A New Approach to Answerer Recommendation in Community Question Answering Services

Zhenlei Yan and Jie Zhou

State Key Laboratory on Intelligent Technology and Systems Tsinghua National Laboratory for Information Science and Technology (TNList)
Department of Automation, Tsinghua University, Beijing, China, 100084
yanzl07@mails.thu.edu.cn, jzhou@thu.edu.cn

Abstract. Community Question Answering (CQA) service which enables users to ask and answer questions have emerged popular on the web. However, lots of questions usually can't be resolved by appropriate answerers effectively. To address this problem, we present a novel approach to recommend users who are most likely to be able to answer the new question. Differently with previous methods, this approach utilizes the inherent semantic relations among asker-question-answerer simultaneously and perform the Answerer Recommendation task based on tensor factorization. Experimental results on two real-world CQA dataset show that the proposed method is able to recommend appropriate answerers for new questions and outperforms other state-of-the-art approaches.

Keywords: answerer recommendation, tensor factorization, community question answering.

1 Introduction

In recent years, Community Question Answering (CQA) service which enables users to ask and answer questions on the web has been a successful application of Web 2.0. CQA portals such as Yahoo! Answer[1] and Tencent Wenwen[2] which is one of the leading providers for CQA service in China provide an online platform for users to post their questions and share their knowledge by answering others' questions. Thousands of millions of questions have been resolved in these popular CQA portals. For example in Tencent Wenwen, 185,580,969 questions have been resolved until September 30, 2011. Although the CQA service has brought significant benefits for users, there are still several drawbacks in current systems. The most important problem is the effectiveness of solving a new question. Previous survey [7] has shown that lots of new questions can't be resolved effectively (within 24 hours) actually. On the other hand, with the rapidly increasing number of new questions, active users who are experienced in specified domains are not easy to find their interested questions which leads to the low participation rate [4].

[1] http://answers.yahoo.com
[2] http://www.wenwen.com

R. Baeza-Yates et al. (Eds.): ECIR 2012, LNCS 7224, pp. 121–132, 2012.

To address these problems, several approaches have been proposed in both industry and academic area. For example, Aardvark[3] is a social search engine which routes the new question to the person in the asker's extended social network (e.g. Facebook[4], LinkedIn[5], and etc.) who are most likely to be able to answer this question, and the details of Aardvark are introduced in [6]. Recently, a new social network of question-answering called Quora[6] has drawn numerous attention. Users in Quora can follow topics and experts as well as following people in Twitter[7], and then answer the questions of the specified topics or post new questions to experts. These improvements make new questions be resolved more effectively than traditional CQA service.

In this paper, we formally describe this *Answerer Recommendation* problem and present a novel approach to recommend potential answerers for new questions effectively. The proposed approach consists of three stages: (a) Learning the topic distributions of questions; (b) Training the model with tensor factorization; (c) Recommending answerers for new questions. In summary, this paper brings two major contributions: (1) We present an Asker-Topic-Answerer (ATA) model by utilizing tensor model and topic model simultaneously to capture the inherent semantic relations among asker, question and answerer. (2) Based on this model, a new approach with tensor factorization is proposed to perform the Answerer Recommendation task.

The rest of this paper is organized as follows. In Section 2, we present an overview of previous related work. Next, we formally describe the Answerer Recommendation problem in Section 3. In Section 4, we introduce the details of the proposed approach. In Section 5, we compare our method with other baseline and state-of-the-art approaches and investigate the impact of model parameters. Finally, we draw conclusions and discuss the future work in Section 6.

2 Related Work

Community Question Answering (CQA) portals such as Yahoo! Answer and Tencent Wenwen have collected plenty of questions and their answers during the past few years. A lot of research work have been conducted on related areas [10,1,4,11,9].

