Neurocomputing 320 (2018) 25–34

Contents lists available at ScienceDirect

Neurocomputing

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

Exploiting cross-source knowledge for warming up community question answering services

Yao Wan[a,b], Guandong Xu[c,*], Liang Chen[d], Zhou Zhao[a], Jian Wu[a,b]

[a] College of Computer Science and Technology, Zhejiang University, Hangzhou, China
[b] Realdoctor Artificial Intelligence Research Center of Zhejiang University, Hangzhou, China
[c] Advanced Analytics Institute, University of Technology, Sydney, Australia
[d] School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China

ARTICLE INFO

Article history:
Received 30 June 2017
Revised 8 January 2018
Accepted 6 August 2018
Available online 11 September 2018

Communicated by Dr. Tie-Yan Liu

Keywords:
CQA
Cross-source
Cold-start
Multi-view learning

ABSTRACT

Community Question Answering (CQA) services such as Yahoo! Answers, Quora and StackOverflow are collaborative platforms where users can share and exchange their knowledge explicitly by asking and answering questions. One essential task in CQA is learning topical expertise of users, which may benefit many applications such as question routing and best answers identification. One limitation of existing related works is that they only consider the warm-start users who have posted many questions or answers, while ignoring cold-start users who have few posts. In this paper, we aim to exploit knowledge from cross sources such as GitHub and StackOverflow to build up the richer views of expertise for better CQA. Inspired by the idea of Bayesian co-training, we propose a topical expertise model from the perspective of multi-view learning. Specifically, we incorporate the consistency existing among multiple views into a unified probabilistic graphic model. Comprehensive experiments on two real-world datasets demonstrate the performance of our proposed model with the comparison of some state-of-the-art ones.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Community Question Answering (CQA) services such as Yahoo! Answers [1], Quora [2] and StackOverflow [3] are collaborative platforms where users can share and exchange their knowledge explicitly by asking and answering questions. For CQA services, it is of critical importance to tap the expertise of each user and make each question dispatched to an appropriate user quickly and properly, which may promote the evolving of QA communities. One essential task to achieve this goal is to learn the topical expertise of users, which may also benefit many applications such as question routing and identification of best answers.

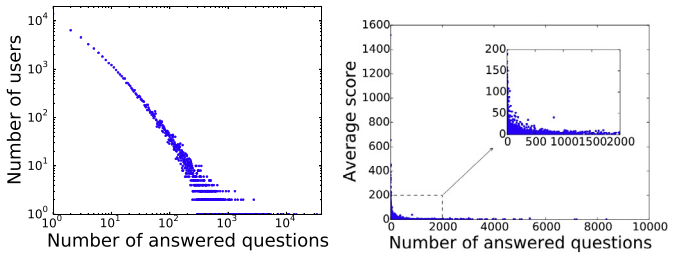
Existing approaches [4–7] mainly rely on their past question-answering activities to build an expertise scoring model, upon which to recommend experts for answering the questions. One limitation of such approaches is that they mainly bias to warm-start users who have posted many questions and answers, while ignoring cold-start users who have limited posts. However, statistics has justified that the users with few posts also contributed certain valuable knowledge and answers to QA communities. Taking StackOverflow as an example, this fact can be illustrated by Fig. 1. From Fig. 1a, we can see that the participation of most users in question-answering activities falls into the long tail part of the power-law curve. This indicates that majority of users only answer very few questions. While considering the thumbs up/downs voted by the community as quality score for users on answering questions, we observe that CQA systems enjoy great benefits contributed by the cold-start users from Fig. 1b. On the other hand, the user expertise might be reflected by multiple aspects. For example, the capability of software developers consists of questions answered, projects completed, and codes written, etc. Apparently the cross-source knowledge is exact the indicator of such capability.

Our paper mainly focuses on capturing the full views of user topical expertise especially for cold-start users via leveraging cross sources. Our intuition lies in the fact that although cold-start users expose little expertise evidence in CQA services, they might leave footprints on other social media websites. As a survey conducted in US shows, 52% of online adults use two or more social media sites such as Facebook, Twitter, MySpace, or LinkedIn, to stay in touch with friends, family members, and business partners [1]. We believe that a user's expertise should be measured from many angles, for

* Corresponding author.
E-mail address: guandong.xu@uts.edu.au (G. Xu).

[1] According Paw Research Internet Project's Social Media Update 2014: http://www.pewinternet.org/.

https://doi.org/10.1016/j.neucom.2018.08.012
0925-2312/© 2018 Elsevier B.V. All rights reserved.



(a) Cold-start property (b) Performance

Fig. 1. Cold-start users in StackOverflow.

example, we can learn one's QA ability and programming ability from StackOverflow and GitHub, respectively. Expertise learnt from these two data sources are complementary. For cold-start users, when a user leaves little footprint in one source, expertise from another source might complete it. In our paper, one's ability learnt from a specific data source is called *view*, our problem can be defined as a multi-view learning problem.

However, integrating of multiple views to learn topical expertise of users in CQA is non-trivial. The first challenge lies in how to fuse users' heterogeneous features from multiple views effectively. One naive approach is to concatenate the feature spaces generated from different sources into a unified feature space. Thereby, traditional machine learning models can be further applied. For example, in [8], Xiao et, al. exploit social media data to boost the performance of cold-start users via social login, in this paper the authors just enrich the information of users by extracting more features. However such approach overlook the interrelations between aspects from different sources, instead simply stacking various views. In fact even different aspects revealed from various data sources are distributed in different feature spaces, they intend to model the same user, resulting in the same characteristics in nature. Such observation should be used to jointly learn user expertise aspects. Another challenge we are facing is on incorporating multi-view co-regularization into a traditional probabilistic generative model and forming a unified framework.

In our paper, we propose a topical expertise model from the perspective of multi-view learning, borrowing the idea from Bayesian co-training [9]. We incorporate the consistency existing among multiple views into a unified probabilistic graphic model. Fig. 2 shows an overview of our proposed framework. We firstly utilize the email matching as a bridge to identify users from different sources such as StackOverflow and GitHub. Thus their profiles containing information from different data sources are generated. Modeling on users' activities from multiple sources, relevant experts are returned with descending order for a given question. Right part of this figure gives a sample project with description and stars in GitHub and a sample QA post with tags and votes in StackOverflow. The main contributions of this paper can be summarized as follows:

- To integrate the multi-view consistency existing among multiple sources, we propose a Bayesian undirected graphical model for co-training.
- Then, we integrate the Bayesian co-training model and the topical expertise model as a unified model and devise an approximate method for inferring the model.
- Finally, we validate our proposed model on two real-world datasets, e.g., StackOverflow and GitHub. Comprehensive experiments show the effectiveness of our model when compared with some baseline models.

The remainder of this paper is organized as follows. Section 2 highlights some works related to this paper. Section 3 introduces some notations and formulates our problem mathematically. Section 4 elaborates our proposed topic expertise model with multiple views In Section 5 we describe the datasets firstly and then present the experimental results and parameter analysis. We conclude this paper and propose some future research directions in Section 6.

2. Related work

In this section, we briefly review some related work of our problem from three perspectives in the literature, including mining in CQA especially for expert finding, cold-start recommendation and multi-view learning.

