of our knowledge, this paper is the first to systematically investigate the distinctions between the two roles and their different influences on the performance of question recommendation in CQA. Moreover, we propose a Dual Role Model (DRM) to model the dual roles of users effectively. With different independence assumptions, two variants of DRM are achieved. Finally, we present the DRM based approach to question recommendation which provides a mechanism for naturally integrating the user relation between the answerer and the asker with the content relevance between the answerer and the question into a unified probabilistic framework. Experiments using a real-world data crawled from Yahoo! Answers show that: (1) there are evident distinctions between the two roles of users in CQA. Additionally, the answerer role is more effective than the asker role for modeling candidate users in question recommendation; (2) compared with baselines utilizing a single role or blended roles based methods, our DRM based approach consistently and significantly improves the performance of question recommendation, demonstrating that our approach can model the user in CQA more reasonably and precisely.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Information Filtering

General Terms

Algorithms, Design, Experimentation

Keywords

Community Question Answering, Role Analysis, Question Recommendation, Dual Role Model, PLSA

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. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.

Copyright 2012 ACM 978-1-4503-1472-5/12/08 ...\$15.00.

1. INTRODUCTION

Community Question Answering (CQA) is a web service where people can seek information (posting a question and getting the answer of it from others) and share knowledge (answering a question). Yahoo! Answers¹ and Baidu Zhidao² are two typical examples of CQA system. Compared with the traditional information retrieval, CQA bases on the community, which is a form of social network, so it can make best of user's collective wisdom to meet the information needs of users more easily and accurately.

In CQA system, there are a large number of questions posted every day. Take Yahoo! Answers for example, there are about 207 thousands new questions asked daily [1]. If we can automatically recommend the new question to appropriate users to answer, it will help the question be resolved as soon as possible, which will improve the CQA system's performance. In addition, it will meet the answerers' needs to answer questions. As we can see, question recommendation is a very important component in a CQA system.

The core issue of question recommendation is how to represent the users' interests (profile) and the questions, which is called the representation model. Based on that, we can assess the match between a question and each user, and then recommend the question to top N users who are the most consistent with it. Of course, we can solve question recommendation from another perspective, which is matching a user with each question and recommending the appropriate questions to him. Both of these types of recommendation tasks aim to make new questions answered as early as possible and satisfy the user better. Essentially, the key issues of both of them are the representation models for users and questions. As our target is recommending a new question to the appropriate users to answer, this paper focuses on the first type of recommendation task. At present, a lot of representation models have been proposed. Dror et al [5] represented the user and question as vectors consisting of multi-channel features and casted question recommendation as a classification problem. Other methods [2, 4, 6] utilized latent semantic models (PLSA, LDA, etc.) to model the user and question as the distribution of several topics.

As we can see, each user in CQA plays two different roles (dual roles) simultaneously: the asker and the answerer. That is, a user not only posts his questions, but also is able to answer someone else's questions. Intuitively, the profiles of the two roles of users are different from each other, which existing methods have not paid attention to. For example, a piano teacher wants to learn some computer knowledge which he is not familiar with. Thus he is most likely to ask lots of questions related to computer, and answer many piano-related questions based on his specialty. As an asker, a user may post some questions in

¹ <http://answers.yahoo.com/>

² <http://zhidao.baidu.com/>

the field that he is not familiar with. In contrast, as an answer, the user will solve the question which he is good at and interested in.

Are there distinctions between users' roles? How do different roles affect the performance of question recommendation? Whether we can legitimately combine the characteristics of different roles to improve the effectiveness of the recommendation system? All of these important issues are worthy of our concern. However, current recommendation methods have not in-depth studied the different characteristics of users' roles and their different influences on question recommendation. All of previous methods only modeled the user using a single role, or simply mixed the two roles together to represent the user without considering the distinctions between roles.

This paper systematically investigates the distinctions between users' dual roles and how they affect the performance of question recommendation differently. To the best of our knowledge, this is the first work on studying these important issues. While Nam et al. [30] observed that users in CQA are divided into askers and answerers and only a few of them both ask and answer in the same category through statistics, they have not theoretically analyzed the distinctions between users' different roles and their different influences on question recommendation. Moreover, we propose the Dual Role Model (DRM) to model the dual roles of users effectively. Finally, we present the DRM based approach to question recommendation, which takes full advantage of users' different roles to improve the effect of question recommendation. There are three primary contributions of our work.

First, DRM which considers the two different roles of users separately provides a more precise and appropriate user representation model for question recommendation in CQA. Specifically, we utilize DRM to analyze the latent topic information of different roles for modeling the user. According to different independence assumptions, two variants of DRM are achieved: (1) independent DRM that assumes that users are independent of each other and models each user individually; (2) dependent DRM which considers the dependence between users.

Next, we carried out systematic experiments on a real-world data to explore the distinctions between users' roles and compare the effects of recommendation methods that are based on asker role, answerer role or blending both of these roles. The results show that not only the two roles but also their influences on question recommendation are different from each other distinctly. In addition, simply mixing the roles together will impair the performance of recommender methods.