Recently, several approaches have been proposed in academic area to tackle this Answerer Recommendation problem in CQA service. Zhou [14] present several approaches with language models to represent the expertise of users based on their previous question&answering activities, and then push the new questions to the appropriate users in online forums. Guo [4] focuses on recommending answer providers in order to increase the participation rate of users in CQA service. Liu [8] employs a probabilistic framework to predict best answerers for

[3] http://vark.com/
[4] http://www.facebook.com
[5] http://www.linkedin.com
[6] http://www.quora.com/
[7] http://www.twitter.com

new questions. The framework combines the language model and LDA model to learn users' interests, and considers the authority and activity of users in the predicting process. Li [7] propose a *Question Routing* (QR) framework to perform the similar task which considers the quality of answer's content when predicting appropriate answerers for new questions.

In order to learn the latent topics of questions, topic models based on Latent Dirichlet Allocation (LDA) [2] have been extensively investigated [4]. Differently from previous methods, our proposed method not only consider the topics of questions, but also combines them with tensor model to capture the semantic relation among asker, question and answerer simultaneously. Although tensor factorization method has been applied applied in information retrieval [12], it has not yet been introduced in community question answering.

3 Problem Formalization

Typically, Community Question Answering service consists of users, questions and answers, while users consist of askers and answerers. We define the set of all askers $U = \{u_i\}_{i=1}^{I}$, the set of all questions $Q = \{q_j\}_{j=1}^{J}$ and the set of all answerers $A = \{a_m\}_{m=1}^{M}$. The ternary relation (u, q, a) means that the answerer a has answered the question q which is posted by asker u. We define the set of all these ternary relations $S \subseteq I \times J \times M$. for each question q, we use u_q to represent the corresponding asker. For convenience in the following statement, we define the functions as below,

$$Q(u) = \{q \mid \forall q \in Q, \exists a, s.t. (u, q, a) \in S\}$$
$$A(q) = \{a \mid \forall a \in A, \exists (u_q, q, a) \in S\}$$

$Q(u)$ denotes the set of all questions posted by asker u, and $A(q)$ represents the set of all answerers who answer question q.

For a new question q, the Answerer Recommendation problem is set to predict the best Top-N answerers who are most likely to be able to answer q. Given a predictor $\hat{Y}$, this means that we should predict a score $\hat{y}_{u_q,q,a}$ for each candidate a. Consequently, the Top-N highest scoring answerers for question q can be calculated as follows:

$$Top(q, N) = arg \max_{a \in A}^{N} \hat{y}_{u_q,q,a} \tag{1}$$

Where the superscript N represents the number of answerers to be predicted.

4 The Approach to Answerer Recommendation

4.1 Learning Latent Topics of Questions

In order to capture the semantic relation between users and questions, we need to discover the latent topics of questions first. We employ the Latent Dirichlet Allocation (LDA) model [2] which has been widely applied in IR tasks to learn

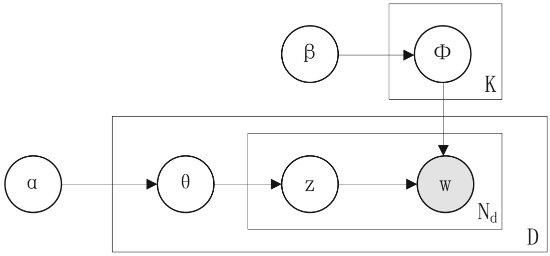


Fig. 1. Graphical model representation of LDA

the latent topics of questions. The graphical model representation of LDA is shown in Figure 1.

Where K is the number of topics, D is the number of documents in corpus and N_d is the number of words in document d; The details are introduced in [2].

Differently from regular documents, the questions including title and body are usually short in community question answering. This will lead to poor performance of LDA model. To address this problem, we construct a new corpus, in which each document consists of a question and the corresponding best answer. Then we train the LDA model on this new corpus with Gibbs Sampling. This approach is reasonable because the best answer which is selected by asker is naturally supposed to have the same topic with the question. We define the set of all topics $T = \{t_k\}_{k=1}^{K}$. After the training process, the topic distribution of each question q has been learned which we denote as $P(q) = \{p(t_k \mid q)\}_{k=1}^{K}$.

