

Expert Recommendation via Tensor Factorization with Regularizing Hierarchical Topical Relationships

Chaoran Huang¹, Lina Yao¹, Xianzhi Wang², Boualem Benatallah¹, Shuai Zhang¹, and Manqing Dong¹

¹ UNSW Sydney, NSW 2052, Australia
{chaoran.huang,lina.yao}@unsw.edu.au

² University of Technology Sydney, Broadway, NSW 2007, Australia
sandyawang@gmail.com

Abstract. Knowledge acquisition and exchange are generally crucial yet costly for both businesses and individuals, especially when the knowledge concerns various areas. Question Answering Communities offer an opportunity for sharing knowledge at a low cost, where communities users, many of whom are domain experts, can potentially provide high-quality solutions to a given problem. In this paper, we propose a framework for finding experts across multiple collaborative networks. We employ the recent techniques of tree-guided learning (via tensor decomposition), and matrix factorization to explore user expertise from past voted posts. Tensor decomposition enables to leverage the latent expertise of users, and the posts and related tags help identify the related areas. The final result is an expertise score for every user on every knowledge area. We experiment on Stack Exchange Networks, a set of question answering websites on different topics with a huge group of users and posts. Experiments show our proposed approach produces steady and premium outputs.

Keywords: Knowledge discovery; Stack Exchange Networks; Expertise finding; Question answering

1 Introduction

Question and Answering (Q&A) websites are gaining momentum as an effective platform for knowledge sharing. These websites usually have numerous users who continuously contribute. Many researchers have shown interests in the recommendation issues on these websites such as identifying experts. Despite the tremendous research efforts on user recommendation, no state-of-the-art algorithms consistently stand out compared with the others. As the recent work increasingly focuses on domain-specific expertise recommendation, there emerges the research on multi-domain (or cross-domain) recommendation in the “Stack Exchange (SE) Networks”³ repository. SE is a network of 98 Q&A subsites, all

³ stackexchange.com

following the same structure. This consistency enables us to expand our approach from one subsites to the all the other subsites on SE. These subsites cover various disciplines from computer science to even the Ukrainian language. Take “Stack Overflow”⁴ (SO) as an example(Figure 4). It is a software-domain-oriented website where users can post and answer questions, or vote up/down to other users’ questions and answers. The author of a question (a.k.a., the requester) can mark an answer as accepted and offer a bounty to the answerer.

So far, there are two popular ways to locate experts: collaborative filtering(CF) and content-based recommendation. The former extracts similar people without understanding the contents while the latter focuses on building user profiles based on users’ activity history. CF relies merely on ratings (e.g., scores in SE networks) and therefore may not well handle sparse Q&A subsites data, where many questions involve very limited users. Usually, users can vote on questions, and the vote counts can serve as ratings to the questions. An earlier work [1] also suggests that the lack of information can be a challenge for recommendation techniques. The work aims to address the data sparsity issue by selectively using the ratings of some experts. This experts presumed by this approach is exactly the same experts we aim to find. As for content-based approaches, a typical approach (e.g., [18]) builds user profiles based on user’s knowledge scores and user authority in link analysis. The knowledge scores are called reputation in [18], which is derived from users’ historical question-answering records. Srba et al. [23] point out that some users may maliciously post low-quality content, and those highly active spammers might be taken as experts in a system. Huna et al. [11] solve this problem by calculating question and answer difficulties based on three aspects of hints: the numbers of user-owned questions and answers, time difference of the question being posted and answered, average answering time, and score of the answer with the maximum of score among all the answers provided by the answerer. Although these approach may compute user reputation, they also take considerable cost on building user profiles. Matrix Factorization is one method that works on sparse data , while matrices can only store two dimensions of data, which is not handy in many applications, where users’ attributes can be vital to the identification of experts. Recently tensor-based approaches became popular as an alternative to matrix factorization, made it feasible to handle multi-faceted data[28]. For example, Ge et al. in [7] decompose a (Users, Topics, Experts) tensor for the personalized expert recommendation; Bhargave et al. [3] propose a (User, Location, Activity, Time) tensor decomposition along with correlated matrix to make recommendations based on user preferences.

We aim to recommend experts in multiple areas simultaneously. In particular, we use the Stack Exchange networks dump, which contains various areas, to build up a multi-domain dataset. We propose group lasso [15] that works on a relationship tree formed upon the natural structure of the SE network. The tree is used to guide the decomposition of 4th rank tensor data consisting of questions, topics, voting and expertise information. We additionally factorize selected matrices to provide additional latent information.