Expert finding in CQA. Our task is built on community question answering site and researchers have studied CQA from many perspectives. One perspective focuses on expert finding. Generally, the main approaches fro expert finding can be categorized into two groups: the authority-oriented approaches and the topic-oriented approaches. The authority-oriented expert finding methods are based on link analysis of the past question-answering activities of the users in CQA systems. Bouguessa et al. [4] discover the experts based on the number of best answers provided by users, which is an in-degree-based method. Zhu et al. [10] select experts based on the authority of the users on the relevant categories of the questions. Jurczyk et al. [11] propose a HITS [12] based method to estimate the ranking score of the users based on question-answering activity graphs. Zhang et al. [6] propose an expertise ranking method and evaluated link algorithms for specific domains. The topic-oriented expert finding methods are based on latent topic modeling techniques. Xu et al. [13] propose a dual role model that jointly represents the roles of answerers and askers using the generative topic model. Liu et al. [14] propose a language model to predict the best answerer. Guo et al. [15] and Zhou et al. [7] devise the topic sensitive model to build the latent user model for expert finding. Liu et al. [5] model both topics and expertise of the users in CQA for expert finding. Saptarshi et al. [16] utilize the crowdsourcing techniques to find the topic experts in microblogs. Fatemeh et al. [17] incorporate the topic modeling techniques to estimate the expertise of the users. Another research perspective on community question answering sites is quality prediction including answer quality prediction and question quality prediction. Since the methods mentioned above are based on the history data, it may suffer from the cold-start problem. For expert finding, it is insufficient to estimate the expertise of users from only one data source (view), because some users may be inactive in one site but active in other site. Our work try to alleviate the cold start users via learning from multiple views.

Cold-start recommendation. Recently, the cold start problem in recommender systems has attracted a lot of attention and several approaches have been proposed to solve this problem [18–21]. Lin et al. [18] extract the information of items from Twitter to overcome the difficulty of cold-start recommendation. Park et al. [19] propose a latent regression model that leverages the available attributes of items and users to enrich the information. Yin et al. [20] propose a random walk based method to choose the right cold-start items for users. Purushothas et al. [21] utilize both textual information of items and social relations of users to user-item recommendation. However, the cold-start recommendation techniques cannot be applied to the scenario of our expert finding problem seamlessly.

Multi-view learning. The basic idea of multiple view learning is making use of the consistency among different views to achieve better performance. One of the earliest works on multi-view learning is co-training algorithm [22], which uses one view's predictor to enlarge the training set for other views. Some improvements of co-training algorithm are also proposed [9,23]. Yu et al. [9] propose

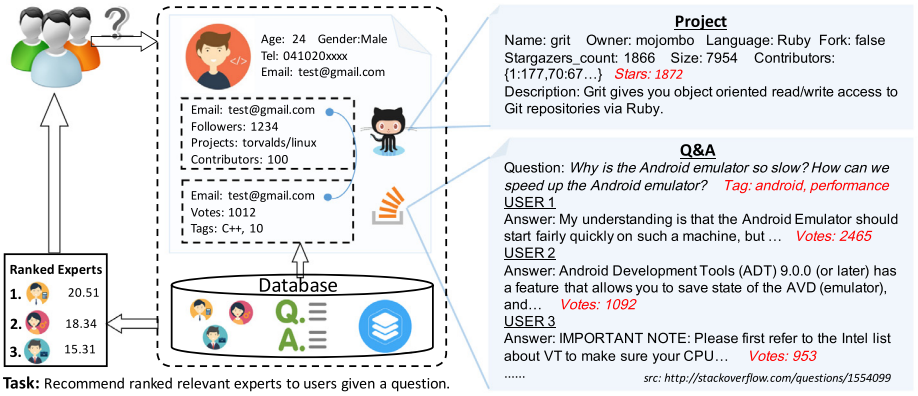


Fig. 2. Illustration of our proposed scheme. We first integrate users from different data sources such StackOverflow and GitHub, according to their email addresses, and their profiles containing information from those two data sources are generated. Modeling on users' activities from multiple sources, relevant users are returned with descending order of expertise for a given question. Right part of this figure gives a sample project with description and stars in GitHub and a sample QA post with tags and votes in StackOverflow.

a graphical model for co-training based on the assumption that multiple views are conditional independent, it illustrates the co-regularized multi-view learning from the perspective of bayesian learning. Other methods are based on co-regularization framework. Sindhwani et al. [24] propose a learning framework for multi-view regularization. SVM-2K [25] is a method which uses kernels for two views learning. Sindhwani et al. [26] construct a single Reproducing Kernel Hilbert Spaces (RKHSs) with a data-dependent "co-regularization" norm that reduces multiple view learning to standard supervised learning. Chen et al. [27] present a large-margin learning framework to discover a predictive latent subspace representation shared by multiple views.

Different from the conventional literatures, our work aims to alleviate the cold-start problem of expert finding in CQA through multi-view learning. Based on the topic-expertise model (TEM) [5] and borrowing the idea from Bayesian co-training [9], we integrate expertise learning from multiple views into an uniform Bayesian network via introducing a latent variable.

3. Problem formulation

To formulate our problem, we first introduce some concepts and declare some notations in this section.

Topical interest. We use *topical interest* to refer to user preference for specific topics in community question answer. For example, some users prefer to "Java", while others are more interested in "algorithm".

Topical expertise. On specific topics, different users have different topical expertise. For example, a user may be a guru for the "Java" topic but a novice for "Python". We use *topical expertise* to refer to their level of expertise on specific topics. For a given user, we can formalize a $<$*user, topic, expertise*$>$ triplet when considering topic interest and expertise simultaneously.

Multi-view learning. Suppose we have U users and $V \geq 2$ data sources. Let $U = \{u_1, u_2, \ldots, u_U\}$ be the set of users. Let $P^{(v)} = \{p_1^{(v)}, p_2^{(v)}, \ldots, p_{N_u^v}^{(v)}\}$ be the posts set of view v. Let $f^{(v)} = \{f_1^{(v)}, f_2^{(v)}, \ldots, f_F^{(v)}\}$ be the scores set of view v, where F is the number of scores in view v. Let $< u_i^{(v)}, p_{i1}^{(1)}, p_{i2}^{(1)}, \ldots, p_{iN_u^1}^{(1)}, p_{i1}^{(2)}, p_{i2}^{(2)}, \ldots, p_{iN_u^2}^{(2)}, \ldots, p_{i1}^{(V)}, p_{i2}^{(V)}, \ldots, p_{iN_u^v}^{(V)} >$ denote the map between users and posts from multiple views. Let $< p_i^{(v)}, s_i >$ be the score map of post i. Let $T = \{t_1, t_2, \ldots, t_T\}$ denote the tags set if one view contains tags information. Multi-view learning is a machine learning technique that can learn from different views simultaneously. When $V = 1$, it boils down to single view learning.

Expert user recommendation. Given the historical activities of users in multiple data sources. Our goal is to learn the topical expertise of each user from the perspective of multi-view learning. Then we apply the user-topic-expertise tuple for expert user recommendation.

The set of notations and description of parameters used in this paper are shown in Table 1.

Table 1
Summary of notations and descriptions.

Notations	Descriptions
U	Number of users
V	Number of data sources
T	Number of unique tags
O	Number of unique words
N_u^v	Number of Q&A posts of user u in view v
L_{un}^v	Number of words in user u's n-th post in view v
P_{un}^v	Number of tags in user u's n-th post in view v
K	Number of topics
E^v	Number of expertise levels view v
μ	Mean of Gaussian distribution
Σ	Precision of Gaussian distribution
w, t, v, e, z	Label for word, tag, vote, expertise, topic
W, T, V, E, Z	Vector for words, tags, votes, expertise, topics
G	Normalization term of the regularizer
m	Number of views
θ	User specific topic distribution
$\mathcal{N}(\mu_e, \sum_e)$	Expertise specific vote distribution
ψ	Topic specific tag distribution
φ	Topic specific word distribution
ϕ	User topical expertise distribution
$\alpha, \beta, \eta, \gamma$	Dirichlet priors
$\alpha_0, \beta_0, \mu_0, k_0$	Normal-Gamma parameters
$\mathcal{NG}(\alpha_0, \beta_0, \mu_0, k_0)$	Normal-Gamma distribution

4. Topical expertise model with multiple views

In this section we elaborate our proposed topical expertise model with multiple views (MultiTEM) including learning and estimating the parameters. We only consider two views in this paper for simplicity, and the idea can be easily extended to the case of more than two views in a similar way.