Finally, our DRM based recommendation approach allows us to naturally integrate the user relation between the answerer and the asker with the content relevance between the answerer and the question into a unified probabilistic framework, which is more interpretable. Most previous methods only consider the content relevance. There have been several approaches that make use of the user relation [3, 5], however in these approaches, the user relation is either obtained through somewhat heuristic statistics outside of the model or combined with the content relevance by a linear interpolation.

The remainder of this paper is organized as follows: Section 2 introduces some prior work related to our approach. Section 3 is the preliminary description of question recommendation in CQA. Section 4 discusses our dual role model and how to use it in question recommendation. Experimental results are presented in Section 5. At last, we conclude the paper and discuss about the future work in Section 6.

2. RELATED WORK

In this part, we review previous work which is related to our approach: recommender system, question recommendation.

2.1 Recommender System

Because question recommendation is a type of recommender system, we first review general recommender systems. Recommender systems can be divided into three stages based on how recommendations are made: content-based recommendations, collaborative filtering and hybrid approaches [18]. In content-based recommendations, the user will be recommended items similar to the ones the user preferred in the past. In collaborative filtering, the user will be recommended items that people with the similar tastes and preferences liked in the past, that is, user will help each other find what they may like. In order to combine the advantage of both previous methods together, hybrid approaches are proposed. All these recommender systems firstly attempt to profile user preferences based on his history logs, and then recommend items according to the relevancy between him and items. Different kinds of methods are used to capture the model of users, such as classifying [24, 28], PLSA [13], matrix factorization [29], and ranking-oriented approach [17]. However, the user in these general recommender systems only plays one single role, which is significantly different from question recommendation. Therefore, we should pay close attention to this difference as we have mentioned in the above section.

2.2 Question Recommendation

With CQA system becoming popular in recent years, many people turn their attention to question recommendation in CQA, e.g., [2, 3, 4, 5, 6, 16]. Overall, there are two main lines to solve this problem in previous work.

On one hand, question recommendation is considered as a classifier problem which is similar to [5]. In [5], Dror et al. proposed a representation model based on multi-channel vector space model, where the user and question are represented as the vector with multiple dimension features from multi-channel data. Then, the matching degree between a user and a question is learned from their respective features using a binary classifier. Although this model treats user attributes in the answered-channel and asked-channel as two groups of features respectively, all the features are integrated into a single vector space model to represent the user's dual roles without considering the distinctions between user's different roles and their different influences on question recommendation.

On the other hand, we can learn a ranking model to generate a recommendation list for question recommendation. In these earlier works, various extensions of Probabilistic Latent Semantic Analysis (PLSA) or other topic models are developed. Wu et al. [2] presented an incremental automatic question recommendation framework based on PLSA. Question recommendation in their work considered both the users' interests and feedback. Guo et al. [4] developed a general generative model based on basic Latent Dirichlet Allocation (LDA) model for questions and answers in CQA. In this approach, they combined topic-level information about questions and users with word-level information to improve question recommendation. In order to deal with the data sparsity, Qu et al. [6] used a user-word aspect model instead of direct aspect model [9] to model user preferences. However, all of these methods have used a single role, or simply blended roles to represent the user, which have not distinguished user's different roles and considered how they affect the performance of question recommendation differently.

3. PRELIMINARIES

Given the question set $Q = \{q_1 \dots q_{|Q|}\}$ and the user set $U = \{u_1 \dots u_{|U|}\}$, where $|Q|$ is the number of questions and $|U|$ is the number of users. Each question in Q is denoted as a triple $q = \langle t, u^a, u^q \rangle$. The t is the text content of the question. For instance, t may include the title or the detailed description of the question. If we assume that words are independent, t can be denoted as a bag of words $\{w_1 \dots w_{|t|}\}$, where $|t|$ is the number of words in t . The u^a denotes the answerer of the question (it is also the answerer role of user u). The u^q denotes the asker of the question (it is also the asker role of user u). If there are multiple answerers in the question, all answerers will be separated. If the question is not answered, u^a is null.

Based on the previous discussion in the section of related work, we choose the idea of ranking to solve the problem of question recommendation. For a new posted question, the question answerer recommendation task is to suggest a ranked list of users who are suitable to answer it. To tackle with this problem, we need to resolve the two sub-problems: question and user representation, the method of ranking recommendation candidates.

Since Probabilistic Latent Semantic Analysis (PLSA) [20] can effectively mines the latent semantic information of users and questions, it has been widely used to obtain the question and user representation in question recommendation, e.g., [2, 3, 4, 6]. PLSA assumes that users and questions are generated from a mixture of some latent topics. We can compute the consistency between the distribution on topics of a user and a question to determine whether to recommend the question to the user. We summarized the previous PLSA based methods for question recommendation and discovered that they can be divided into two main categories: (1) methods that model the user indirectly. Similarly to [2], it takes a question as one document and use PLSA to model the question to gain its distribution on topics at first. Then the user can be represented as the average of topic distributions of all the questions that he accesses; (2) methods that obtain the model of the user directly. In these methods, all the questions that a user accesses are treated as one document. Then PLSA is used directly to get the topic information of the user. A typical approach is the user-word aspect model applied by Qu et al. [6]. This model is proposed by Popescul et al. [7], which improves Hofmann's aspect model [9] for collaborative filtering.