4.2 Asker-Topic-Answerer Model with Tensor Factorization

After learning the topic distribution $P(q)$ of question q, we replace the ternary $(u, q, a) \in S$ with a set of $\{(u, t_k, a)\}_{k=1}^{K_q}$. K_q the number of highest probability $p(t_k \mid q)$. This set can be used to capture the semantic relation among u, q and a, because (u, t_k, a) reveals that a has answered the question of u in topic t_k. Then $S = \{(u, t, a)\} \subseteq I \times K \times M$. The set of all different $(u, t) \in S$ is denoted as O_S. Now, $\hat{y}_{u,q,a}$ can be estimated by $\hat{z}_{u,t_k,a}$ as Equation 2.

$$\hat{y}_{u,q,a} = \sum_{k=1}^{K_q} p(t_k \mid q) \cdot \phi(\hat{z}_{u,t_k,a}) \tag{2}$$

Where $\hat{Z}$ is the corresponding predictor for (u, t, a) and $\phi(\cdot)$ is a normalizing function. We define the set of all asker's topic (u, t).

The remaining problem is how to calculate $\hat{z}_{u,t,a}$. In this paper, we estimate $\hat{Z}$ by factorizing (u, t, a) to three low-rank feature matrices which represents askers, topics and answerers respectively, and a core tensor. The predictor $\hat{Z}$ is estimated by multiplying the three feature matrices to the core tensor:

$$\hat{Z} = \hat{C} \times_u \hat{U} \times_t \hat{T} \times_a \hat{A}$$

Where the core tensor $\hat{C}$ and the feature matrices $\hat{U}$, $\hat{T}$ and $\hat{A}$ are the model parameters to be learned. $\times_x$ is the tensor product to multiply a matrix on dimension x with a tensor. These parameters are denoted as $\hat{\theta} := (\hat{C}, \hat{U}, \hat{T}, \hat{A})$ and have the following sizes: $\hat{C} \in \Re^{k_U \times k_T \times k_A}$, $\hat{U} \in \Re^{|U| \times k_U}$, $\hat{T} \in \Re^{|T| \times k_T}$, $\hat{A} \in \Re^{|A| \times k_A}$. k_U, k_T and k_A are the dimensions of the corresponding low-rank approximation. Then given $\hat{\theta}$, $\hat{z}_{u,t,a}$ can be calculated as follows:

$$\hat{z}_{u,t,a} = \sum_{\tilde{u}=1}^{k_U} \sum_{\tilde{t}=1}^{k_T} \sum_{\tilde{a}=1}^{k_A} \hat{c}_{\tilde{u},\tilde{t},\tilde{a}} \cdot \hat{u}_{u,\tilde{u}} \cdot \hat{t}_{t,\tilde{t}} \cdot \hat{a}_{a,\tilde{a}} \tag{3}$$

4.3 Learning Model Parameters

In this section, we employ an optimization criterion similar with [12] to learn the best model parameters $\hat{\theta}$ by maximizing the ranking statistics AUC (area under the ROC-curve) as described in Equation 4 and 5.

$$arg \max_{\hat{\theta}} \sum_{(u,t) \in O_S} AUC(\hat{\theta}, u, t) \tag{4}$$

Where

$$AUC(\hat{\theta}, u, t) = \frac{1}{\mid A^+_{u,t} \mid\mid A^-_{u,t} \mid} \sum_{a^+ \in A^+_{u,t}} \sum_{a^- \in A^-_{u,t}} H_{0.5}(\hat{z}_{u,t,a^+} - \hat{z}_{u,t,a^-}) \tag{5}$$

Where A^+ is the set of positive answerers and A^- is the set of negative answerers which are defined as below:

$$A^+_{u,t} = \{a \mid (u,t) \in O_S \wedge (u,t,a) \in S\}$$
$$A^-_{u,t} = \{a \mid (u,t) \in O_S \wedge (u,t,a) \notin S\}$$

and $H_{0.5}$ is the Heaviside function.