4.1. Model

In our model user "topical interest" z represents topics the user interested in. It can be represented as a topic distribution in

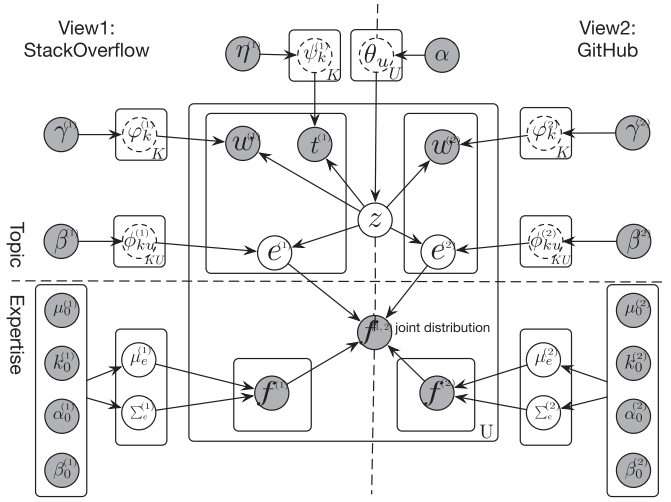


Fig. 3. The plate notation of our proposed MultiTEM model. Dashed variables will be collapsed out in Gibbs sampling.

topical models. In other hand, user "topical expertise" *e* is the level of knowledge and ability of a user *u* under a topic *z*. It is apparent that one user may show different topical interests and different expertise from different views. For instance, one user may post few posts about *Java* in StackOverflow as a newcomer, while he/she may contribute many *Java* projects of high quality in GitHub. So it's one-sided to evaluate the expertise of users from single view. Based on this intuition, we try to take account these two views simultaneously based on a hypothesis consistency.

Hypothesis consistency. We assume that one user should have similar expertise on similar topic in different views. For example, if one user's expertise on *Java* in StackOverflow is level five, the expertise on *Java* drawn from GitHub should also be close to level five.

We model "topical interest" and "topical expertise" from cross sources separately and integrate them with a hypothesis consistency in a unified probabilistic graph model. The plate notation of our proposed probabilistic graph model (MultiTEM) is shown in Fig. 3.

4.1.1. Model topical interest

Similar to Latent Dirichlet Allocation (LDA) model, it's easy to think of introducing two independent latent variables when the corpora are from two different sources. However, this will make it difficult to incorporate the hypothesis consistency, since the two latent variables are not in same space. In our condition, we assume that corpora from multiple views share one common latent variable *z*, which is drawn from multinomial distribution θ. In other words, we enrich the textual information of users by considering description of their posts and projects, without increasing the complexity of models. For posts from StackOverflow, each one is composed of words and tags. For projects from GitHub, each one is only composed of words. Those words and tags are also drawn from multinomial distributions with corresponding Dirichlet priors.

4.1.2. Model topical expertise

To model topical expertise, we assume that there exist $E^{(1)}$ expertise levels in StackOverflow and $E^{(2)}$ expertise levels in GitHub, and each with a Gaussian distribution on vote/stars scores. The reason why we choose Gaussian distribution is that it is with a high range of scores, and the expertise level can be reflected by looking at mean of its corresponding Gaussian distribution. Specifically, a high expertise level is often associated with high vote scores which can be modeled by a Gaussian distribution with high mean. On the contrary, a low expertise level is with a Gaussian distribution with low mean. To incorporate the consistency, we sample the expertise score of votes and stars simultaneously with a regularization. Particularly, given a pair of post and project for a user, if their topics are same, we sample their votes and stars simultaneously according to a joint probability distribution with regularization, otherwise, we sample them independently.

Since the topic distribution of users is learnt from a hierarchical Bayesian network, the challenge of our proposed model lies on how to incorporate the regularization term into a Bayesian network and formalized a unified model. In [9], Yu et.al. propose a Bayesian co-training model which illustrate multi-view co-regularization from a perspective of probabilistic graph model. Borrowing the idea from [9], we define the joint distribution of expertise scores of votes f_1 and stars f_2, taking the consensus among multiple views into consideration.

$$p\left(f^{(1)}, f^{(2)}\right) = p\left(f^{(1)}\right) \cdot p\left(f^{(2)}\right) \cdot \frac{1}{G} exp\left\{ -\frac{1}{2} \sum_{j<k}^{m} \frac{\left\| \bar{f^{(j)}} - \bar{f^{(k)}} \right\|^2}{\delta_j^2 + \delta_k^2} \right\}, \tag{1}$$

where $\bar{f^{(1)}}$ and $\bar{f^{(2)}}$ are normalized $f^{(1)}$ and $f^{(2)}$ which represents expertise votes and stars, respectively, $f^{(1)} \sim \mathcal{N}(\mu_e^{(1)}, \sum_e^{(1)})$ and $f^{(2)} \sim \mathcal{N}(\mu_e^{(2)}, \sum_e^{(2)})$; the third multiplier is a penalty term which constraints that expertise scores from different views need to agree with each other (inversely weighted by the sum of the corresponding variances δ_j^2 and δ_k^2), *G* is the normalization term and *m* is the number of views. When the regularizer term is removed, this joint distribution can be used to sample expertise from different topics, ignoring the consistency among same topics. Note that on an extreme condition that $\left\| \bar{f^{(j)}} - \bar{f^{(k)}} \right\|^2$ is large and $\delta_j^2 + \delta_k^2$ is small, calculating the regularizer term may result in numerical exploding, thus we set the regularizer term to be zero on this condition.

4.1.3. Generative process

To model user topical interest and expertise simultaneously, we assume each user *u* has an expertise level distribution on each topic *z*, denoted as $\phi_{z,u}$. In this case, if this user is an expert in topic *z*, the probability proportions $\phi_{z,u}$ will have high values for expertise levels in different views which correspond to Gaussian distributions with high mean.

For each post in StackOverflow, we observe its vote, multiple words and tags. We assume that each post has latent variables $e^{(1)}$ and *z*, which denote the expertise and topic of this post, respectively. Similarly, for each project in GitHub, we observe its stars count and multiple words. We assume that project has latent variables $e^{(2)}$ and *z*, which denote the expertise and topic of this project, respectively. For each post of a given user u_i in StackOverflow, topics are generated from a user specific topic distribution θ_u and its expertise is generated from the user topical expertise distribution $\phi_{z,u}$. For each topic *z*, in StackOverflow, words are generated from a topic specific word distribution ϕ_z, tags are generated from a topic specific tag distribution ψ_z, in GitHub, words are generated from a topic specific word distribution ϕ_z. Note that we assume tags of answers are the same with the corresponding question in StackOverflow. For each expertise *e*, votes are generated from an expertise specific Gaussian distribution $\mathcal{N}(\mu_e, \sum_e)$ with Normal-Gamma distribution priors.

