SocialTransfer: Transferring Social Knowledge for Cold-Start Crowdsourcing

Zhou Zhao[*], James Cheng, Furu Wei, Ming Zhou, Wilfred Ng, Yingjun Wu
Department of Computer Science and Engineering, Hong Kong University of Science and Technology
Department of Computer Science and Engineering, The Chinese University of Hong Kong
Microsoft Research Asia, Beijing, China
School of Computing, National University of Singapore, Singapore
{zhaozhou, wilfred}@cse.ust.hk, jcheng@cse.cuhk.edu.hk,
{fuwei,mingzhou}@microsoft.com, yingjun@comp.nus.edu.sg

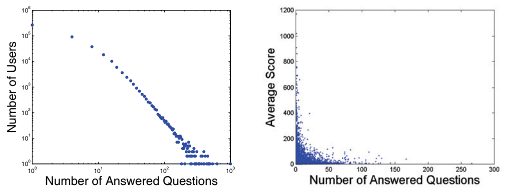
ABSTRACT

An essential component of building a successful crowdsourcing market is effective *task matching*, which matches a given task to the right crowdworkers. In order to provide high-quality task matching, crowdsourcing systems rely on past task-solving activities of crowdworkers. However, the average number of past activities of crowdworkers in most crowdsourcing systems is very small. We call the workers who have only solved a small number of tasks *cold-start crowdworkers*. We observe that most of the workers in crowdsourcing systems are cold-start crowdworkers, and crowdsourcing systems actually enjoy great benefits from cold-start crowdworkers. However, the problem of task matching with the presence of many cold-start crowdworkers has not been well studied. We propose a new approach to address this issue. Our main idea, motivated by the prevalence of online social networks, is to transfer the knowledge about crowdworkers in their social networks to crowdsourcing systems for task matching. We propose a SocialTransfer model for cold-start crowdsourcing, which not only infers the expertise of warm-start crowdworkers from their past activities, but also transfers the expertise knowledge to cold-start crowdworkers via social connections. We evaluate the SocialTransfer model on the well-known crowdsourcing system Quora, using knowledge from the popular social network Twitter. Experimental results show that, by transferring social knowledge, our method achieves significant improvements over the state-of-the-art methods.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications

[*]Part of the work was done when the author was an intern at Microsoft Research Asia



(a) Cold-Start Property (b) Task Performance

Figure 1: Cold-Start Crowdworkers in Quora

General Terms

Algorithms, Design, Experiments

Keywords

Crowdsourcing, Social Network

1. INTRODUCTION

The benefits of crowdsourcing have been widely recognized today [26, 27, 1] and we have witnessed many successful systems such as Amazon Mechanical Turk [21] for generic tasks, LabelMe [11] for image annotation, and Quora [28] for question answering. As crowdsourcing systems are gaining more and more popularity, their rapid growth has also made it increasingly difficult for crowdworkers to find tasks that are suitable for them to perform in the market.

In this paper, we study the problem of *task matching query* in crowdsourcing systems, which is to find the right crowdworkers to solve a given task. In order to provide high-quality task matching, crowdsourcing systems rely on histories of past task-solving activities of crowdworkers. However, when new crowdworkers join a system, no prior activity can be observed. In fact, a vast majority of existing crowdworkers in a real-life crowdsourcing system, including many that have joined the system for a relatively long period of time, do not have sufficient prior task-solving records for inferring high-quality task matching. To give an example, we aggregate the question-solving activities of the Quora crowdworkers in Figure 1(a), from which we can observe that the participation in the question-answering activities of most crowdworkers fall into the *Long Tail* part of the power-law curve, i.e., the majority of the crowdworkers have performed less than 10 tasks. These crowdworkers who have only performed a small number of tasks are called ***cold-start crowdworkers***.

Interestingly, the crowdsourcing market also enjoys the great benefits brought by cold-start crowdworkers. We have observed the excellent performance of cold-start crowdworkers on an extensive number of tasks. We demonstrate the task performance of the Quora crowdworkers with respect to the number of task-solving activities in Figure 1(b), where we use the thumb-ups/downs of users in Quora to measure the quality of the task performance of crowdworkers. We can see that a significant number of cold-start crowdworkers attained high scores for their tasks. One explanation for this phenomenon is that cold-start workers usually participate in tasks that they feel confident to solve. Although cold-start crowdworkers can contribute greatly to the crowdsourcing market, automatical and quality task matching for such crowdworkers is challenging in real-life crowdsourcing systems due to the lack of information to analyze them.