This AUC optimization criterion implies the following pairwise ranking constraint of $\hat{Z}$:

$$\hat{z}_{u,t,a_1} > \hat{z}_{u,t,a_2} \Leftrightarrow a_1 \in A^+_{u,t} \wedge a_2 \in A^-_{u,t}$$

This constraint means that the answerer who has answered the question of u in topic t should rank higher than others who haven't answered the questions.

Then we use the gradient descent algorithm to learn $\hat{\theta}$. The details of learning process are similar with [12].

4.4 Recommending Answerers

Once we have learned the model parameters $\hat{\theta}$, we are able to recommend appropriate answerers for new questions.

For a new question q_{new} posted by asker u, we firstly infer the topic distribution of q_{new}, $P(q_{new}) = \{p(t_k \mid q_{new}), \mid k = 1, \ldots, K\}$. Then, for each topic

t_k, we calculate $\{\hat{z}_{u,t_k,a}\}$, $\forall a \in A$ as Equation 3. Next, we use $\phi(\cdot)$ to normalize $\{\hat{z}_{u,t_k,a}\}$, $k = 1, \ldots, K$ as follows:

$$\phi(\hat{z}_{u,t,a}) = \frac{\hat{z}_{u,t,a}}{max(\{\hat{z}_{u,t,a} \mid \forall a \in A\})} \tag{6}$$

Then, the score $\hat{y}_{u,q,a}$ is calculated as Equation 2. At last, we select the users with Top-N highest $\hat{y}_{u_{q_{new}},q_{new},a}$ as the recommended answerers for q_{new}. The details of our framework approach are shown in Algorithm 1.

Algorithm 1. The Framework for Answerer Recommendation

Input: S_{train}, S_{test}, N, K, k_U, k_T, k_A, δ, γ, γ_C;
Output: The set of recommended answerers A_{rec} for S_{test}
begin
  Train:
  Train the LDA model to learn the topic distribution of questions;
  foreach $(u, q, a) \in S_{train}$ **do**
    Decompose (u, q, a) to $\{(u, t, a)\}$ with the topic distribution $P(q)$;
  end
  Learn $\hat{\theta}$ with tensor factorization as described in Section 4.3;
  Predict:
  foreach $q' \in S_{test}$ **do**
    Apply trained LDA model to infer $P(q')$;
    foreach $a \in A$ **do**
      Calculate $\hat{z}_{u_{q'},t,a}$ as Equation 3;
    end
    Calculate $\hat{y}_{u_{q'},q,a} = \sum_{k=1}^{K} p(t_k \mid q') \cdot \hat{z}_{u_{q'},t_k,a}$;
  end
  Calculate $Top(q', N) = arg \max_{a \in A}^{N} \hat{y}_{u_{q'},q',a}$;
  Add $Top(q', N)$ to the output result A_{rec};
end

5 Experimental Evaluation

In this section, we conduct several experiments to compare the performance of the proposed method with other baseline and state-of-the-art approaches.

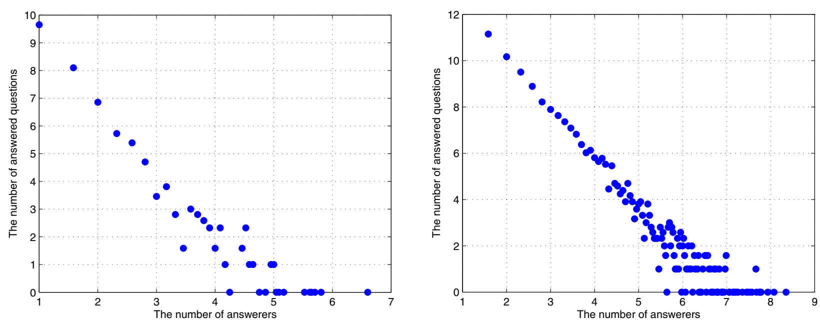
Our experiments are designed to address the following questions:

1. How does the proposed method compare with other published approaches?
2. How does the number of topics K affect the performance of our method?
3. How do the parameters k_p, k_t, and k_a affect the performance of our method?