The generative process of Q&A posts of users can be summarized as follows:

- For the *u*-th user, $(u = 1, 2, \ldots, U)$
  - Draw a user specific topic distribution $\theta \sim Dir(\alpha)$
- For the *u*-th user

- Draw a user topical expertise distribution for posts $\phi_k^{(1)} \sim Dir(\beta^{(1)})$
- Draw a user topical expertise distribution for projects $\phi_k^{(2)} \sim Dir(\beta^{(2)})$

• For the k-th topic, $(k = 1, 2, \ldots, K)$
- Draw a topic specific word distribution for posts $\varphi^{(1)} \sim Dir(\gamma^{(1)})$
- Draw a topic specific tag distribution for posts $\psi^{(1)} \sim Dir(\eta^{(1)})$
- Draw a topic specific word distribution for projects $\varphi^{(2)} \sim Dir(\gamma^{(2)})$

• For the e-th expertise, $(e = 1, 2, \ldots, E^{(1)})$
- Draw an expertise specific vote distribution for posts $\mathcal{N}(\mu_e^{(1)}, \sum_e^{(1)}) \sim \mathcal{NG}(\alpha_0^{(1)}, \beta_0^{(1)}, \mu_0^{(1)}, k_0^{(1)})$

• For the e-th expertise, $(e = 1, 2, \ldots, E^{(2)})$
- Draw an expertise specific stars distribution for projects $\mathcal{N}(\mu_e^{(2)}, \sum_e^{(2)}) \sim \mathcal{NG}(\alpha_0^{(2)}, \beta_0^{(2)}, \mu_0^{(2)}, k_0^{(2)})$

• For the u-th user $(u = 1, 2, \ldots, U)$
- For the n-th post $(n = 1, 2, \ldots, N_u^{(1)})$
  * Draw topic $z \sim Multi(\theta_u)$
  * Draw word $w \sim Multi(\varphi_z^{(1)})$
  * Draw tags $t \sim Multi(\psi_z^{(1)})$
- For the m-th project $(m = 1, 2, \ldots, N_u^{(2)})$
  * Draw topic $z \sim Multi(\theta_u)$
  * Draw word $w \sim Multi(\varphi_z^{(2)})$

• For the u-th user $(u = 1, 2, \ldots, U)$
- For the $< n, m >$ in $< N_u^{(1)}, N_u^{(2)} >$
  * if $\mathbf{Z}[u][n] = \mathbf{Z}[u][m]$ (same topic)
    • Draw vote $f_1, f_2 \sim p(f_1, f_2)$
  * else
    • Draw vote $f_1 \sim Multi(\phi_u^{(1)}), \quad f_2 \sim Multi(\phi_u^{(2)})$

4.2. Learning and parameter estimation

To learn and estimate the parameter of our proposed model, we use collapsed Gibbs sampling to obtain samples of the hidden variable assignment and estimate the model parameters of MultiTEM. The Gibbs Sampling process is described in Algorithm 1.

Algorithm 1 Gibbs Sampling for MultiTEM.

```
procedure GIBBS SAMPLING
    Initialize Z
    for each Gibbs sampling iteration do
        for each user do
            for u's n-th post, n = 1, ..., N_u^(1) do
                Let c denote {u, n}
                Draw z_c according to Eq. (2)
            for u's m-th project, m = 1, ..., N_u^(2) do
                Let c denote {u, m}
                Draw z_c according to Eq. (2)
    Estimate θ, φ^(1), φ^(2) and ψ^(1)
    Initialize E
    for each Gibbs sampling iteration do
        for each user do
            for u's n-th post, n = 1, ..., N_u^(1) do
                for u's m-th project, m = 1, ..., N_u^(2) do
                    Let c denote {u, n, m}
                    if Z[u][n] = Z[u][m] then
                        p(f^(1), f^(2)) ← Eq. (1) (reg.)
                    else
                        p(f^(1), f^(2)) ← Eq. (1) (no reg.)
                    Draw z_c according to Eq. (3)
```

We sample topic z and expertise e sequentially. To start with, we sample topic z_u for each user u and his/her corresponding post n and project m. Let c denote $\{u, n\}$ or $\{u, m\}$, we derive the Gibbs update rule for z_c as follows:

$$
\begin{aligned}
&p(z_c = z|\mathbf{Z}_{\neg c}, \mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{T}^{(1)}, \mathbf{f}^{(1)}, \mathbf{f}^{(2)}, \boldsymbol{\Theta}) \\
&\propto \frac{p(\mathbf{Z}, \mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{T}^{(1)}, \mathbf{f}^{(1)}, \mathbf{f}^{(2)}|\boldsymbol{\Theta})}{p(\mathbf{Z}_{\neg c}^{(1)}, \mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{T}, \mathbf{f}^{(1)}, \mathbf{f}^{(2)}|\boldsymbol{\Theta})} \\
&= \frac{\Delta(C_u^{k(1)} + \alpha^{(1)})}{\Delta(C_{u,\neg c}^{k(1)} + \alpha^{(1)})} \cdot \frac{\Delta(C_z^{w(1)} + \gamma^{(1)})}{\Delta(C_{z,\neg c}^{w(1)} + \gamma^{(1)})} \cdot \frac{\Delta(C_z^{w(1)} + \gamma^{(2)})}{\Delta(C_{z,\neg c}^{w(2)} + \gamma^{(2)})} \\
&\quad \cdot \frac{\Delta(C_z^{t(1)} + \eta^{(1)})}{\Delta(C_{z,\neg c}^{t(1)} + \eta^{(1)})} \\
&= \frac{C_{u,\neg c}^{z(1)} + \alpha^{(1)}}{\sum_{k=1}^{K} C_{u,\neg c}^{k(1)} + K\alpha^{(1)}} \cdot \frac{\prod_{w=1}^{V^{(1)}} \prod_{i=1}^{n_c^w} (C_{z,\neg c}^{w(1)} + \gamma^{(1)} + i - 1)}{\prod_{j=1}^{n_c^w} \prod_{w=1}^{V^{(1)}} (C_{z,\neg c}^{w(1)} + V^{(1)}\gamma^{(1)} + j - 1)} \\
&\quad \cdot \frac{\prod_{w=1}^{V^{(2)}} \prod_{i=1}^{n_c^w} (C_{z,\neg c}^{w(2)} + \gamma^{(2)} + i - 1)}{\prod_{j=1}^{n_c^w} \prod_{w=1}^{V^{(2)}} (C_{z,\neg c}^{w(2)} + V^{(2)}\gamma^{(2)} + j - 1)} \\
&\quad \cdot \frac{\prod_{t=1}^{T} \prod_{p=1}^{n_c^t} (C_{z,\neg c}^{t(1)} + \eta^{(1)} + p - 1)}{\prod_{q=1}^{n_c^t} \prod_{t=1}^{T} (C_{z,\neg c}^{t(1)} + T\eta^{(1)} + q - 1)},
\end{aligned} \tag{2}
$$

where $\Delta(\cdot)$ is a "Dirichlet delta function" which can be seen as a multidimensional extension to beta function.