However, when these PLSA based methods modeling the user, they did not pay attention to the user's dual roles and their distinctions. In order to effectively analyze characteristics of different roles and make use of both of user roles to improve the performance of question recommendation, we propose a Dual Role Model (DRM) based on PLSA to model the user in CQA precisely. According to different independence assumptions, we implement two variants of DRM. In the next section, we will detail generation processes of these variants and describe the DRM based method for question recommendation.

4. MODEL DESCRIPTION

4.1 Independent DRM

With the assumption that all users are independent of each other in independent DRM (IDRM), we separately model the dual roles of each user. As Figure 1 illustrates, the IDRM can be divided into two steps. First, we employ the PLSA to analyze the topic information of all the questions, and then model the answerer role and asker role of each user based on questions which he answers or asks.

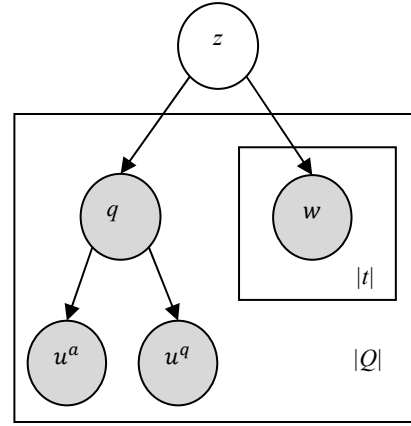


Figure 1: Independent DRM.

We introduce the latent variable $z \in Z = \{z_i\}$ to indicate each topic under users and questions. The model of user's answerer role can be represented as its topic distribution $P(u^a|z)$. Similarly, the asker role is characterized by $P(u^q|z)$ and the latent topic information of the question is $P(q|z)$. According to the first step of IDRM in the Figure 1, the generative model for question/word co-occurrences is defined as: a latent topic z is obtained with probability $P(z)$, and then a question q is generated with probability $P(q|z)$ and a word w is generated with probability $P(w|z)$. Therefore, we can compute the joint probability $P(q, w)$ of observing a question q together with a word w based on topic variable z as follows:

$$P(q, w) = \sum_z P(z)P(q|z)P(w|z)$$

Then considering all question/word pairs $\langle q, w \rangle$ in question set Q , the log likelihood L is

$$L = \sum_{q,w} c(q, w) \log P(q, w)$$

where $c(q, w)$ is the frequency of word w in the question q .

We use the Expectation Maximization (EM) method to learn the model parameters $P(z)$, $P(q|z)$ and $P(w|z)$:

E-Step.

$$P(z|q, w) = \frac{P(z)P(q|z)P(w|z)}{\sum_{z'} P(z')P(q|z')P(w|z')}$$

M-Step.

$$P(z) \propto \sum_{q,w} c(q, w)P(z|q, w)$$

$$P(q|z) \propto \sum_w c(q, w)P(z|q, w)$$

$$P(w|z) \propto \sum_q c(q, w)P(z|q, w)$$

After obtaining all questions' representations, we perform the second step to get the representations of users' different roles. The user's answerer role is defined as the combination of topic distributions of all questions that he answers, and the modeling method is similar for the asker role. Intuitively, we can give an example to illustrate the feasibility of this approach. For example, if a user answers lots of questions related to using computer, so the profile

of his answerer role is very likely to be related to this topic. Specifically, the role models $P(u^a|z)$ and $P(u^q|z)$ are estimated as:

$$P(u^a|z) = \frac{\sum_{q \in \text{Answer}(u)} P(q|z)}{\sum_{z'} \sum_{q \in \text{Answer}(u)} P(q|z')}$$

$$P(u^q|z) = \frac{\sum_{q \in \text{Ask}(u)} P(q|z)}{\sum_{z'} \sum_{q \in \text{Ask}(u)} P(q|z')}$$

where, $\text{Answer}(u)$ is the set of questions that the user u answers, and $\text{Ask}(u)$ is the set of question he asks.

4.2 Dependent DRM

Different from the IDRM, the assumption made in dependent DRM (DDRM) is that there is dependence between users. As we can see in Figure 2, DDRM assumes that the answerer and the asker are dependent on each other when not observing the latent variable. The assumed generative model is as follows. We first pick a latent topic to some prior $P(z)$. We then generate the answerer u^a , the asker u^q , and the content $t = \{w_i\}$ of question q with corresponding probability $P(u^a|z)$, $P(u^q|z)$, and $\prod_{w \in t} P(w|z)^{c(w,t)}$. Thus, the joint probability distribution of a triple $\langle t, u^a, u^q \rangle$ of question q is defined as:

$$P(t, u^a, u^q) = \sum_z P(z) P(u^a|z) P(u^q|z) \prod_{w \in t} P(w|z)^{c(w,t)}$$

In the above equation, $c(q, w)$ is the frequency of word w in the content t .