5.1 Dataset

We collect two real-world CQA datasets from Yahoo! Answer (YA) and Tencent Wenwen (TW) respectively. The YA dataset is crawled from February 21, 2010 to April 3, 2011 in the category of Internet, while the TW dataset is crawled from October 2010 to November 2010 in four categories, i.e. Computer/IT, Health/Medical, Education/Science and Social/Culture. The stopwords in question and answer content are removed for both YA and TW dataset. Notice that for TW dataset which is in Chinese we should perform Chinese word segmentation firstly.

In order to investigate the characteristics of the crawled datasets, Figure 2(a) shows the degree distribution of the number of questions answered by the same user of YA dataset, and Figure 2(b) shows the degree distribution of TW dataset. We can see that both of them follow a Power Law distribution which is a typical characteristic of social medias.



(a) YA dataset (b) TW dataset

Fig. 2. Degree distribution of answerer-question

After that, we use a p-core[8] - for YA 2-core and for TW 3-core. The statistics of the final experimental datasets are shown in Table 1.

Table 1. Statistics of the experimental datasets

Dataset	**| U |**	**| Q |**	**| A |**	**| S |**
YA	1,811	1,884	1,414	5,358
TW	18,149	19,586	6,741	60,590

[8] The p-core of S is the largest subset of S with the property that every user, every item and every tag has to occur in at least p posts.

5.2 Evaluation Methodology

We use four well known metrics to evaluate the performance of methods, namely Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), Precision at rank N (P@N), and Recall at rank N (R@N), which are widely used in information retrieval [5].

In our experiments, we split the whole dataset S into training and test set as follows. For each asker u in dataset S, we select one of questions posted by u at random to compose the test set S_{test}, and then the groundtruth is the answerers who have answered these questions. Further more, the training set S_{train} consists of the remaining question posts of u and the related answerers. Consequently, we train the proposed method on S_{train} and predict Top-N best answerers for each question in S_{test}.

5.3 Experimental Parameters

At the first stage of the proposed framework, we learn the topic distribution of questions by LDA[9]. The number of topics K is set to 500. The other parameters of LDA model are: Dirichlet priors $\alpha = 0.1$ and $\beta = 0.1$; iteration number $iter_{LDA} = 1000$. Next, to learn the best parameters of ATA model, we set the dimension of three low-rank feature matrices, $k_p = k_t = k_a = k_{dim}$ and $k_{dim} = 32$. The corresponding method is called $ATA - 32$. The other parameters of gradient descent algorithm are: learning rate $\delta = 0.1$ and iteration number $iter_{ATA} = 100$. The model parameters $\hat{\theta}$ are initialized with the random values drawn from the Gaussian distribution $N(0, 0.01)$. K_q is set to 5 for all questions.

5.4 Results and Discussion

Topics of Questions Discovered by LDA. Table 2 shows several typical topics of questions discovered from YA and TW dataset by LDA. For each topic, the Top-10 words with highest probability are selected. From Table 2, we can see that for YA dataset topic 6 focuses on *blog* and topic 17 is related to *online videos*. For TW dataset, we first translate the words in Chinese into English firstly for convenience. Topic 16 is related to *human body*, while topic 47 is about the nations in the world.

Performance Comparison of Methods. In this section, we compare the results of the proposed method ATA with the following methods:

- **QLL**: the query likelihood language (QLL) model is the baseline method which is proposed by [13].
- **QLDA**: Liu [8] propose this method by linearly combining LDA model and QLL model together to predict the best answerers.
- **QLUC**: This method is also proposed by Liu [8] which considers answerer's activity and authority into the QLDA model.