After the procedure of topic sampling, we can obtain the topic distribution of each user u on his/her each post n and project m. Then for user u we sample the topic of post and project simultaneously. We assume the (μ, Σ) for all the expertise levels are known. Let c denote $\{u, n\}$ or $\{u, m\}$, we derive the Gibbs update rule for $e_{u,n}$ or $e_{u,m}$ as follows:

$$
\begin{aligned}
p(e_c^{(1)} = e^{(1)}, e_c^{(2)} = e^{(2)}|\mathbf{E}_{\neg c}^{(1)}, \mathbf{E}_{\neg c}^{(2)}, \mathbf{f}^{(1)}, \mathbf{f}^{(2)}, \boldsymbol{\Theta}) &\propto \frac{p(\mathbf{E}^{(1)}, \mathbf{E}^{(2)}, \mathbf{f}^{(1)}, \mathbf{f}^{(2)}|\boldsymbol{\Theta})}{p(\mathbf{E}_{\neg c}^{(1)}, \mathbf{E}_{\neg c}^{(2)}, \mathbf{f}^{(1)}, \mathbf{f}^{(2)}|\boldsymbol{\Theta})} \\
&= \frac{\Delta(C_{z,u}^{e(1)} + \beta^{(1)})}{\Delta(C_{z,u,\neg c}^{e(1)} + \beta^{(1)})} \cdot \frac{\Delta(C_{z,u}^{e(2)} + \beta^{(2)})}{\Delta(C_{z,u,\neg c}^{e(2)} + \beta^{(2)})} \\
&\quad \cdot p(f_1, f_2) \\
&= \frac{C_{z,u,\neg c}^{e(1)} + \beta^{(1)}}{\sum_{e=1}^{E^{(1)}} C_{z,u,\neg c}^{e(1)} + E^{(1)}\beta^{(1)}} \\
&\quad \cdot \frac{C_{z,u,\neg c}^{e(2)} + \beta^{(2)}}{\sum_{e=1}^{E^{(2)}} C_{z,u,\neg c}^{e(2)} + E^{(2)}\beta^{(2)}} \cdot p(f_1, f_2),
\end{aligned} \tag{3}
$$

where $p(f^{(1)}, f^{(2)})$ is set to be Eq. (1) with regularization if post n and project m share same topic, otherwise, $p(f^{(1)}, f^{(2)})$ is set to be Eq. (1) without regularization.

To estimate parameters (μ_e, Σ_e) for an expertise level e, we need to consider all the votes/stars associated with e and derive the posterior distribution. We report the derived formula in the following, one can refer to [28] for the detailed derivations.

$$
\begin{aligned}
&p(\mu_e, \sum_e |\mathbf{v}_{\mathbf{i}e_i=e}, \Theta) \\
&\propto p(\mathbf{v}_{\mathbf{i}e_i=e}|\mu_e, \sum_e) \cdot \mathcal{NG}(\mu_0, \kappa_0, \alpha_0, \beta_0) \\
&= \prod_{v:\mathbf{v}_{\mathbf{i}e_i=e}} \mathcal{N}(\mu_e, \sum_e) \cdot \mathcal{NG}(\mu_k, \sum_k |\mu_0, \kappa_0, \alpha_0, \beta_0) \\
&= \mathcal{NG}(\mu_e, \sum_e |\mu_e', \kappa_e', \alpha_e', \beta_e'),
\end{aligned} \tag{4}
$$

where μ'_e, κ'_e, α'_e, β'_e are defined as follows:

$$\mu'_e = \frac{\kappa_0\mu_0 + n_e\bar{\mathbf{v}}_e}{\kappa_0 + n_e}$$
$$\kappa'_e = \kappa_0 + n_e$$
$$\alpha'_e = \alpha_0 + \frac{n_e}{2}$$
$$\beta'_e = \beta_0 + \frac{1}{2}\sum_{v:\mathbf{v}_{i_j}=e}(v-\bar{\mathbf{v}}_e)^2 + \frac{\kappa_0 n_e(\bar{\mathbf{v}}_e-\mu_0)^2}{\kappa_0+n_e}, \quad (5)$$

where $\bar{\mathbf{v}}_e$ is the average vote score/stars for expertise e, n_e is the total number of votes/stars with expertise level e.

Given Eqs. (4) and (5), we can update (μ_e, Σ_e) as follows:

$$\mu_e = \mu'_e, \ \sum_e = \frac{\alpha'_e}{\beta'_e}. \quad (6)$$

With Gibbs Sampling, we can make the following parameter estimation:

$$\theta_{u,k} = \frac{C_u^k + \alpha}{\sum_{k=1}^K C_u^k + K\alpha}, \ \psi_{u,k} = \frac{C_k^t + \eta}{\sum_{t=1}^T C_k^t + T\eta},$$
$$\varphi_{k,w}^{(1)} = \frac{C_k^{w(1)} + \gamma^{(1)}}{\sum_{w=1}^W C_k^{w(1)} + V\gamma^{(1)}}, \ \varphi_{k,w}^{(2)} = \frac{C_k^{w(2)} + \gamma^{(2)}}{\sum_{w=1}^W C_k^{w(2)} + V\gamma^{(2)}},$$
$$\phi_{k,u,e}^{(1)} = \frac{C_{k,u}^{e(1)} + \beta^{(1)}}{\sum_{e=1}^E C_{k,u}^{e(1)} + E\beta^{(1)}}, \ \phi_{k,u,e}^{(2)} = \frac{C_{k,u}^{e(2)} + \beta^{(2)}}{\sum_{e=1}^E C_{k,u}^{e(2)} + E\beta^{(2)}}. \quad (7)$$

4.3. Expert users recommendation

Given a question q and a set of test users U, the target is to rank all these users by their interests and expertise to answer the question q. We score each user u by considering user topic similarity with the question $Sim(u, q)$ and user expertise in the question $Expert(u, q)$, where the intuition is that if the user is interested and have a high expertise for the question, then the user tends to provide a good answer wining high votes. The recommendation score function is defined as follows:

$$S(u,q) = Sim(u,q)\cdot Expert(u,q)$$
$$= (1 - JS(\theta_u,\theta_q))\cdot\sum_{z=1}^{Z}\theta_{q,z}\cdot Expert(u,z), \quad (8)$$

where $JS(\cdot)$ is JS-divergence distance. Note that θ_u and $\phi_{z,u}$ can be obtained from our model results, and $Expert(u, z)$ is the expertise of user u under topic z which can be calculated as follows:

$$Expert(u,z) = \sum_{e=1}^{E^{(1)}}\phi_{z,u,e}^{(1)}\cdot\mu_e^{(1)}, \quad (9)$$

θ_q is the question's topic distribution and it needs to be estimated by computing its posterior probabilities. Specifically, we compute $\theta_{q,z}$ as follows:

$$\theta_{q,z} \propto p(z|\mathbf{w}_q^{(1)}, \mathbf{w}_q^{(2)}, \mathbf{t}_q^{(1)}, u)$$
$$= p(z|u)p(\mathbf{w}_q^{(1)}|z)p(\mathbf{w}_q^{(2)}|z)p(\mathbf{t}_q^{(1)}|z)$$
$$= \theta_{u,z}\sum_{w:\mathbf{w}_q^{(1)}}p(w|z)\sum_{w:\mathbf{w}_q^{(2)}}p(w|z)\sum_{t:\mathbf{t}_q^{(1)}}p(t|z)$$
$$= \theta_{u,z}\sum_{w:\mathbf{w}_q^{(1)}}\varphi^{(1)}(z,w)\sum_{w:\mathbf{w}_q^{(2)}}\varphi^{(2)}(z,w)\sum_{t:\mathbf{t}_q^{(1)}}\psi(z,t), \quad (10)$$

where $\mathbf{w}$ and $\mathbf{t}$ are the set of all the words and tags in question q. Here $\theta_{u,z}$, $\phi(z, w)$ and $\varphi(z, t)$ can be obtained from our model results. After we score each user in U, we rank them in decreasing order of the score. The expert users recommendation algorithm is summarized in Algorithm 2.

Algorithm 2 Expert Users Recommendation Algorithm.