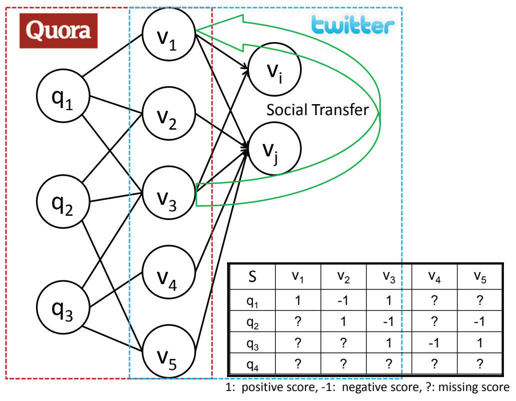
Social Knowledge for Cold-Start Crowdsourcing. The classic algorithms [31, 39, 32, 18, 25] mainly focus on task matching for warm-start crowdworkers who have solved many tasks. However, the task matching for cold-start crowdworkers still remains a challenging problem.

To attain high-quality task matching, we need to find sufficient knowledge for cold-start crowdworkers, but knowledge about them in the crowdsourcing system is scarce. Fortunately, in this big data era, we may find other sources to obtain the required knowledge. In particular, we propose a novel idea of transferring knowledge from online social networks for crowdsourcing task matching. Indeed, with the prevalence of online social networks today, it is not difficult to find the activities of crowdworkers, including both warm-start and cold-start crowdworkers, as well as their connections, in various online social networks (e.g., Facebook, Twitter, etc.). The crowdworkers broadcast messages, post blogs, and follow the activities of other users in social networks. Thus, a vast amount of data related to crowdworkers, such as workers' tweet contents and worker-to-worker social connection, can be obtained. For example, we observe that more than one third of the crowdworkers in Quora has a twitter account and crowdworkers are followed by other crowdworkers.

How can we transfer rich information hidden in online social networks to achieve high-quality task matching in crowdsourcing systems?

We focus on the social activities of the Quora crowdworkers in a popular social networking site, Twitter, and propose a ***SocialTransfer graph*** in order to transfer the social knowledge of both cold-start and warm-start crowdworkers to solve task matching. We extract the Twitter accounts of the crowdworkers from their profiles in the Quora system. We mainly utilize two types of social knowledge of the crowdworkers, namely worker's tweets and worker-to-worker social connections. We model the social interests of the crowdworkers from their tweets and infer the similarity of the crowdworkers by their worker-to-worker connections.

We illustrate our main idea using an example of a SocialTransfer graph in Figure 2. We extracted four task matching queries $Q = \{q_1, q_2, q_3, q_4\}$ and five crowdworkers $V = \{v_1, v_2, \ldots, v_5\}$ from Quora. First, the past task-solving activities of the crowdworkers on the three task matching queries are indicated by the edges between queries Q and



S	v_1	v_2	v_3	v_4	v_5
q_1	1	-1	1	?	?
q_2	?	1	-1	?	-1
q_3	?	?	1	-1	1
q_4	?	?	?	?	?

1: positive score, -1: negative score, ?: missing score

Figure 2: An Example of a SocialTransfer Graph

crowdworkers V in the rectangle on the left in Figure 2. The quality scores of the answers given by the crowdworkers for each query in Q are shown in Table S. A score of 1 indicates a positive performance of the crowdworker on the task, while a score of -1 indicates a negative performance of the crowdworker on the task. The question mark means that the crowdworker did not participate in the task. The quality score is based on the thumb-ups and thumb-downs of the users who posted the tasks or viewed the answers in Quora. Second, we show the social following relations (in Twitter) of the crowdworkers in the rectangle on the right in Figure 2. Each crowdworker in Twitter is associated with the worker's tweets and social relations. A crowdworker can follow other crowdworkers or other Twitter users. The process of building a SocialTransfer graph is to construct a heterogenous graph by integrating the social network graph of the crowdworkers into the task assignment bipartite graph.