[9] http://gibbslda.sourceforge.net/

Table 2. Topics discovered on YA and TW dataset

YA-Topic-6	YA-Topic-17	TW-Topic-16	TW-Topic-47
blog 0.0548	watch 0.0420	leg 0.0199	France 0.0196
create 0.0164	movies 0.0333	motion 0.0197	USA 0.0180
blogging 0.0142	movie 0.0183	muscle 0.0193	UK 0.0172
blogger 0.0130	quality 0.0138	sports 0.0136	world 0.0144
site 0.0105	tv 0.0130	body 0.0125	Europe 0.0143
page 0.0083	online 0.0088	shank 0.0096	nation 0.0140
share 0.0081	high 0.0063	hands 0.0095	Africa 0.0101
blogs 0.0081	watching 0.0054	thigh 0.0091	airport 0.0082
wordpress 0.0075	dvd 0.0042	abdomen 0.0088	Spain 0.0076
free 0.0071	hulu 0.0030	exercise 0.0086	Italy 0.0074

– **BQAE**: Li [7] present an approach which considers the answering quality and availability estimation when recommending answerers.
– **SQAE**: this method is an improved version of BQAE, and the details are introduced in Li [7].

Table 3. Performance comparison on YA dataset

Metrics	QLL	QLDA	QLUC	BQAE	SQAE	ATA-32
MRR	0.0316	0.0438	0.0652	0.0882	0.1078	**0.1218**
Improve	285.4%	178.1%	86.8%	38.1%	13.0%	
MAP	0.0055	0.0061	0.0087	0.0126	0.0162	**0.0193**
Improve	251.0%	216.4%	121.8%	53.2%	19.1%	
P@N	0.0032	0.0037	0.0050	0.0061	0.0070	**0.0081**
Improve	151.1%	119.0%	62.0%	32.8%	15.7%	
R@N	0.1073	0.1238	0.1560	0.1925	0.2139	**0.2386**
Improve	122.4%	92.7%	52.9%	23.9%	11.5%	

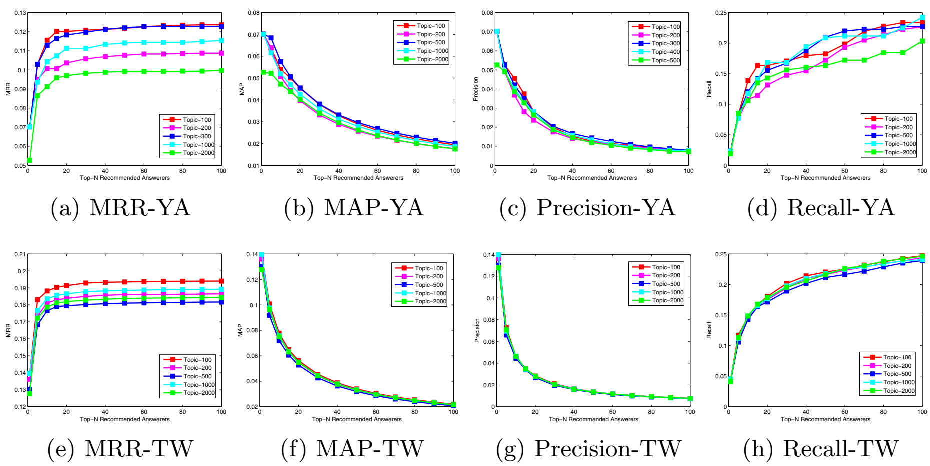
Table3 and Table 4 present the performance comparison of the proposed method and other published approaches on YA dataset and TW dataset respectively. From Table 3, we can see that the proposed method ATA-32 outperforms other approaches consistently in all metrics, MRR, MAP, P@N and R@N (N = 100) on YA dataset. From Table 4, we can draw the same conclusion that our method outperforms other approaches consistently in all metrics.