Require: Question q
Ensure: Ranked expert users
procedure Expert User Recommendation
  for $z = 1, \cdots, K$ **do**
    Calculate $\theta_{q,z}$ according to Eq. (10)
  for user $u = 1, \cdots, U$ **do**
    for topic $z = 1, \cdots, K$ **do**
      Calculate $Expert(u, z)$ according to Eq. (9)
    Calculate $S(u, q)$ according to Eq. (8)
  return Ranked expert users according to S

Table 2
An overview of dataset.

User			
GH	15,647,255	Intersection	98,760
SO	1,259,622	Filtered	2959
Training		**Testing**	
Question	3,708	Question	141
Answer	18,961	Answer	454

5. Experiments

5.1. Data collection

According to [29], expertise derived from StackOverflow can somehow also be reflected from GitHub indirectly. In this paper, we consider StackOverflow as a main source for CQA and GitHub as an external source to verify the effectiveness of our proposed model.

StackOverflow. StackOverflow is a popular online programming question and answer community started in 2008. It dumps and releases the dataset every three months. The data used in our experiment is released in August 2012, containing about 1,295,622 registered users dating from July 2008 to August 2012.

GitHub. GitHub, one of the most popular social coding sites, has gained much popularity among a large number of software developers around the world. It has a publicly accessible API. Crawling GitHub website by its API, we get 28,362,019 projects, 15,647,255 users.

Intersection. To intersect data from StackOverflow and GitHub, a conservative approach matching email address is adopted in our experiment. In the GitHub dataset email address are present, while in the StackOverflow dataset email address are obscured, but their MD5 hashes are available. Therefore, we merge a GitHub and a StackOverflow user if their MD5 email hashes are identical.

To train and test our model, we split all the posts by date August 8th 2011. In other words, posts that are posted before that date are used for training and the left for testing. Then we select users who have posted more than 5 posts and contributed to more than one project, so that we can vary their density. In training data, we get 2959 users with 3708 questions, 18,961 answers and 67,601 projects. In testing data, we remove testing questions which have less than 3 answers, then we have 141 questions and 454 answers. Table 2 summarizes the data size as mentioned above. It is worth mention that, the scale of our testing data is a little smaller than that used in [5], this is because that the size of users in our paper is much smaller after the intersection procedure. For data preprocessing, we tokenize text and discard all code snippets. Then we remove the stop words and HTML tags in text.

Fig. 4 shows the activities and expertise distribution of users on cross platforms. From Fig. 4a we note that some users answer few questions in StackOverflow, but contribute lots of projects

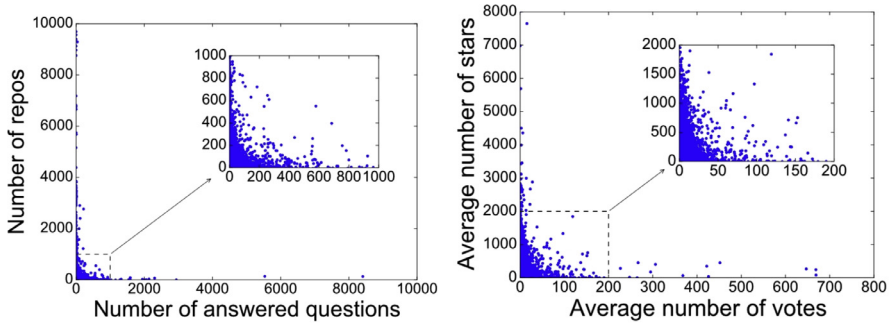


Fig. 4. Activities and expertise distribution of users on cross platforms. (a) Number of questions a user answers in StackOverflow v.s. number of projects this user contributed in GitHub. (b) Average number of votes in StackOverflow v.s. average number of stars in GitHub.

Table 3
Top 20 tags in StackOverflow.

Tag	Frequency	Tag	Frequency
javascript	182,509	ruby	55,463
php	146,926	ios	51,025
java	144,660	css	49,756
c#	143,001	mysql	49,642
python	120,203	.net	48,523
jquery	118,475	objective-c	45,180
c++	79,503	iphone	43,805
android	76,592	c	39,789
ruby-on-rails	74,207	asp.net	36,026
html	70,047	sql	32,371

Table 4
Top words for different topics discovered by MultiTEM.

Topic1	Topic2	Topic3	Topic4	Topic5	Topic6
server	class	presentation	repository	library	page
data	method	bootstrap	sample	javascript	javascript
application	object	project	rails	plugin	html
time	data	framework	data	files	jquery
user	type	app	docker	perl	css
database	table	angular	sideci	module	data
web	database	slides	project	python	make
make	sql	web	openstack	ruby	user
sql	make	angularjs	ruby	data	text
system	methods	application	module	test	control

(repositories) in GitHub. From Fig. 4b we find that although some users' expertise in StackOverflow (votes) is low, they may have a high expertise in GitHub (stars). This exactly verifies the assumption we introduced previously.

Table 3 lists the Top 20 tags in StackOverflow. The frequency column in this table represents the number of question which are tagged with the corresponding tag. From this table, we can find that most tags in StackOverflow are programming languages like *javascript, php* and *java*, which is in accordance with our expectation.

5.2. Experimental setup

For all experiments, we empirically set hyperparameters according to suggestions in [5]. For set Dirichlet parameters, we set $\alpha = 50/K$, $\beta^{(1)} = \beta^{(2)} = 0.01$, $\gamma^{(1)} = \gamma^{(2)} = 0.01$, $\eta = 0.001$. For Norma-Gamma parameters, we set μ_0 as the mean of votes/stars from our datasets, κ_0 as 1, α_0 as 1, and β_0 as the mean distance between randomly sampled 1000 votes/stars. We run MultiTEM with 1000 iterations of Gibbs sampling. With some trails on the number of topics and expertise, we set topic number $K = 15$, expertise number $E^{(1)} = E^{(2)} = 10$ as they provide meaningful topics and vote Gaussian distributions for our datasets.

Evaluation metrics. By following the study [5], we use NDCG (normalized Discounted Cumulative Gain) as the evaluation metrics. Let $vote(i)$ denote the vote counts of the answer ranked at i in a system output. The NDCG is formally defined as follows:

$$DCG@N = \sum_{i=1}^{N} \frac{\log(vote(i)+1)}{\log(i+1)} \tag{11}$$

$$NDCG@N = \frac{DCG@N}{maxDCG@N} \tag{12}$$

where *maxNDC* is the DCG of the ideal ranking where answer list is sorted by vote counts in a descending order.

Baselines. To evaluate the effectiveness of MultiTEM, we compare against some baselines in previous related works:

- *TEM.* TEM [5] is our main reference model. This model integrates users' expertise and interests together in the context of single view. We apply this model into StackOverflow and GitHub data separately and it derives two related models, e.g., TEM+SO and TEM+GH.
- *UQA.* [15] proposed a User-Question-Answer Model for modeling of Q&A text. One major difference between UQA and TEM model is that UQA does not model user topical expertise. We apply this model into StackOverflow and GitHub data separately and it also derives two related models, e.g., UQA+SO and UQA+GH. Besides, it can also be extended handle multiple sources (MultiUQA).
- *SingleUQA/SingleTEM.* To illustrate the advantages of multiple views over single view, we only consider the projects information from GitHub into account and concatenate the multiple views as a single view. Based on UQA and TEM model, we derive the SingleUQA and SingleTEM models from the perspective of single view.
- *MultiTEM.* This model is the model proposed in this paper.