Accordingly, the log likelihood L in DDRM is

$$L = \sum_{t, u^a, u^q} \log P(t, u^a, u^q)$$

and we can also train the model using EM method as follows:

E-step,

$$P(z|t, u^a, u^q) = \frac{P(z) P(u^a|z) P(u^q|z) \prod_{w \in t} P(w|z)^{c(w,t)}}{\sum_{z'} P(z') P(u^a|z') P(u^q|z') \prod_{w \in t} P(w|z')^{c(w,t)}}$$

M-step,

$$\begin{aligned} P(z) &\propto \sum_{t, u^a, u^q} P(z|t, u^a, u^q) \\ P(w|z) &\propto \sum_{w \in t, u^a, u^q} c(w, t) P(z|t, u^a, u^q) \\ P(u^a|z) &\propto \sum_{t, u^q} P(z|t, u^a, u^q) \\ P(u^q|z) &\propto \sum_{t, u^a} P(z|t, u^a, u^q) \end{aligned}$$

The IDRM and the DDRM respectively model the user's dual roles from different perspectives. Compared with previous models that do not take the dual roles and their distinctions into account, DRM provides a more precise and appropriate user representation model for question recommendation. Apart from different independence assumptions between users, we can see that the IDRM is a type of method modeling user role indirectly while the DDRM is a method which learns the role model directly.

4.3 Question Recommendation

Based on any one of the above DRM variants, we build the DRM based method for question recommendation that takes full advantage the characteristics of different user roles. When a new question q arriving, we compute posterior probability $P(u_i|q)$ for

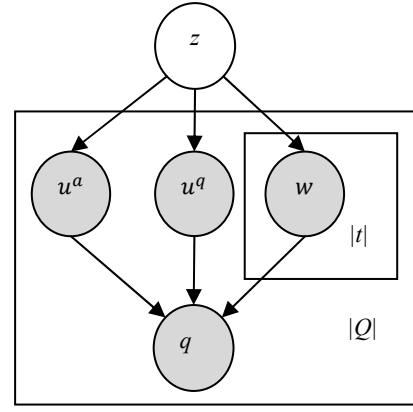


Figure 2: Dependent DRM.

each candidate user u_i , and then recommend this question to the top N users. $P(u_i|q)$ is obtained by:

$$\begin{aligned} P(u_i|q) &\propto P(u_i, q) \\ &= P(u_i^a, u^q, t) \\ &= \sum_z P(z) P(u_i^a|z) P(u^q|z) P(t|z) \\ &= \sum_z P(z) P(u_i^a|z) P(u^q|z) \left[\prod_{w \in t} P(w|z)^{c(w,t)} \right]^{1/|t|} \end{aligned}$$

where, the first step uses the Bayesian formula for an equivalent transformation. In the second step, the question q is decomposed into its content t and its asker u^q . In addition, we only need to consider the candidate's answerer role when we evaluate whether he is suitable to answer this question. Therefore, the u_i is represented as his answerer role u_i^a . The third step and fourth step is based on the role models and the question model obtained in DRM, where the generation probability $P(t|z)$ is normalized by the length of question content $|t|$.

In this recommendation approach, $P(u_i^a|z)P(u^q|z)$ denotes the consistency of the answerer and the asker over topics, which models the user relation between the answerer and the asker. Correspondingly, $P(u_i^a|z) \prod_{w \in t} P(w|z)^{c(w,t)}$ measures the consistency of the answerer and the question content over topics, which models the content relevance between the answerer and the question. As we can see, our DRM based method takes full advantage of users' dual roles to improve the performance of question recommendation. Moreover, this method utilizes a unified probabilistic framework to naturally associate the user relation with the content relevance together, which is more interpretable.

Compared with the DRM, both the methods described in [2] and [6] employed a single role model to represent the user and ignored the user relation when recommending question to users. In the next section, these methods will be used as two groups of baselines in our experiments.

5. EXPERIMENTS

We evaluate the proposed approach using a real-world data from Yahoo! Answers and conduct different experiments to address the following questions: (1) Are there any distinctions between users' dual roles and how they affect the result of question recommendation? (2) Does the proposed DRM improve the effectiveness of question recommendation compared with other base-

line methods? (3) Which of the two variants of DRM for question recommendation, namely IDR and DDR, is more effective?

5.1 Data Sets

In order to obtain the data sets for experiments, we used Yahoo! Answers API³ to crawl 246490 resolved questions posted in 2011 from Yahoo! Answers. All the questions are lowercased and all stop words are removed from questions using a standard list of 418 common terms before further experiments.

In our question set Q , we divide the whole question set and user set (Q, U) into three subsets according to the user participation degree. For each subset, we split it into the training set and the testing set based on the asked time of questions. The training set is used solely for parameter estimation and the test set is used for evaluation purposes. In each subset, we take about 9/10 of questions as training set, and the rest as testing set. The data set statistics of all subsets are listed in Table 1. Each dataset contains a question set and a user set. For instance, the question set and user set of User-10 are Q_{10} and U_{10} . We selected users who asked or answered more than 10 questions as the user set U_{10} and then collected questions which were asked or answered by users in U_{10} as the question set Q_{10} . Other subsets are similar to User-10.

	Question Number	Answer Number	User Number
User-10	32009	97911	2515
User-15	28404	89144	1339
User-20	25690	80677	870

Table 1: Statistic of Yahoo! Answers data set.