⁴ stackoverflow.com

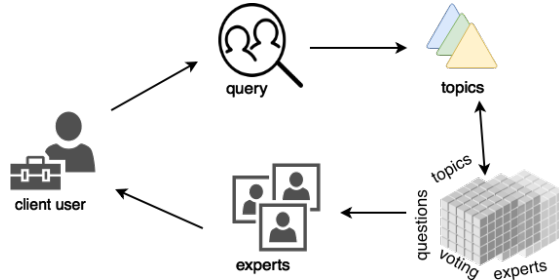


Fig. 1. Work-flow of our proposed methodology: For a given input query, experts are output based on the detected topic of the query combined with our 4th order tensor, which contains latent information like topics, questions, voting, and experts.

Our contributions in this work are as follows:

1. We take the hierarchical relationship between participants and topics into account and build a model that combines tree-guided tensor decomposition and matrix factorization;
2. We introduce the relationship tree group lasso to alleviate the data sparsity problem;
3. We conduct experiments on real-world data and evaluate the proposed approach against state-of-the-art baselines.

2 Related Works

Expert recommendation has been studied extensively in the past decade. Generally, skillfulness and resourcefulness of experts can assist users in making decisions more professionally and solving problems more effectively and efficiently. That is, making appropriate recommendations to users with the different requirement can be important.

The expert recommendation techniques apply to many areas, and different fields may require differently in methodologies to handle different situations. Baloga et al. [2] introduce a generative probabilistic framework for find experts in various enterprise data sources. Daud et al. [4] devise a Temporal-Expert-Topic model to capture both the semantic and dynamic expert information and to identify experts for different time periods. Fazelzarandi et al. [6] develop an expert recommendation system with utilizing the social networks analysis and multiple data source integration techniques. Wang et al. [24] propose a model ExpertRank which take both document profile and authority of experts into consideration to perform better. Huang et al. [10] take advantage of word embedding technology to rank experts both semantically and numerically. More relate works can be found in a survey by Wang et al. [25].

The works mentioned above mostly focus on recommend experts for organizations, enterprises or institutes. There is also some literature on recommending

experts in Q&A System, which is more related to our work. Kao et al. [13] propose to incorporate user subject relevance, user reputation and authority of categories into expert finding system in Q&A websites. Riahi et al. [22] investigate two topic model namely Segmented Topic Model and Latent Dirichlet Allocation model to direct new questions in Stack-overflow to related experts. Ge et al. [7] propose a personalized tensor-based method for expert recommendation by considering factors like geospatial, topical and preferences. Liu et al. in [19] propose a method to rank user authority by exploiting interactions between users, which is aimed to avoid potential impacts of users with considerable social influences. They introduced topical similarities into link analysis to rank user authorities for each question. Latent Dirichlet allocation is applied to extract topics from both the questions and answers of users so that topical similarities between questions and answers can be measured, and then related users can be ranked by links. Huna et al. found Q&A communities often evaluate user reputation limited to the number of user activities[11], regardless of efforts on creating high-quality contents. This causes inaccurate measurements in user expertise and their value. Inspired by former works, they calculate user reputations for asking and answering questions. The reputation results from the combination of the difficulty score of a question and the utility score for the question or answer. A utility score measures the distance between a score and the maximum score of the post, and the difficulty measures the times that a user spends on the question. The time spent on questions is normalized on each topic. Fang et al. [5] are well aware of the quantity of social information Q&A website can provide, along with the importance of user-generated textual contents. Their idea to simultaneously model both social links and textual contents leads to the proposed framework named “HSNL”(CQA via **H**eterogeneous **S**ocial **N**etwork **L**earning). The framework adopts random walk to exploit social information and build the heterogeneous social network, and a deep recurrent neural network was trained to give a text-based matching score for questions and answers.

Our proposed model builds on tensor decomposition, which has been applied to various fields such as neuroscience, computer vision, and data mining [17]. CANDECOMP/PARAFAC (CP) and Tucker decomposition are two effective ways to solve tensor decomposition problems. We adopt the former in this work. Tensor decomposition based recommender systems can also be found widespread in recent studies. Rendle et al. [20] introduce a tensor factorization based ranking approach for tag recommendation. They further improve the model by introducing pairwise interaction and significantly improve the optimization efficiency. Xiong et al. [26] propose a probabilistic tensor decomposition model and regard the temporal dynamics as the third-dimension of the tensor. Karatzoglou et al. [14] offer a context-aware tensor decomposition model to integrate context information with collaborative filtering tightly. Hidas et al. [9] investigate approach which combines implicit feedback with context-aware decomposition. Bhargava et al. [3] present a tensor decomposition-based approach to model the influence of multi-dimensional data sources. Yao et al. [27] decompose tensor with contextual regularization to recommend location points of interest.