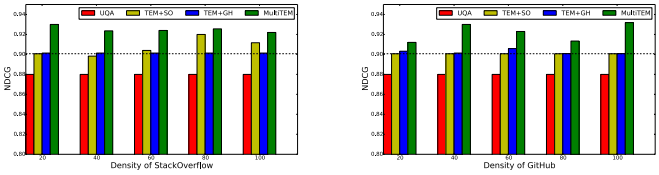
All the experiments in this paper are implemented with Python 2.7, and run on a computer with an 2.2 GHz Intel Core i7 CPU and 64 GB 1600 MHz DDR3 RAM, running Debian 7.0.

5.3. Experimental results

To have an intuitive feel for topical interest of users, we show six "topics" (i.e., highly probable words) that are discovered from the intersection dataset of GitHub and StackOverflow using our MultiTEM model in Table 4. From this table we can see that these discovered "topics" are related to programming which are within our expectation.

Table 5
Best performance of different models.

Model	NDCG
UQA+SO (80%)	0.9204 ± 0.0130
UQA+GH (40%)	0.9124 ± 0.0201
TEM+SO (60%)	0.9231 ± 0.0114
TEM+GH (60%)	0.9103 ± 0.0136
SingleUQA (20%SO+80%GH)	0.9221 ± 0.0289
SingleTEM (20%SO+100%GH)	0.9238 ± 0.0346
MultiUQA (20%SO+80%GH)	0.9282 ± 0.0063
MultiTEM (20%SO+80%GH)	**0.9312 ± 0.0095**



(a) Impact of density of StackOverflow. (b) Impact of density of GitHub.

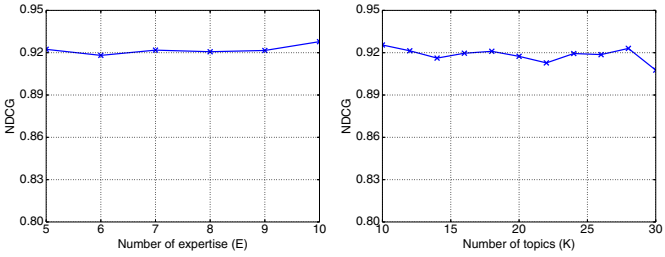
Fig. 5. Impact of density of StackOverflow and GitHub dataset.

We run each comparison model for five times and calculate the NDCG values and their corresponding variances. Table 5 shows the best performance of different models for the task of expert user recommendation. Some conclusions can be drawn from this table.

- When considering only one dataset (e.g., StackOverflow or GitHub), the performances of UQA and TEM mode are very similar, UQA which just considers topical interest performs even better than the TEM which also considers the topical expertise. This can be illustrated by that on the condition of textual information is limited, considering the expertise factor will only increase the complexity of the model, resulting in the performance degradation.
- From the results of UQA model, we note that when we just introduce the textual information from external source (e.g., GitHub), ignoring the expertise information, the performance does have a certain improvement, indicating the effectiveness of introducing more textual information.
- Comparing the results of single view version of UQA/TEM with that of multiple views, we can find that introducing multiple views of users really boost the performance, which also verifies the effectiveness of our proposed Bayesian co-training based multi-view learning model.
- Comparing the results of MultiTEM and MultiUQA, we can note that our MultiTEM model which considers the textual information and expertise factor from external source simultaneously achieves a better performance than the MultiUQA which just takes the textual information into consideration, indicating the effectiveness of our probabilistic graph model proposed in this paper.

Overall, in the expert users recommendation task, our MultiTEM model outperforms other baseline methods in same condition.

In order to measure the impact of data sparsity, as well as the degree of external information we added, we vary the density of StackOverflow and GitHub and conduct experiments on these data respectively. We first fix the density of GitHub to be 40% and vary the density of StackOverflow from 20% to 100% with step of 20%. Fig. 5a shows the impact of density of StackOverflow. From this figure we can find with the varying of density of StackOverflow, our MultiTEM achieves a better performance than baseline models, indicating the effectiveness of information extracted from external sources. However, it's worth noting that with the increase in density of StackOverflow, the performance advantage of MultiTEM roughly becomes smaller. This indicates that when the density of StackOverflow is large enough, some traditional models have performed well enough.



(a) Impact of E (b) Impact of K

Fig. 6. Impact of expertise number E and topic number K.

We then fix the density of StackOveflow to be 20% and vary density of GitHub from 20% to 100% with step of 20%. Fig. 5b shows the impact of density of GitHub. Similar to Fig. 5a, from Fig. 5b we can find that the performance of our MultiTEM still keeps better then that of other baselines when varying of density of GitHub. It is worth noting that with the increase in density of GitHub, the performance of MultiTEM does not always increase, indicating that it is not the more external information the better. This can be illustrated by the possibility that while incorporating the information from external sources, the noise are also introduced at the same time. It is necessary to balance the density between StackOveflow and GitHub.

5.4. Parameter analysis

In our model, E ($E^{(1)} = E^{(2)} = E$) and K represent the number of expertise level and topics, respectively. To study the impact of those parameters on the NDCG metric of our approach, we set the density of StackOverflow to be 20%, the density of GitHub to be 40%. And we vary E from 5 to 15 with step of 1 and K from 10 to 30 with step of 2. Fig. 6 shows the impact of those parameters.

From Fig. 6(a), we observe that the impact of parameter E to *NDCG* is a little small since the *NDCG* is stable when varying its values. From Fig. 6(b), we observe that the parameter K really has an impact on *NDCG*. The optimal value of K may be $14-18$ or $24-28$.

6. Conclusion and future work

Learning topical expertise of users is an essential task in CQA. Considering one limitation of existing approaches that based on historical posts of users, ignoring the cold-start users who are inactive in CQA while leave footprints in other site. In this paper, we aims to warm these cold-start users via exploiting cross-source knowledge. We propose a unified probabilistic graph model which jointly models topics and expertise from cross sources with a hypothesis consistency. We apply our model to a specific task of expert user recommendation for a given question. Comprehensive experimental studies on StackOverflow and GitHub datasets demonstrate the effectiveness of our model when compared to other existing methods.

As our paper can only handle the code-start users who have left footprint in at least one data source and can not deal with the users who are beyond the intersection of data sources, in our future work, we will focus on dealing with those cold-start users whose information are difficult to collect from any data source. And we also plan to apply our model to some other tasks such as answers recommendation and similar questions to better validate our model. We expect to further study the temporal aspect

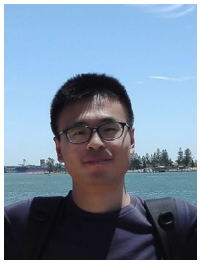
of users in CQA. In real world, the interests and expertise of users change with time. Capturing such temporal information could be more beneficial in recommendation tasks of CQA. Another interesting aspect is the social influence of users on CQA. The answerer profile might influence the voting behavior of users and hence impacts the recommendation methods. It is an interesting problem to analyze the correlated components in CQA for adaptive recommendation systems. In addition, we will try some transfer learning methods so that we do not need to intersect the StackOverflow and GitHub data, and we will get more samples for training and testing.

Acknowledgments

This work was done during Yao Wan's visit to University of Technology Sydney. This work is partially supported by the Ministry of Education of China under grant of No. 2017PT18, the Natural Science Foundation of China under grant of Nos. 61672453, 61773361, 61473273, 61602405, the WE-DOCTOR company under grant of No. 124000–11110 and the Zhejiang University Education Foundation under grant of No. K18-511120-004, No. K17-511120-017. This work is also supported by Australian Research Council Linkage Project (LP140100937). The authors would like to thank the reviewers for their valuable comments on this work.