The main idea of using a SocialTransfer graph to address the problem of task matching for cold-start crowdworkers is as follows. First, for cold-start crowdworkers with sufficient tweet information, we infer the expertise of the crowdworkers for different types of tasks by transferring the knowledge of the tweets. Second, for cold-start crowdworkers with limited tweet information, we infer their expertise by transferring the social knowledge from some warm-start crowdworkers. For example, v_1 is a cold-start crowdworker in Figure 2. Apart from inferring the expertise of v_1 from his tweets, we can also transfer the knowledge from v_3, who is a warm-start crowdworker, if v_1 and v_3 share many common followings in Twitter.

Task Matching for Cold-Start Crowdsourcing. There exists some work addressing the cold-start problem in user-item recommendation systems [23, 13, 24, 14, 40, 34, 10, 15]. However, most of them are not applicable in addressing the problem of task matching in cold-start crowdsourcing. Even though matching a task to the right crowdworker is analogous to recommending an item to a user, there are fundamental differences between them. First, the existing work incorporates the social relations of users to recommend the best item among a set of items to a user. In our work, we map a question to the right crowdworker but there is no relation among the questions; thus, the existing cold-start recommendation techniques cannot be applied to solve our problem. Second, existing recommendation techniques focus on recommending *existing* items to users, while task matching in crowdsourcing aims to find the right crowdworkers to solve *new* tasks.

The main contributions of our work are summarized as follows:

- We propose a new approach to address the problem of task matching in cold-start crowdsourcing by transferring the knowledge embedded in online social networks to crowdsourcing systems.
- We develop a SocialTransfer model for Cold-start crowdsourcing (STC) that not only infers the expertise of the warm-start crowdworkers, but also transfers the knowledge to the cold-start crowdworkers via social connections.
- We devise an efficient inference algorithm to build our STC model and propose a task matching algorithm in crowdsourcing systems based on STC.
- We evaluate the proposed SocialTransfer model on a real-life crowdsourcing system Quora. We demonstrate that, by transferring social knowledge, our approach effectively addresses the task matching problem in cold-start crowdsourcing and significantly outperforms the state-of-the-art task matching techniques.

The rest of the paper is organized as follows. Section 2 introduces the notations and formulates the problem. We propose a novel model that transfers social knowledge for cold-start crowdsourcing in Section 3. We report the experimental results in Section 4 and survey the related work in Section 5. We conclude the paper in Section 6.

2. NOTATIONS AND PROBLEM DEFINITION

We first define some frequently used notations and then formulate the problem of task matching in crowdsourcing systems. The summary of the notation is given in Table 1.

Table 1: Definition of Notations

Group	Notation	Notation Description
Data	G_{ST}	A SocialTransfer graph
	$Q = \{q_i\}_{i=1}^M$	Matching query set
	$V = \{v_i\}_{i=1}^N$	Crowdworker set
	$A = \{A_{(q_i,v_j)}\}$	Task matchings
	$S = \{S_{(q_i,v_j)}\}$	Feedback scores
	$B = \{B_{(v_i,v_j)}\}$	Common followings
	W	Words in queries and tweets
Model	$\mathbf{Dir}(\cdot)$	Dirichlet distribution
	$\mathbf{Mult}(\cdot)$	Multi-nomial distribution
	$\mathbf{N}(\cdot)$	Normal distribution
	$\overrightarrow{Q} = \{\overrightarrow{q_i}\}_{i=1}^N$	Latent query factor
	$\overrightarrow{V} = \{\overrightarrow{v_i}\}_{i=1}^M$	Latent crowdworker factor
	$\Theta = \{\theta_V, \theta_Q\}$	Topic proportions
	$Z = \{Z_V, Z_Q\}$	Topic assignments
	λ_B, λ_V	Priors

DEFINITION 1. *(Task Matching Queries Q) The set Q contains a collection of existing processed task matching queries in a crowdsourcing system. We denote by Q the set $\{q_1, q_2, \ldots, q_M\}$, where q_i is a task matching query and M is the number of queries. For each query q_i, we denote its related task description by w_{q_i}, which is a bag of words.*

For example, the description of a task matching query in Quora can be a posted question, while the description of a query in Amazon Mechanical Turk can be several instructions.