3 Methodology

CANDECOMP/PARAFAC Tensor Decomposition, or CP Decomposition, is discovered by Kiers and Möcks independently[17]. For a Rank- R size- N tensor \mathcal{X} ($R \in \mathbb{N}$), let $U_1 \in \mathbb{R}^{I_1 \times R}$, $U_2 \in \mathbb{R}^{I_2 \times R}$, ..., $U_R \in \mathbb{R}^{I_N \times R}$, we have the decomposition:

$$\mathcal{X} \approx \sum_{r=1}^R U_{1i_1r} U_{2i_2r} \cdots U_{Ri_Nr} \quad (1)$$

While multiple methods can do tensor decomposition, the most common and effective one shall be the alternating least squares(ALS)[17].

3.1 Relationship Tree Modelling

Our data is naturally divided into subsites, topics, and posts, as shown in Figure 3. This decomposition forms a tree, with subsites on top, and posts as leaves. As our tensor models the expertise information based on user activities, this tree reserves the relationships of entities. We illustrate the construction of the tree as follows.

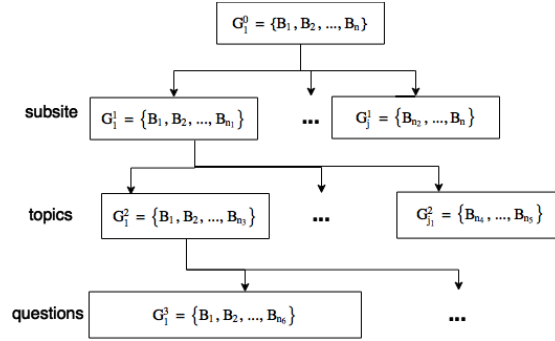


Fig. 2. An example of modeled tree representation of hierarchical relationship

Given the tree \mathcal{T} , we assume that the i -th level of \mathcal{T} has n_i nodes and organized as $\mathcal{T}_i = \{G_1^i, G_2^i, \dots, G_{n_i}^i\}$. And so, a group G_v where node $v \in V$ is in the tree, and all all leaves under v are in G_v . Now we can define a tree-structured regulation as

$$Weight(\mathbf{U}_1) = \frac{\lambda_W}{2} \sum_{k=1}^J \omega_j^i \|\mathbf{U}_{1_k}\|_2^2 \quad (\mathbf{U}_{1_k} \in G_j^i) \quad (2)$$

This inspired from Moreau-Yosida regularization, and here λ_W is the Moreau-Yoshida regulation parameter for tree \mathcal{T} , $\|\cdot\|$ denotes Euclidean norm, \mathbf{U}_{1_k} is

a vector of \mathbf{U}_1 , where \mathbf{U}_1 is the first factor matrix of the tensor \mathcal{X} , which corresponding to a question post and detailed explanation can be found in the following subsection. Additionally, ω_j^i is set by following Kim’s approach[16] and it means a pre-set weight for j -th node at level i . ω_j^i can be obtained by setting two variables summed up to 1, i.e. s_j^i for the weight of independent relevant covariates selecting and g_j^i for group relevant covariates selecting. We have:

$$\sum_i^d \sum_j^n \omega_j^i \|\mathbf{U}_{1_{G_j^i}}\|_2 = \lambda \omega_0^j \quad (3)$$

where

$$\omega_j^i = \begin{cases} s_j^i \cdot \sum_{c_p^g \in \text{Child}(v_j^i)} |\omega_p^g| + g_j^i \cdot \|\mathbf{U}_{1_{G_j^i}}\|_2 & v_j^i \text{ is a internal node,} \\ |\mathbf{U}_{1_{G_j^i}}| & v_j^i \text{ is a leaf node.} \end{cases} \quad (4)$$

3.2 Proposed Model

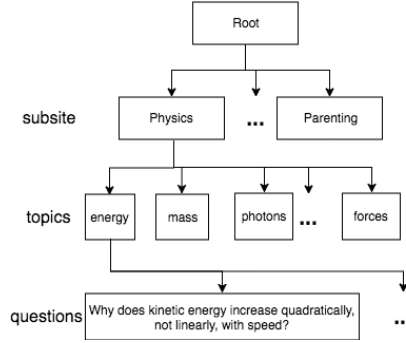


Fig. 3. Tree representation of hierarchical entity relationship



Fig. 4. An example of Stack Overflow post(postId:34672987), here demonstrates a question with its description and comments, along with score of the question.

Our dataset is obtained naturally categorized by their subdomains, which we call it “subsites” here. Additionally, in each subsite, we can find tags in every post, and such information is often an indicator of the post’s topics. Accordingly, after gathering those data, we can build a tree to represent such hierarchical information(shown in Figure 3).

All Stack Exchange subsites share the same structure. That means, in all this subsites, answerers may propose multiple answers and questioners can adopt only one answer for each question. Also, both question and answers can be commented and voted, and the difference between vote-ups or vote-downs on each question is calculated into a score. Figure 4 show an example.