Impact of the Number of Topics. In order to investigate the impact of the number of topics K, we run our method with $K = \{100, 200, 500, 1000, 2000\}$ and $k_{dim} = 64$ respectively. Figure 3 shows the results of performance comparison in all metrics on YA and TW dataset. Figure 3(a)-(d) show the results in

Table 4. Performance comparison on TW dataset

Metrics	QLL	QLDA	QLUC	BQAE	SQAE	ATA-32
MRR Improve	0.0828 124.3%	0.0910 104.1%	0.1052 76.5%	0.1390 33.6%	0.1481 25.4%	**0.1857**
MAP Improve	0.0126 58.7%	0.0138 44.9%	0.0150 33.3%	0.0164 22.0%	0.0185 8.2%	**0.0200**
P@N Improve	0.0062 17.7%	0.0065 12.3%	0.0067 9.0%	0.0070 4.3%	0.0071 2.8%	**0.0073**
R@N Improve	0.1974 18.0%	0.2071 12.4%	0.2120 9.8%	0.2230 4.4%	0.2265 2.8%	**0.2329**

four metrics, i.e. MRR, MAP, Precision and Recall on YA dataset respectively and Figure 3(e)-(f) show the results on TW dataset. From Figure 3, we can see that there is no obvious relations between K and the performance of the corresponding method. However, a larger number of topics K will even lead to hurt the performance, e.g. $K = 2000$. On the contrary, the method with $K = 100$ outperforms other methods in many metrics such as MRR and Recall.

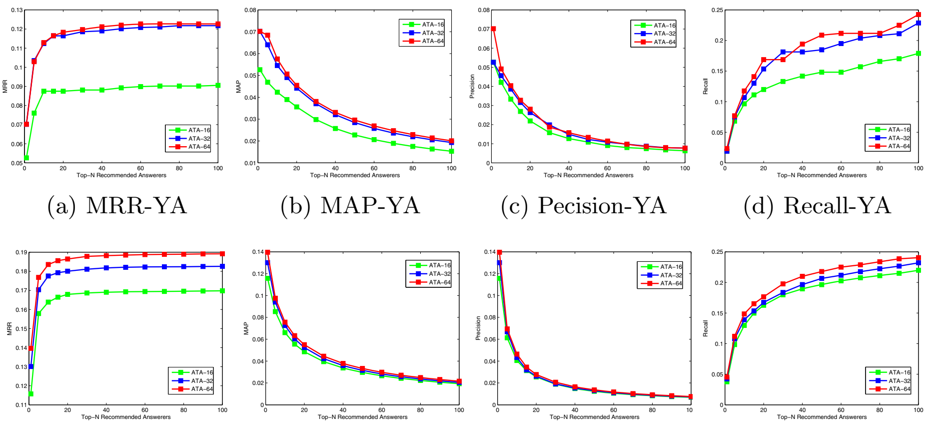


(a) MRR-YA (b) MAP-YA (c) Precision-YA (d) Recall-YA

(e) MRR-TW (f) MAP-TW (g) Precision-TW (h) Recall-TW

Fig. 3. Impact of the number of topics

Impact of Parameters k_U, k_T and k_A. To resolve the last question mentioned above, we investigate the impact of the parameters k_U, k_T and k_A. We run ATA method with $k_{dim} = \{16, 32, 64\}$ respectively. Figure 4 shows the results of performance comparison of methods in all metrics on YA and TW dataset. Figure 4(a)-(d) show the results in four metrics, i.e. MRR, MAP, Precision and Recall on YA dataset respectively and Figure 4(e)-(f) show the results on TW dataset. From Figure 4, we can see that as k_{dim} increasing from 16 to 64, the performance of corresponding method is also improved. On the other hand, we should spend more time to train the corresponding model as k_{dim} increasing. In practice, we should strike a balance between the performance and complexity of the method.