DEFINITION 2. *(Crowdworkers V) The set V consists of all the crowdworkers in a crowdsourcing system. We denote by V the set $\{v_1, v_2, \ldots, v_N\}$, where v_i is a crowdworker and N is the number of crowdworkers.*

DEFINITION 3. *(Task Matching Graph G_Q) The bipartite task matching graph, $G_Q = (Q \bigcup V, A)$, stores the existing matchings of queries Q and crowdworkers V in a crowdsourcing system. The set of vertices, $Q \bigcup V$, is the union of queries and crowdworkers. The set of edges, A, is the set of matchings between queries in Q and crowdworkers in V.*

For example, the graph G_Q for the matchings between queries $Q = \{q_1, q_2, q_3, q_4\}$ and crowdworkers $V = \{v_1, v_2, \ldots, v_5\}$ is given in Figure 2. The edge $A_{(q_i,v_j)} = 1$ if the task in query q_i is assigned to crowdworker v_j, otherwise $A_{(q_i,v_j)} = 0$.

DEFINITION 4. *(Feedback Score S) The scores S are the feedbacks given by users and are used for measuring the quality of the tasks performed by crowdworkers. For each task matching (q_i, v_j) in graph G_Q, there is a feedback score $S_{(q_i,v_j)}$ for it.*

The feedback score S can be the satisfactory rate of the task sponsor (i.e. $S_{(q_i,v_j)} \in [0,1]$) in Amazon Mechanical Turk or the thumb-ups/downs for the answers by crowdworkers in Quora (i.e. $S_{(q_i,v_j)} \in Z$).

Let $F(v_i)$ be the set of followings of a crowdworker v_i in a social network, i.e., the user v_i follows the set of users in $F(v_i)$ in the social network.

DEFINITION 5. *(Social Graph G_V) The social graph $G_V = (V, B)$ stores the activities of crowdworkers in a social network such as workers' tweet content and worker-to-worker social connection. For each crowdworker $v_i \in V$, we denote its tweet content by w_{v_i}, which is a bag of words. For each edge $(v_i, v_j) \in B$, it indicates that crowdworkers v_i and v_j share some common followings in the social network, i.e., $F(v_i) \bigcap F(v_j) \neq \emptyset$. We set the weight of the edge (v_i, v_j) to be $\frac{|F(v_i) \bigcap F(v_j)|}{|F(v_i) \bigcup F(v_j)|}$, i.e., the ratio of the common followings between v_i and v_j to their total followings in the social network.*

Note that the weight of the edges in B is within the range $[0, 1]$.

Now, we introduce a SocialTransfer graph that transfers the social knowledge of crowdworkers in a social network to address the cold-start crowdsourcing problem, in order to improve the quality of task matching.

DEFINITION 6. *(SocialTransfer Graph G_{ST}) The SocialTransfer graph $G_{ST} = (Q \bigcup V, A \bigcup B)$ models the activities of crowdworkers in both the task matching graph G_Q and the social graph G_V. The set of vertices in G_{ST} is the union of the existing task matching queries Q and crowdworkers V. The set of edges in G_{ST} contains both the set of existing matchings modeled in A and the set of pairwise common followings of crowdworkers modeled in B.*

Note that G_{ST} is not (and need not be) a tripartite graph, though we show a tripartite graph for the example in Figure 2 for simplicity of discussion there.

We now define the problem as follows.

PROBLEM 1. *(Task Matching Query in Cold-Start Crowdsourcing) Consider a crowdsourcing system with many cold-start crowdworkers and a SocialTransfer graph $G_{ST} = (Q \bigcup V, A \bigcup B)$. Given a new task matching query q, find the crowdworkers with high predicted feedback score to solve the given task.*

3. SOCIALTRANSFER MODEL

In this section, we propose our model, STC, to tackle the problem of task matching in cold-start crowdsourcing. We first introduce the background of task matching in crowdsourcing and present the idea of transferring social knowledge for cold-start crowdsourcing. Then, we summarize our proposed STC model and devise an effective inference algorithm to build STC model. Finally, we present a task matching algorithm in crowdsourcing.

3.1 Background of Task Matching

We first give the definitions of latent query factor and latent crowdworker factor for task matching queries $\overrightarrow{Q}$ and crowdworkers $\overrightarrow{V}$. Then, we propose a score generative model for the already processed tasks based on latent query factor and latent crowdworker factor.

DEFINITION 7. *(Latent Query Factor) Given the dimension of latent space K, the latent factor of the queries in crowdsourcing systems is denoted by a K-dimensional vector of real values. For each task matching query $q_i \in Q$, we denote by $\overrightarrow{q_i}$ the latent query factor, which is given by* $\overrightarrow{q_i} = \begin{pmatrix} q_{i,1} \\ \cdots \\ q_{i,K} \end{pmatrix} \in R^K$.