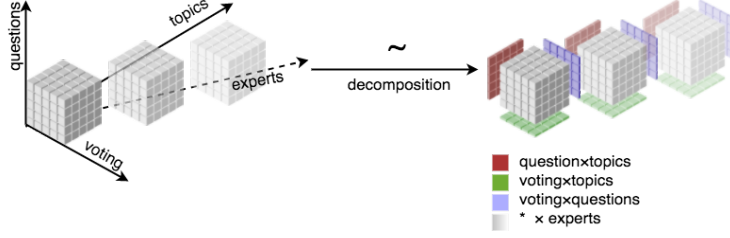


Fig. 5. Proposed decomposition

Instead of the simple score-user matrix based recommendation, we propose a tensor-decomposition based tree-guided method, based on the basic idea of Tree-Guided Sparse Learning[12].

1. A 4th-order-tensor, **Question** \times **Topic** \times **Voting** \times **Expert**. Shown in Figure 5, we denoted it as $\mathcal{X} \in \mathbb{R}^{I \times J \times K \times L}$, where I is the number of questions, J is the the number of Topics, K is the number of voting of question towards questioners, L is the expert users and the value of the tensor is the number of expertise evaluation criterion. With limited users participated in certain domains, it is believed that the tensor is very sparse. Additionally we denote $\mathbf{U}_1 \in \mathbb{R}^{I \times R}$, $\mathbf{U}_2 \in \mathbb{R}^{J \times R}$, $\mathbf{U}_3 \in \mathbb{R}^{X \times R}$, $\mathbf{U}_4 \in \mathbb{R}^{L \times R}$ as factor matrices of tensor \mathcal{X} .
2. A **subsite** \times **answerer** matrix. We denoted this as $M \in \mathbb{R}^{X \times Z}$, where if answerer z appears in subsite x , $M_{x,z} = 1$ else $M_{x,z} = 0$.
3. A **topics** \times **answerer** matrix. We denoted this as $N \in \mathbb{R}^{Y \times Z}$, similarly here, when answerer z appears in topic y , $M_{y,z} = 1$ else $M_{y,z} = 0$.
4. Hierarchical relationship tree \mathcal{T} of depth d . Due to the isolation of subsites and their topics, our data show clearly a structured sparsity. Thus, we can utilize tree-guided group lasso in our model. That is, besides above two supplement matrices, we also use the tree shown in Figure3 to guide the learning.

After modeling the data, we apply CANDECOMP/PARAFAC (CP) tensor decomposition to factorize the tensor and solve the tree-structured regression with group lasso.

First, we decompose the 4th-order tensor with regulation by Alternating Least Square (ALS) as follows:

$$\begin{aligned} \text{Tensor}(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \mathbf{U}_4) = & \frac{1}{2} \|\mathcal{X} - [\![\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \mathbf{U}_4]\!]\|_F^2 \\ & + \frac{\lambda_{\mathcal{X}}}{2} (\|\mathbf{U}_1\|_F^2 + \|\mathbf{U}_2\|_F^2 + \|\mathbf{U}_3\|_F^2 + \|\mathbf{U}_4\|_F^2) \end{aligned} \quad (5)$$

Then, we can have the aforementioned 2 matrices decompose as :

$$\text{Networks}(\mathbf{S}, \mathbf{A}) = \frac{1}{2} \|\mathbf{M}_{site} - \mathbf{S}\mathbf{A}^T\|_F^2 + \frac{\lambda_S}{2} (\|\mathbf{S}\|_F^2 + \|\mathbf{A}\|_F^2) \quad (6)$$

Table 1. Symbol table

Symbol	Description
$\mathcal{X} \in \mathbb{R}^{I \times J \times K \times L}$	a 4th-order-tensor, I, J, K, L accordingly is the number of Question, Topic, Voting and Expert
$\mathbf{U}_1 \in \mathbb{R}^{I \times R}$ $\mathbf{U}_2 \in \mathbb{R}^{J \times R}$ $\mathbf{U}_3 \in \mathbb{R}^{K \times R}$ $\mathbf{U}_4 \in \mathbb{R}^{L \times R}$	factor matrices of tensor \mathcal{X}
$M \in \mathbb{R}^{X \times Z}$	subsite \times answerer matrix where X, Z are the number of subsite and answerer
$N \in \mathbb{R}^{Y \times Z}$	topic \times answerer matrix where Y are the number of topic
$\mathcal{T}_i = \{G_1^i, G_2^i, \dots, G_{n_i}^i\}$	set of node in the i -th level of tree \mathcal{T} G_{n_i} is the n_i -th node in the level

$$Topic(\mathbf{T}, \mathbf{A}) = \frac{1}{2} \|\mathbf{M}_{topic} - \mathbf{T}\mathbf{A}^T\|_F^2 + \frac{\lambda_T}{2} (\|\mathbf{T}\|_F^2 + \|\mathbf{A}\|_F^2) \quad (7)$$