5.2 Evaluation Metric

In traditional recommender systems, precision is a commonly used measure to evaluate the performance. However, precision is not suitable in the CQA context. There are so many questions asked in a CQA community every day [1] that the user can only access a very small portion of all questions. While the questions one accessed are those he is interested in, we can not guarantee that the remaining unaccessed questions are those he does not like. That is, in some cases, a user did not access a question just because he had no chance to see the question in CQA system. Therefore, we employ a new metric proposed in [6] to evaluate the effectiveness of question recommendation in CQA.

For a question in testing set, the user who provides the best answer (named the best answerer, Adamic et al. [27] have verified that answers selected as the best ones are mostly indeed the most suitable for the questions.) of this question is seemlier to answer it compared with other answerers, so it is more reasonable to recommend this question to the best answerer than other answerers. Based on this intuition, we only recommend the question to the users who actually answered it instead of all possible users in the whole dataset. Then the recommendation accuracy for this question is defined according to the rank of the user who provides the best answer. (We only keep the questions which have more than one answer and are already labeled with the best answers in the testing set.) Therefore, according to the evaluation metric applied

in [6], we utilize the best answerer’s rank as the ground truth of our evaluation metric:

$$accuracy = \frac{|R| - R_{best} - 1}{|R| - 1}$$

where $|R|$ is the length of recommending list, which is equally the number of answers, and R_{best} is the rank of the best answerer.

5.3 Role Analysis

We first discuss an interesting subject: the distinction between the user’s two roles. Based on latent topic analysis of user roles in DRM, the distinction between the answer role and the asker role is defined as the difference between their topic distributions. The larger the difference between the topic distributions is, the greater the distinction between roles is. In information theory, the Kullback-Leibler divergence (KL divergence) is a commonly used non-symmetric measure of the difference between two probability distributions. We apply the KL divergence to assess the distinction between user roles. According to the modeling results of user roles in DRM, we obtain the latent topic distributions of each user role:

$$P(z|u^a) = \frac{P(u^a|z)P(z)}{P(u^a)} = \frac{P(u^a|z)P(z)}{\sum_{z'} P(u^a|z')P(z')}$$

$$P(z|u^q) = \frac{P(u^q|z)P(z)}{P(u^q)} = \frac{P(u^q|z)P(z)}{\sum_{z'} P(u^q|z')P(z')}$$

Based on the above two equations, the distinction between answer role and asker role of the user u is

$$D(u^a||u^q) = \sum_z P(z|u^a) \log \frac{P(z|u^a)}{P(z|u^q)}$$

In DRM, the number of topics is a parameter that has significant impact on the performance. We utilize cross-validation to estimate the parameters. Based on experiments of tuning parameter, we empirically set topic number to 70 to train our DRM.

First, we analyze the average of KL divergence of all users in the user set of each subset to measure the overall distinction between user roles. The results are summarized in Table 2. Across data subsets, the overall role distinction in IDR is about 1.3 to 1.5, and that in DDR is about 2.4. Compared with IDR, the role distinction in DDR is greater and relatively more stable over different data subsets.

	User-10	User-15	User-20
IDR	1.349	1.379	1.502
DDR	2.440	2.467	2.441

Table 2: The overall distinction between user roles.

Furthermore, we take User-10 as an example to detail the distribution of role distinction, which is illustrated in Figure 3. For the DDR, role distinction of 65.5% of users is more than 1.0, and most of them is in the range of [0.5, 3.5]. For the IDR, role distinction of 74.3% of users is more than 2, most of which is in the range of [1.5, 4.5]. This shows that there are clear differences between different roles of most of users in CQA.

Another important result to note is that the role distinction in DDR is more obvious and relatively more stable than that in IDR, which can be observed in both of overall and detailed role analysis. This result may be due to that DDR models the depen-