(a) MRR-YA (b) MAP-YA (c) Pecision-YA (d) Recall-YA

(e) MRR-TW (f) MAP-TW (g) Precision-TW (h) Recall-TW

Fig. 4. Impact of parameters k_U,k_T and k_A

6 Conclusion

In this paper, we formally describe the Answerer Recommendation problem and present a novel approach to recommend appropriate answerers for new questions effectively. The proposed approach firstly learns the topic distributions of questions by LDA model, and then utilizes the 3-order tensor to model the semantic relation among asker, question and answerer. Next, we employ tensor factorization to learn the best model parameters by maximizing AUC area. Finally, we are able to predict the Top-N answerers for new questions. We conduct several experiments on two real-world datasets which are crawled from Yahoo! Answer and Tencent Wenwen. The results show that the proposed method outperforms other related approaches, and then we investigate that how the parameters T, k_U, k_T and k_A affect the performance of the proposed method.

In the future, we plan to investigate other topic models which are more suitable for CQA than LDA to learn the topic distribution of questions. On the other hand, since the timestamp information of users' answering activities is ignored in current model, we want to consider the information of time to improve the performance of the proposed method.

Acknowledgments. The authors would like to thank Feng Jiao and Xiang He for the experimental data and helpful comments. Part of this work was done while Zhenlei Yan was on a summer internship at Tencent. This work was supported by the Natural Science Foundation of China under Grant 60721003, 60875017, 61020106004 and 61021063. This work was supported by the Science & Technology Support Program of China under Grant 2009BAH40B03.

References

1. Adamic, L.A., Zhang, J., Bakshy, E., Ackerman, M.S.: Knowledge sharing and yahoo answers: everyone knows something. In: WWW, pp. 665–674. ACM (2008)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. The Journal of Machine Learning Research 3, 993–1022 (2003)
3. Cormack, G.V., Clarke, C.L.A., Buttcher, S.: Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In: SIGIR, pp. 758–759. ACM (2009)
4. Guo, J., Xu, S., Bao, S., Yu, Y.: Tapping on the potential of q&a community by recommending answer providers. In: CIKM, pp. 921–930. ACM (2008)
5. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS) 22(1), 5–53 (2004)
6. Horowitz, D., Kamvar, S.D.: The anatomy of a large-scale social search engine. In: WWW, pp. 431–440. ACM (2010)
7. Li, B., King, I.: Routing questions to appropriate answerers in community question answering services. In: CIKM, pp. 1585–1588. ACM (2010)
8. Liu, M., Liu, Y., Yang, Q.: Predicting Best Answerers for New Questions in Community Question Answering. In: Chen, L., Tang, C., Yang, J., Gao, Y. (eds.) WAIM 2010. LNCS, vol. 6184, pp. 127–138. Springer, Heidelberg (2010)
9. Gu, Q., Zhou, J.: Local relevance weighted maximum margin criterion for text classification. In: SDM, pp. 1135–1146 (2009)
10. Liu, X., Croft, W.B., Koll, M.: Finding experts in community-based question-answering services. In: CIKM, pp. 315–316. ACM (2005)
11. Qu, M., Qiu, G., He, X., Zhang, C., Wu, H., Bu, J., Chen, C.: Probabilistic question recommendation for question answering communities. In: WWW, pp. 1229–1230. ACM (2009)
12. Rendle, S., Balby Marinho, L., Nanopoulos, A., Schmidt-Thieme, L.: Learning optimal ranking with tensor factorization for tag recommendation. In: SIGKDD, pp. 727–736. ACM (2009)
13. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to information retrieval. ACM Transactions on Information Systems (TOIS) 22(2), 214 (2004)
14. Zhou, Y., Cong, G., Cui, B., Jensen, C.S., Yao, J.: Routing questions to the right users in online communities. In: IEEE International Conference on Data Engineering, pp. 700–711. IEEE (2009)