Since each subsite S_j contains a group of questions \mathbf{U}_{1_j} , we expect S_j to be similar to the average \mathbf{U}_{1_j} , which can be solved as a regulation:

$$Site(\mathbf{S}, \mathbf{U}_1) = \frac{\lambda_S}{2} \sum_{j=1}^U \|\mathbf{S}_j - \frac{1}{G_j^1} \sum_{\mathbf{U}_{1_k} \in G_j^1} \mathbf{U}_{1_k}\|_2^2 \quad (8)$$

By combining those objectives and regulations, we have the following objective function:

$$\begin{aligned} f(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \mathbf{U}_4, \mathbf{S}, \mathbf{A}, \mathbf{T}) = & Tensor(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \mathbf{U}_4) \\ & + Weight(\mathbf{U}_1) + Networks(\mathbf{S}, \mathbf{A}) \\ & + Topic(\mathbf{T}, \mathbf{A}) + Site(\mathbf{S}, \mathbf{U}_2) \end{aligned} \quad (9)$$

Equation 5 follows the CANDECOMP/PARAFAC Decomposition, accomplished by the ALS algorithm (see Algorithm 1), which is a popular way to decompose a tensor.

Computational Complexity Analysis. The time complexity of the above decomposition includes two parts. The first concerns initializing the set of $\mathbf{A}^{(n)}$ s. We note the average of the dimension of our tensor as D , which we use to represent the size of the tensor as \mathbf{D}^N . The initialization is a traverse of $\mathbf{A}^{(n)}$ s

Algorithm 1 CP Decomposition via Alternating Least Squares, where N -th order tensor \mathcal{X} of size $I_1 \times I_2 \times \dots \times I_N$ is decompose into R components

Input: \mathcal{X}, R

Output: $\lambda, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(n)}$

Algorithm $CP - ALS(\mathcal{X}, R)$

initialize $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ for $n = 1, 2, \dots, N$

```

1: for  $n = 1, 2, \dots, N$  do
2:    $\mathbf{V} \leftarrow \mathbf{A}^{(1)\top} \mathbf{A}^{(1)} * \dots * \mathbf{A}^{(n-1)\top} \mathbf{A}^{(n-1)} * \mathbf{A}^{(n+1)\top} \mathbf{A}^{(n+1)} * \dots * \mathbf{A}^{(N)\top} \mathbf{A}^{(N)}$ 
3:    $\mathbf{A}^{(n)} \leftarrow \mathbf{X}^n(\mathbf{A}^{(n)} \odot \dots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \dots \odot \mathbf{A}^{(1)}) \mathbf{V}^*$ 
4:   normalize columns of  $\mathbf{A}^{(n)}$  and store norms as  $\lambda$ 
5:   if fit stops improve or iteration reach threshold then
6:     break
7:   end if
8: end for
9: return  $\lambda, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(n)}$ 

```

and has a time complexity of $\mathcal{O}(NDR)$. Assuming that we use index flip to implement the matrix transpose, its time complexity is $\mathcal{O}(1)$. Thus, the total time complexity on N loops is $\mathcal{O}((NDR)^2 + N^2DR)$ time. Combining the two steps, we now have the time complexity of the algorithm as $\mathcal{O}((NDR)^2)$.

4 Experiments and Evaluation

In this section, we report our experiments to evaluate our proposed approach. We first briefly introduce our dataset and the evaluation metrics, and then present the results analysis and evaluation.

Until now, there is no “gold standard” to evaluate our approach regarding expert recommendation, to the best of our knowledge. Also, it is difficult to judgment user’s expertise manually due to the large-scale data (e.g., our test data contains more than 2 million users and nearly 20 million voting activities on 5 million posts) and the lack of ranking information in the dataset—the reputation scores of users in Stack Exchange systems are computed globally, which cannot be utilized to evaluate individual’s ability in specific domains or topics.

Similar to Huna et al. [11], we calculate the reputation score of each user by topics, according to the rules adopted by Stack Exchange⁵. We simplify the rule by removing bounty-related and edition-related reputation differences. Table 2 summarizes the simplification results. A rank can be established based on the built-in reputation scores of users, following the approach proposed by Huna et al.[11]. The rank serves as a baseline for comparative performance evaluation. Given the lack of a standard to measure verifiable expertise of users, we adopt this idea and conduct comparison experiments.