³ <http://developer.yahoo.com/answers>

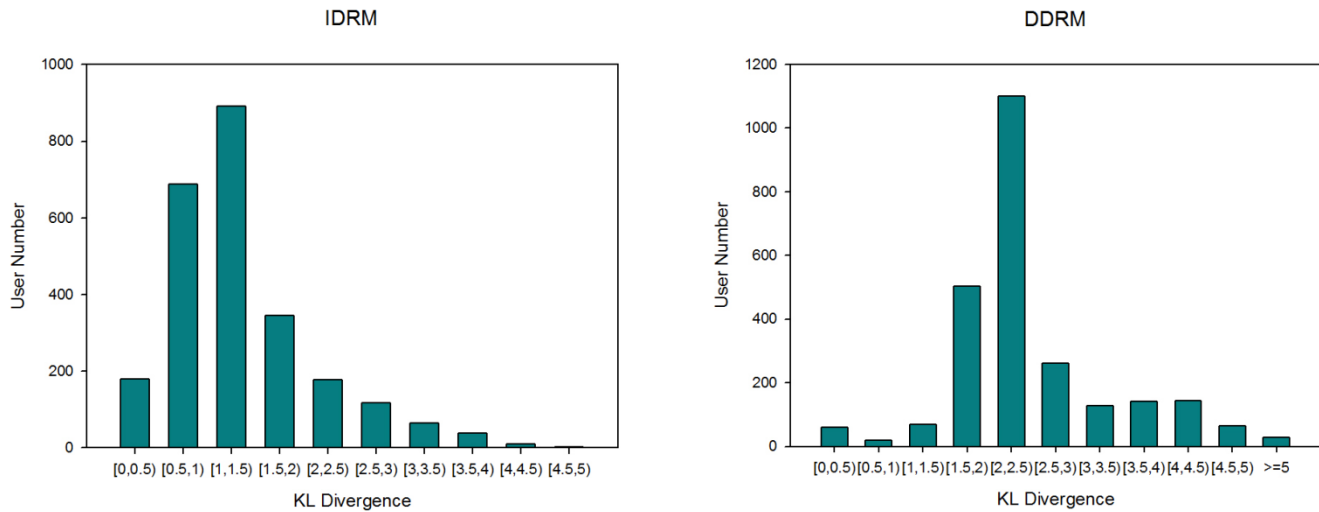


Figure 3: Distribution of the role distinction.

User Role	Topic ID	Top Words
Answerer	41 (using computer)	free, best, windows, antivirus, virus, anti, software, spyware, download, program, xp, vista, hard, drive
Asker	30 (programming)	c, program, file, java, programming, write, language, convert, system, net, code, array, php, number

Table 3: An example of the difference between user roles.

Role	VSM	PLSA1 (modeling user indirectly)	PLSA2 (modeling user directly)
Answerer Role	VSM-an	PLSA1-an	PLSA2-an
Asker Role	VSM-as	PLSA1-as	PLSA2-as
Blended Roles	VSM-bl	PLSA1-bl	PLSA2-bl

Table 4: All versions of baselines considering users' different roles.

dence between users which is more effective to capture the peculiarities of different user roles.

After discussing the role analysis of all users, we take a typical user in DDRM as an illustrative example to show the modeling results of user roles. Table 3 lists the topic with maximum probability and corresponding top words of the topic for both of user roles. As the result shows, this user is most likely to be a junior programmer (such as junior college students from school of computer science). He has some basic knowledge of computer and is familiar with using computer, so he solved many questions about that topic. While he may be just getting started with computer programming, a lot of questions he asked are related to programming.

5.4 Question Recommendation

5.4.1 Result Comparison

In this section, we explored how different roles of users affect the result of question recommendation. Moreover, we compared our DRM-based question recommendation method with other methods.

The models proposed in previous work are classified into two main categories. The first one is the word-level Vector Space Model (VSM) which is directly used to compute the similarity between users and questions. VSM only make use of word-level information to model users and questions. For example, the user and question are represented as vectors with $tf.idf$ word weights, and then cosine similarity between them is defined as:

$$\begin{aligned}
 similarity(u, q) &= \frac{\vec{u} \cdot \vec{q}}{\|\vec{u}\| \cdot \|\vec{q}\|} \\
 &= \frac{\sum_w tf.idf(w, u) tf.idf(w, q)}{\sqrt{\sum_w tf.idf(w, u)^2} \sqrt{\sum_w tf.idf(w, q)^2}}
 \end{aligned}$$

where $tf.idf(w, q)$ is the word w 's $tf.idf$ weight in question q , and $tf.idf(w, u)$ is the sum of w 's $tf.idf$ weights in questions that u asks or answers.

The second one is PLSA based methods. As we have specified in section 3, these methods model the user either indirectly or directly. For the former, we took the model in [2] (PLSA1) as

		User-10	User-15	User-20
VSM	VSM-an	<u>0.555</u>	<u>0.600</u>	<u>0.612</u>
	VSM-as	0.456	0.492	0.466
	VSM-bl	0.541	0.591	0.581
PLSA1	PLSA1-an	<u>0.639</u>	<u>0.650</u>	<u>0.671</u>
	PLSA1-as	0.445	0.482	0.448
	PLSA1-bl	0.622	0.642	0.628
PLSA2	PLSA2-an	<u>0.638</u>	<u>0.651</u>	<u>0.674</u>
	PLSA2-as	0.447	0.485	0.441
	PLSA2-bl	0.619	0.647	0.622
DRM	IDRM	0.669*	0.675*	0.683*
	DDRM	0.685*	0.690*	0.697*
		7.2%	6%	3.4%

Table 5: Recommendation accuracies of different methods for question recommendation. Each underlined value means the best result for each baseline group. ‘*’ means the corresponding improvement over all baselines is statistically significant.

baseline. For the latter, we implemented the “user-word aspect model” presented in [6] (PLSA2) as another baseline. The details of PLSA1 and PLSA2 have been described in section 3.

In order to explore the impact of different user roles on question recommendation, we implemented the different versions of the three groups of baselines. These versions are based on the answerer role, the asker role, or the blended roles. Table 4 shows the labels of all baseline methods. Each version of a baseline is trained on the question set that users access under the corresponding role. Specifically, the blended roles mean all the question that each user answers or asks are simply mixed together as one set.

Based on cross-validation, we selected the best topic number for PLSA1 and PLSA2. The recommendation results of baselines and our DRM are summarized in Table 5, where the best result for each baseline group is underlined and the best result in each data subset is highlighted.