References

[1] Yahoo! Answers, 2017, (https://answers.yahoo.com). [Online; Accessed 24.01.17].
[2] Quora, 2017, (https://www.quora.com/). [Online; Accessed 24.01.17].
[3] StackOverflow, 2017, (http://stackoverflow.com). [Online; Accessed 24.01.17].
[4] M. Bouguessa, B. Dumoulin, S. Wang, Identifying authoritative actors in question-answering forums: the case of yahoo! answers, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2008, pp. 866–874.
[5] L. Yang, M. Qiu, S. Gottipati, F. Zhu, J. Jiang, H. Sun, Z. Chen, CQARANK: jointly model topics and expertise in community question answering, in: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, ACM, 2013, pp. 99–108.
[6] J. Zhang, M.S. Ackerman, L. Adamic, Expertise networks in online communities: structure and algorithms, in: Proceedings of the 16th International Conference on World Wide Web, ACM, 2007, pp. 221–230.
[7] G. Zhou, S. Lai, K. Liu, J. Zhao, Topic-sensitive probabilistic model for expert finding in question answer communities, in: Proceedings of the 21st ACM International conference on Information and Knowledge Management, ACM, 2012, pp. 1662–1666.
[8] Y. Xiao, W.X. Zhao, K. Wang, Z. Xiao, Knowledge sharing via social login: exploiting microblogging service for warming up social question answering websites., in: Proceedings of the COLING, Citeseer, 2014, pp. 656–666.
[9] S. Yu, B. Krishnapuram, R. Rosales, R.B. Rao, Bayesian co-training, J. Mach. Learn. Res. 12 (Sep) (2011) 2649–2680.
[10] H. Zhu, E. Chen, H. Xiong, H. Cao, J. Tian, Ranking user authority with relevant knowledge categories for expert finding, World Wide Web 17 (5) (2014) 1081–1107.
[11] P. Jurczyk, E. Agichtein, Discovering authorities in question answer communities by using link analysis, in: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge management, ACM, 2007, pp. 919–922.
[12] J.M. Kleinberg, Authoritative sources in a hyperlinked environment, J. ACM (JACM) 46 (5) (1999) 604–632.
[13] F. Xu, Z. Ji, B. Wang, Dual role model for question recommendation in community question answering, in: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2012, pp. 771–780.
[14] X. Liu, W.B. Croft, M. Koll, Finding experts in community-based question-answering services, in: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, ACM, 2005, pp. 315–316.
[15] J. Guo, S. Xu, S. Bao, Y. Yu, Tapping on the potential of Q&A community by recommending answer providers, in: Proceedings of the 17th ACM Conference on Information and Knowledge Management, ACM, 2008, pp. 921–930.
[16] S. Ghosh, N. Sharma, F. Benevenuto, N. Ganguly, K. Gummadi, Cognos: crowdsourcing search for topic experts in microblogs, in: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2012, pp. 575–590.
[17] F. Riahi, Z. Zolaktaf, M. Shafiei, E. Milios, Finding expert users in community question answering, in: Proceedings of the 21st International Conference on World Wide Web, ACM, 2012, pp. 791–798.
[18] J. Lin, K. Sugiyama, M.-Y. Kan, T.-S. Chua, Addressing cold-start in app recommendation: latent user models constructed from twitter followers, in: Proceedings of the 36th international ACM SIGIR conference on Research and Development in Information Retrieval, ACM, 2013, pp. 283–292.
[19] S.-T. Park, W. Chu, Pairwise preference regression for cold-start recommendation, in: Proceedings of the Third ACM Conference on Recommender Systems, ACM, 2009, pp. 21–28.
[20] H. Yin, B. Cui, J. Li, J. Yao, C. Chen, Challenging the long tail recommendation, Proc. VLDB Endow. 5 (9) (2012) 896–907.
[21] S. Purushotham, Y. Liu, C.-C. J. Kuo, Collaborative topic regression with social matrix factorization for recommendation systems, arXiv: 1206.4684(2012).
[22] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, ACM, 1998, pp. 92–100.
[23] I. Muslea, S. Minton, C.A. Knoblock, Active+ semi-supervised learning= robust multi-view learning, in: Proceedings of the ICML, 2, 2002, pp. 435–442.
[24] V. Sindhwani, P. Niyogi, M. Belkin, A co-regularization approach to semi-supervised learning with multiple views, in: Proceedings of ICML Workshop on Learning with Multiple Views, Citeseer, 2005, pp. 74–79.
[25] J. Farquhar, D. Hardoon, H. Meng, J.S. Shawe-taylor, S. Szedmak, Two view learning: SVM-2k, theory and practice, in: Proceedings of the Advances in Neural Information Processing Systems, 2005, pp. 355–362.
[26] V. Sindhwani, D.S. Rosenberg, An rkhs for multi-view learning and manifold co-regularization, in: Proceedings of the 25th International Conference on Machine Learning, ACM, 2008, pp. 976–983.
[27] N. Chen, J. Zhu, E.P. Xing, Predictive subspace learning for multi-view data: a large margin approach, in: Proceedings of the Advances in Neural Information Processing Systems, 2010, pp. 361–369.
[28] K.P. Murphy, Conjugate Bayesian analysis of the Gaussian distribution, def 1 ($2\sigma 2$) (2007) 16.
[29] B. Vasilescu, V. Filkov, A. Serebrenik, Stackoverflow and github: associations between software development and crowdsourced knowledge, in: Proceedings of the 2013 International Conference on Social Computing (SocialCom), IEEE, 2013, pp. 188–195.



Yao Wan received the B.S degree in the Colleage of Software Engineering, Northeastern University, China, in 2014. He is currently working toward the Ph.D degree in the College of Computer Science, Zhejiang University. His research interests include machine learning and data mining.

Guandong Xu received the Ph.D. degree in Computer Science from Victoria University, Australia. He is currently an Associate Professor of School of Software and the Advanced Analytics Institute at University of Technology Sydney. He has authored three monographs with the Springer and the CRC Press, and 100+ journal and conference papers. His current research interests include data science and data analytics, Web data mining, behaviour analytics, recommender systems, predictive analytics, social network analysis. Dr. Xu has served in the Editorial Board or as Guest Editor for several international journals. He is the Assistant Editor-in-Chief of the World Wide Web Journal.

Liang Chen received the Ph.D. and B.S. degree in computer science from Zhejiang University, Hangzhou, China in 2009 and 2015, respectively. He is currently a Postdoc Research Fellow in Advanced Analytical Institute, University of Sydney Technology, Australia. Dr. Chen has served as the workshop co-chair in many conferences, such as CIKM, BigData, PAKDD, etc. His publications have appeared in many well-known conference proceedings and international journals, e.g, WWW, ICSE, ICDM, CIKM, ICSOC, ICWS, TSC, TSMC, WWWJ, etc. His research interests include service computing, recommender system, and data mining.

Zhou Zhao received the B.S. and Ph.D. degrees in computer science from The Hong Kong University of Science and Technology, in 2010 and 2015, respectively. He is currently an Associate Professor with the College of Computer Science, Zhejiang University. His research interests include machine learning and data mining.

Jian Wu received his B.S. and Ph.D. Degrees in computer science from Zhejiang University, Hangzhou, China, in 1998 and 2004, respectively. He is currently a full professor at the College of Computer Science, Zhejiang University, and visiting professor at University of Illinois at Urbana-Champaign. His research interests include service computing and data mining. He is the recipient of the second grade prize of the National Science Progress Award. He is currently leading some research projects supported by National Natural Science Foundation of China and National High-tech R&D Program of China (863 Program).