⁵ <https://stackoverflow.com/help/whats-reputation>

Table 2. Adopted reputation rules

	activity	reputation gains
	Answer is upvoted	+10
	Question is upvoted	+5
	Answer/question is downvoted	-2
	Downvote an answer	-1
	Answer is accepted	+15

4.1 Dataset and Experiment Settings

Table 3. Selected statistics profiles of experiment dataset

	# of Users	# of Posts	# of Tags	# of Votes
apple	153360	202239	1048	720540
askUbuntu	420227	598530	3022	2543467
gis	63977	179507	2221	573263
math	315792	1807772	1518	6046107
physics	95485	234583	876	1055850
serverFault	302850	645711	3514	2048746
stat	111974	195038	1331	782689
superuser	500264	859690	5190	3281616
unix	188934	284114	2438	1276409

Dataset As mentioned above, the Stack Exchange Networks includes 98 subsites and massive data. We identified 14,220,976 users, 46,575,393 posts, 178,575 tags, and 178,184,014 votes. Computing at such a scale can be challenging to any existing systems. Thus, in this work, we conducted experiments on several reasonably selected subsets, which contains a feasible yet still decent volume of data.

Note that, our method is a tree-guided tensor decomposition approach, where the tree models the hierarchical entity relationships including topics information. To keep the variance of the topics, we generate our testing subsets from several independent subsites. These subsites are named as “*apple*”, “*math*”, “*stats*”, “*askubuntu*”, “*physics*”, “*superuser*”, “*gis*”, “*serverfault*”, and “*unix*”. Some selected statistics profiles can be found at Table 3.

Due to the massive scale of our data source and its high degree of sparseness, a random sampling could end up output posts with an enormous number of unrelated users and topics. Hence, we first sample randomly to select a subset of users and then enumerations on posts tags and voting are performed. This ensures the selected posts and votes are all related to the sampled users.

4.2 Results Analysis and Evaluation

Evaluation Metrics

- **Precision@k** Precision@k is one of standard evaluation metrics in information retrieval tasks and recommender systems. It is defined to calculate the proportion of retrieved items in the top-k set that are relevant. Here our frameworks return a list of users so that the Precision@k can be calculated as follows:

$$P@k = \frac{|\{relevant_top - k_users\} \cap \{retrived_top - k_users\}|}{|\{retrived_top - k_users\}|}$$

- **MRR** The Mean Reciprocal Rank is a statistic measure for evaluating response orderly to a list, which here is average of reciprocal ranks for all tested questions:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{Rank_i}$$

Compared methods

- **Baselines** Apart from the reputation value calculated by Stack Exchange rules mentioned earlier in Table 2, it also can be found that some baselines are also often used apart from reputation value. Namely, lists generated by rank by "Best Answer Ratio" of users and rank by "Number of Answers" produced by users.
- **MF-BPR**[21] Rendel et al. introduce pairwise BPR ranking loss into standard Matrix Factorization models. It is specifically designed to optimize ranking problems.
- **Zhang et al.**[29], Z-Score by Zhang et al., is a well-known reputation measure, despite their original work is a PageRank based system and is not aimed at measurements. This feature-based score can be resolved by q the number of questions a user asked and a , the number of answers the user posted. That is,

$$Z - Score = \frac{a - q}{\sqrt{a + q}}$$

- **ConvNCF**[8] Outer Product-based Neural Collaborative Filtering, a multi-layer neural network architecture based collaborative filtering method. it use an outer product to find out the pairwise correlations between the dimensions of the embedding space.

Results Analysis Figure 6 shows the evaluation results with respect to the Precision and MRR of different methods, where precision measures the ability to find experts and MRR the performance of outputting list of experts in correct order. We observed that our approach generally outperformed other tested approaches, although some other approaches produces more accurate list when the length of the requested list is no more than 3, and this can be claimed less

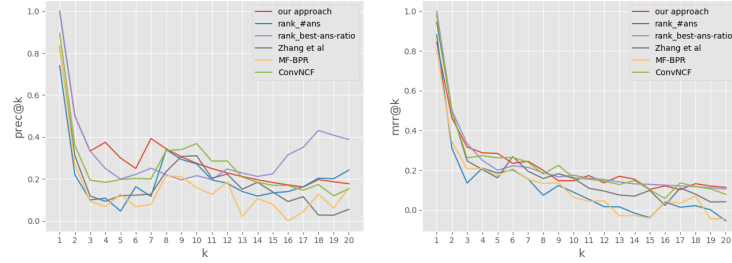


Fig. 6. Performance comparison of our approach to others, tested with 250 users and their historical data