We first compare the performances of different roles in each baseline group. From Table 5, we observe that the answerer role always wins the best result in all baseline groups across data subsets. Especially, the answerer role is obviously better than the asker role over all the recommendation results. When data sets become denser and denser from User-10 to User-20, the effect of the answerer role becomes better and better as we expect. On the contrary, the result of the asker role appears an unexpected decrease in User-20. Furthermore, we examine the recommendation results of blended roles. As we can see, simply mixing the asker role into the answerer role not only fails to improve but worsens recommendation results of answerer role instead. According to above experiments, we conclude that different user roles reflect the different aspects of the user, moreover, there are clear distinctions between their influences on question recommendation. When modeling the user in CQA, we must distinguish the different user roles. When recommending new questions to users, it would be more appropriate to use the answerer role model to represent the candidate users.

Since the recommendation methods based on blended roles do not work well, whether our DRM based method can make full use of user’s dual roles to improve the recommendation result? Tested

on each data subset, our model exhibits good performance, significantly outperforming all baselines. The relative improvement of DDRM over the best baseline result is 7.2% for User-10, 6% for User-15, and 3.4% for User-20. In addition, DDRM is significantly better than IDRM across data sets, which means considering the dependence between uses is more effective to model user roles. This result is also consistent with the above role analysis.

Another interesting result to note is that the PLSA1 which models the user indirectly is almost equivalent to PLSA2 which models the user directly on three data subsets, suggesting that it is feasible to model the user indirectly by combining the topic information questions that he accesses. Additionally, it is clear that all methods based on latent topic analysis (PLSA1, PLSA2, and DRM) always perform better than word-level VSM, which demonstrates that the latent topic based model can be more effective to represent the profile of user. And this result also verifies the conclusion drew in [6]. Moreover, it is part of the reason for that Guo et al. [4] introduced topic-level model to improve heuristic word-level methods.

5.4.2 Parameter Sensitivity

We note that the topic number K is an important parameter in our proposed DRM. Therefore, we are interested in analyzing the sensitivity of the recommendation performance of DRM with respect to the topic number. We tested these two DRM variants with 8 different values of K , which is illustrated in Figure 4. Like previous role analysis, we only present the final results in data subset User-10. The results for other subsets are similar. As we can see from Figure 4, the recommendation accuracy gradually increases when the topic number varies from 10 to 40. Then we observe that the effectiveness of both DRM based recommendation approaches begins to be relatively stable when topic number is more than 40.

6. CONCLUSION & FUTURE WORK

The user in CQA plays two different roles (dual roles) simultaneously, which is different from the user in a general recommendation system. In this paper, we have systematically investigated the

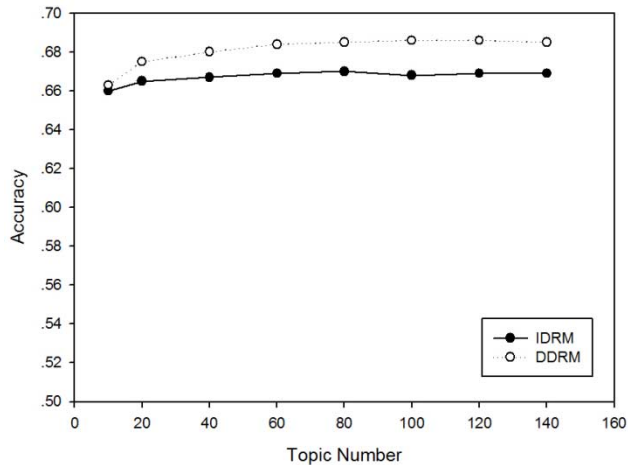


Figure 4: Sensitivity to the topic number of DRM.

distinctions between users' dual roles and how they affect the performance of question recommendation differently. Moreover, in order to represent the user in CQA with dual roles more reasonably and precisely, we proposed a Dual Role Model (DRM) to model the user's different roles. With different independence assumptions, two variants of DRM were achieved, which were independent DRM (IDRM) and dependent DRM (DDRM). Finally, we presented the DRM based approach to question recommendation which can take full advantage of the particularities of users' different roles. Based on a unified probabilistic framework, our DRM based method naturally combines the user relation between the answerer and the asker with the content relevance between the answerer and the question.

Our experiments were carried out on a real-world data crawled from Yahoo! Answers. First, the results of user role analysis showed that there are evident differences between the answerer role and asker role of users in CQA. Comparing the effects of the two roles on question recommendation, we discovered that the answerer role model is more appropriate to represent the candidate users when recommending new questions to users. Additionally, an interesting result was that simply mixing the asker role into the answerer role not only failed to improve but impaired recommendation results of answerer role instead. Furthermore, we compared our DRM based recommendation methods with baseline methods based on a single role or blended roles. Experiment results on three data subsets showed our DRM significantly outperforms all baselines, where the relative improvement of DRM over the best baseline result is 7.2% for User-10, 6% for User-15, and 3.4% for User-20. In addition, DDRM is more effective than IDRM across data subsets, suggesting it is more effective to model users' dual roles. Finally, the parameter sensitivity analysis showed our DRM approach is robust.

There are two interesting future research directions to explore. One of the most interesting directions is to further study how the roles of users will vary over time, and whether that will have influence on question recommendation. The other interesting direction is how to diversify the recommendation results to satisfy users better.

7. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their useful comments. This work is supported by the National Science Foundation of China under Grant No. 61070111.

8. REFERENCES

- [1] L. Rao. Yahoo mail and im users update their status 800 million times a month. TechCrunch, Oct282009. <http://techcrunch.com/2009/10/28/yahoo-mail-and-im-users-update-their-status-800-million-times-a-month/>.
- [2] Hu Wu, Yongji Wang and Xiang Cheng. Incremental Probabilistic Latent Semantic Analysis for automatic question recommendation. In *RecSys '08*, pages 99-106, 2008.
- [3] Damon Horowitz and Sepandar D. Kamvar. The anatomy of a large-scale social search engine. In *WWW'10*, pages 431-440, 2010.
- [4] Jinwen Guo, Shengliang Xu, Shenghua Bao, and Yong Yu. Tapping on the potential of Q&A community by recommending answer providers. In *CIKM'08*, pages 921-930, 2008.
- [5] Gideon Dror, Yehuda Koren, Yoelle Maarek and Idan Szpektor. I want to answer, who has a question? Yahoo! Answers recommender system. In *SIGKDD'11*, pages 1109-1117, 2011.
- [6] Mingcheng Qu, Guang Qiu, Xiaofei He, Cheng Zhang, Hao Wu, Jiajun Bu, and Chun Chen. Probabilistic question recommendation for question answering communities. In *WWW'09*, pages 1229-1230, 2009.
- [7] Alexandrin Popescul, Lyle H. Ungar, David M. Pennock and Steve Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *UAI'01*, pages 437-444, 2001.
- [8] Baichuan Li, Irwin King and Michael R. Lyu. Question routing in community question answering- putting category in its place. In *CIKM'11*, pages 2041-2044, 2011.
- [9] Thomas Hofmann and Jan Puzicha. Latent class models for collaborative filtering. In *IJCAI'99*, pages 688-693. 1999.
- [10] Thomas Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99*, pages 50-57, 1999.
- [11] Luo Si and Rong Jin. Flexible mixture model for collaborative filtering. In *ICML'03*, 2003.
- [12] David M. Blei, Andrew Y. Ng and Michael I. Jordan. Latent dirichlet allocation. In *Journal of Machine Learning Research*, pages 993-1022, 2003.
- [13] Thomas Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *SIGIR'03*, pages 259-266, 2003.
- [14] Tom Chao Zhou, Chin-Yew Lin, Irwin King, Michael R. Lyu, Young-In Song and Yunbo Cao. Learning to suggest questions in online forums. In *AAAI'11*, pages 1298-1303, 2011.
- [15] Qiaoling Liu and Eugene Agichtein. Modeling answerer behavior in collaborative question answering systems. In *ECIR'11*, pages 67-79, 2011.
- [16] Ke Sun, Yunbo Cao, Xinying Song, Young-In Song, Xiaolong Wang and Chin-Yew Lin. Learning to recommend questions based on user rating. In *CIKM'09*, pages 751-758, 2009.
- [17] Nathan N. Liu and Qiang Yang. EigenRank: A ranking-oriented approach to collaborative filtering. In *SIGIR'08*, pages 83-90, 2008.

- [18] Adomavicius G., and Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. In *IEEE Trans. on Knowledge and Data Engineering*, 17(6), 2005.
- [19] P. Kantor, F. Ricci, L. Rokach, and B. Shapira. *Recommender Systems Handbook: A complete guide for research scientists and practitioners*. Springer, 2010.
- [20] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Maching Learning Journal*, Vol. 42, No. 1-2, pages. 177-196, 2001.
- [21] Jiwoon Jeon, W. Bruce Croft, Joon Ho Lee and Soyeon Park. A framework to predict the quality of answers with non-textual features. In *SIGIR'06*, pages 228-235, 2006.
- [22] Jiwoon Jeon, W. Bruce Croft and Joon Ho Lee. Finding semantically similar questions based on their answers. In *SIGIR'05*, pages 617-618, 2005.
- [23] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. In *IEEE Internet Computing*, 07(1):76-80, 2003.
- [24] Jiahui Liu, Peter Dolan, Elin Rønby Pedersen. Personalized news recommendation based on click behavior. In *IUI'10*, pages 31-40, 2010.
- [25] Y. Cao, H. Duan, Chin-Yew Lin, Y. Yu and Hsiao-Wuen Hon. Recommending questions using the MDL-based tree cut model. In *WWW'08*, pages 81-90, 2008.
- [26] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391-407, 1990.
- [27] J. Zhang, L. A. Adamic, E. Bakshy and Mark S. Ackerman. Every-one knows something: Examining knowledge sharing on Yahoo! Answers. In *WWW'08*, pages 665-674, 2008.
- [28] M. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*. vol. 27, pages 313-331, 1997.
- [29] Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Journal of Computer*, 42(8):30-37, 2009.
- [30] Kevin K. Nam, Mark S. Ackerman, Lada A. Adamic. Questions in, knowledge in? A study of Naver's question answering community. In *CHI'09*, pages 779-788, 2009.
- [31] Pawel Jurczyk, Eugene Agichtein. Discovering authorities in question answer communities by using link analysis. In *CIKM'07*, pages 919-922, 2007.