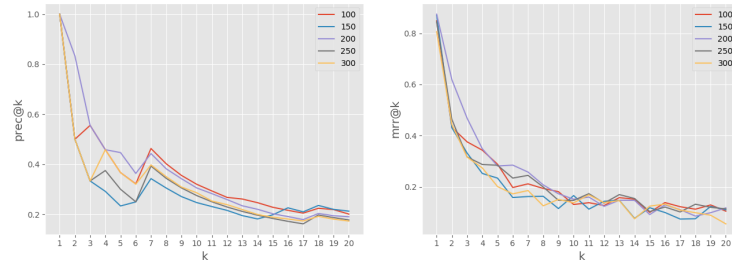


Fig. 7. Precision and MRR of tests at various number of users

likely to be practical. Our approach yielded better ranks in most cases except some case where very short lists were requested. Yet, It can be argued, in real life applications, a the list of approximately 10 or more experts is largely sensible and our approach will have substantial better performance. Also interestingly, here we can see both precision and MRR decreases by the increase of K , which differs from our experience of previous work. And a further look at the distribution of reputation in our tested data reveals it actually sensible, as we can see in Figure 8, the distribution of users' reputation is considerably uneven, given very few people high have reputation, which are our goal of output, and most people in the dataset are reputed at value 1. Additionally, to assess the stability of our approach, we conducted tests with various size of input data, ranging from 100 users to 300 users. Besides acceptable fluctuations, the results demonstrate our approach performs relatively stable, both in accuracy and quality.

5 Conclusion

In this paper, we have proposed a framework to identify experts across different collaborative networks. The framework use tree-guided tensor decomposition to exploit insights from Q&A networks. In particular, we decompose a 4th rank tensor with tree-guided lasso and matrix factorization to exploit the topic

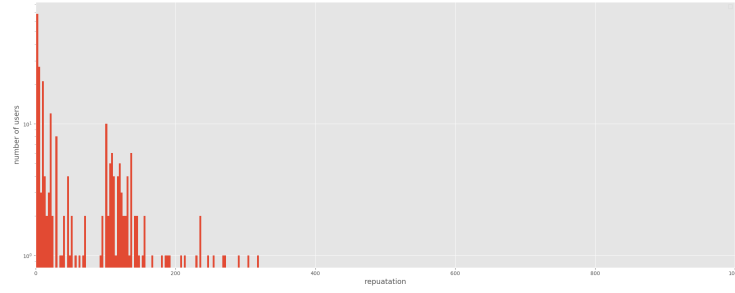


Fig. 8. Distribution of reputation of users in our dataset

information from a collection of Q&A websites in Stack Exchange Networks to alleviate the data sparsity issue. The 4th rank tensor model of the data ensures to keep as much as information as needed, which confirmed by experiments and evaluation. Due to the lack of “Gold Standard”, we compared our approach with baselines accordingly to the rank by the reputation score calculated by Stack Exchange built-in approaches on each topic. The comparison results demonstrate the feasibility of our approach. The proposed approach can be applied to broader scenarios such as finding the most appropriate person to consult on some specific problems for individuals, or identifying the desired employees for enterprises.

Acknowledgment

This research was undertaken with the assistance of resources and services from the National Computational Infrastructure (NCI), which is supported by the Australian Government.

References

1. Amatriain, X., Lathia, N., Pujol, J.M., Kwak, H., Oliver, N.: The wisdom of the few: a collaborative filtering approach based on expert opinions from the web. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. pp. 532–539. ACM (2009)
2. Balog, K., Azzopardi, L., de Rijke, M.: A language modeling framework for expert finding. *Information Processing & Management* **45**(1), 1–19 (2009)
3. Bhargava, P., Phan, T., Zhou, J., Lee, J.: Who, what, when, and where: Multi-dimensional collaborative recommendations using tensor factorization on sparse user-generated data. In: Proceedings of the 24th International Conference on World Wide Web. pp. 130–140. ACM (2015)
4. Daud, A., Li, J., Zhou, L., Muhammad, F.: Temporal expert finding through generalized time topic modeling. *Knowledge-Based Systems* **23**(6), 615–625 (2010)
5. Fang, H., Wu, F., Zhao, Z., Duan, X., Zhuang, Y., Ester, M.: Community-based question answering via heterogeneous social network learning. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)

6. Fazel-Zarandi, M., Devlin, H.J., Huang, Y., Contractor, N.: Expert recommendation based on social drivers, social network analysis, and semantic data representation. In: Proceedings of the 2nd international workshop on information heterogeneity and fusion in recommender systems. pp. 41–48. ACM (2011)
7. Ge, H., Caverlee, J., Lu, H.: Taper: A contextual tensor-based approach for personalized expert recommendation. *Proc. of RecSys* (2016)
8. He, X., Du, X., Wang, X., Tian, F., Tang, J., Chua, T.S.: Outer product-based neural collaborative filtering
9. Hidasi, B., Tikk, D.: Fast als-based tensor factorization for context-aware recommendation from implicit feedback. *Machine Learning and Knowledge Discovery in Databases* pp. 67–82 (2012)
10. Huang, C., Yao, L., Wang, X., Benatallah, B., Sheng, Q.Z.: Expert as a service: Software expert recommendation via knowledge domain embeddings in stack overflow. In: 2017 IEEE International Conference on Web Services (ICWS). pp. 317–324 (June 2017). <https://doi.org/10.1109/ICWS.2017.122>
11. Huna, A., Srba, I., Bielikova, M.: Exploiting content quality and question difficulty in cqa reputation systems. In: International Conference and School on Network Science. pp. 68–81. Springer (2016)
12. Jenatton, R., Mairal, J., Bach, F.R., Obozinski, G.R.: Proximal methods for sparse hierarchical dictionary learning. In: Proceedings of the 27th international conference on machine learning (ICML-10). pp. 487–494 (2010)
13. Kao, W.C., Liu, D.R., Wang, S.W.: Expert finding in question-answering websites: a novel hybrid approach. In: Proceedings of the 2010 ACM symposium on applied computing. pp. 867–871. ACM (2010)
14. Karatzoglou, A., Amatriain, X., Baltrunas, L., Oliver, N.: Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In: Proceedings of the fourth ACM conference on Recommender systems. pp. 79–86. ACM (2010)
15. Kim, S., Xing, E.P.: Tree-guided group lasso for multi-task regression with structured sparsity. In: Proceedings of the 27th International Conference on International Conference on Machine Learning. pp. 543–550. ICML’10, Omnipress, USA (2010), <http://dl.acm.org/citation.cfm?id=3104322.3104392>
16. Kim, S., Xing, E.P.: Tree-guided group lasso for multi-task regression with structured sparsity. In: Proceedings of the 27th International Conference on International Conference on Machine Learning. pp. 543–550. ICML’10, Omnipress, USA (2010), <http://dl.acm.org/citation.cfm?id=3104322.3104392>
17. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. *SIAM review* **51**(3), 455–500 (2009)
18. Liu, D.R., Chen, Y.H., Kao, W.C., Wang, H.W.: Integrating expert profile, reputation and link analysis for expert finding in question-answering websites. *Inf. Process. Manage.* **49**(1), 312–329 (Jan 2013). <https://doi.org/10.1016/j.ipm.2012.07.002>, <http://dx.doi.org/10.1016/j.ipm.2012.07.002>
19. Liu, X., Ye, S., Li, X., Luo, Y., Rao, Y.: Zhihurank: A topic-sensitive expert finding algorithm in community question answering websites. In: International Conference on Web-Based Learning. pp. 165–173. Springer (2015)
20. Rendle, S., Balby Marinho, L., Nanopoulos, A., Schmidt-Thieme, L.: Learning optimal ranking with tensor factorization for tag recommendation. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 727–736. ACM (2009)

21. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. pp. 452–461. AUAI Press (2009)
22. Riahi, F., Zolakhtaf, Z., Shafiei, M., Milios, E.: Finding expert users in community question answering. In: Proceedings of the 21st International Conference on World Wide Web. pp. 791–798. ACM (2012)
23. Srba, I., Bielikova, M.: Why is stack overflow failing? preserving sustainability in community question answering. *IEEE Software* **33**(4), 80–89 (2016)
24. Wang, G.A., Jiao, J., Abrahams, A.S., Fan, W., Zhang, Z.: Expertrank: A topic-aware expert finding algorithm for online knowledge communities. *Decision Support Systems* **54**(3), 1442–1451 (2013)
25. Wang, X., Huang, C., Yao, L., Benatallah, B., Dong, M.: A survey on expert recommendation in community question answering. *Journal of Computer Science and Technology* **33**(4), 625–653 (2018)
26. Xiong, L., Chen, X., Huang, T.K., Schneider, J., Carbonell, J.G.: Temporal collaborative filtering with bayesian probabilistic tensor factorization. In: Proceedings of the 2010 SIAM International Conference on Data Mining. pp. 211–222. SIAM (2010)
27. Yao, L., Sheng, Q.Z., Qin, Y., Wang, X., Shemshadi, A., He, Q.: Context-aware point-of-interest recommendation using tensor factorization with social regularization. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1007–1010. SIGIR '15, ACM, New York, NY, USA (2015). <https://doi.org/10.1145/2766462.2767794>, <http://doi.acm.org/10.1145/2766462.2767794>
28. Yao, L., Sheng, Q.Z., Wang, X., Zhang, W.E., Qin, Y.: Collaborative location recommendation by integrating multi-dimensional contextual information. *ACM Transactions on Internet Technology (TOIT)* **18**(3), 32 (2018)
29. Zhang, J., Ackerman, M.S., Adamic, L.: Expertise networks in online communities: structure and algorithms. In: Proceedings of the 16th international conference on World Wide Web. pp. 221–230. ACM (2007)