In this paper, we consider that the prior distribution of the latent query factor is based on the latent topic of the task description w_q. We employ a well-known latent Dirichlet allocation (LDA) [2] to discover the latent topic of the task description, for which we choose the normal distribution for the prior distribution of latent query factor, given by

$$\overrightarrow{q_i} \sim N(\theta_{q_i}, \lambda_Q^{-1}),$$

where $N(\cdot)$ is a multivariate normal distribution and θ_{q_i} is the latent topic of question q_i. We consider that θ_{q_i} is its mean and λ_Q^{-1} is standard deviation.

DEFINITION 8. *(Latent Crowdworker Factor) The latent factor of the crowdworkers in crowdsourcing systems is denoted by a K-dimensional vector of real values. For each crowdworker $v_j \in V$, we denote by $\overrightarrow{v_j}$ the latent crowdworker factor, which is given by* $\overrightarrow{v_j} = \begin{pmatrix} v_{j,1} \\ \cdots \\ v_{j,K} \end{pmatrix} \in R^K$.

We also choose the normal distribution as the prior for latent crowdworker factor. For the crowdworkers with activities in social networks, the prior distribution of them is given by

$$\overrightarrow{v_j} \sim N(\theta_{v_j}, \lambda_V^{-1}), \tag{1}$$

where θ_{v_j} is the latent topic of v_j's tweets and λ_V^{-1} is standard deviation. For other crowdworkers without activities, the prior distribution of them is given by

$$\overrightarrow{v_j} \sim N(0, \lambda_V^{-1}).$$

Using the definitions of latent query factor and latent crowdworker factor, we introduce the score generative model for processed tasks below.

DEFINITION 9. *(Score Generative Model) Given latent query factor $\overrightarrow{Q}$ and latent crowdworker factor $\overrightarrow{V}$, the feedback score on processed tasks is given by*

$$S \sim N(\overrightarrow{Q}^T \overrightarrow{V}, \lambda_S^{-1})$$

where $N(\cdot)$ is a multivariate normal distribution with mean $\overrightarrow{Q}^T \overrightarrow{V}$ and standard deviation λ_S^{-1}.

For the score of each processed task (q_i, v_j), we consider that its feedback score is generated by

$$S_{(q_i,v_j)} \sim N(\overrightarrow{q_i}^T \overrightarrow{v_j}, \lambda_S^{-1}) = N(\sum_{k=1}^{K} q_{i,k} v_{j,k}, \lambda_S^{-1}). \tag{2}$$

We now justify the score generative model as follows: We generate the latent query factor by the latent topic of the task description and the latent crowdworker factor by the latent topic of workers' tweets. Thus, we consider that the latent crowdworker factor is the topical expertise of the crowdworkers. Therefore, the score of the processed task is proportional to the dot-product of latent query factor and latent crowdworker factor.

3.2 Knowledge Transfer via Social Network

We now introduce the idea of transferring social knowledge to improve the inference of latent crowdworker factor. We mainly utilize two types of social knowledge, workers' tweets and worker-to-worker social connections.

We consider that the latent topic of workers' tweets represents the topical interests of the crowdworkers. For the cold-start crowdworkers, it is difficult to infer the latent crowdworker factor from their few task-solving activities in crowdsourcing systems. Thus, we transfer the discovered topical interests as an external source to improve the inference quality of the latent factor of the cold-start crowdworkers by Formula 1.

We also notice that the crowdworkers with similar topical interests have a lot of common following users or entities in social networks. Therefore, we utilize the common *followings* of the crowdworkers to regularize the inference of topical interests.

Consider two crowdworkers v_i and v_j in a social network. If both of them follow a number of common followings (i.e., the weight of the edge (v_i, v_j), $B_{(v_i,v_j)}$, in the social graph G_V is large), then their topic interests are similar (i.e., $|\theta_{v_i} - \theta_{v_j}|$ is small). Based on this assumption, we propose a prior distribution for the topic interests of crowdworkers with social regularization, which is a product of the Dirichlet distribution and the Normal distributions. The prior distribution for the topic interests of crowdworker v_i is given by

$$\theta_{v_i} \sim Dir(\alpha) \cdot \prod_{j \neq i} N(\theta_{v_i} - \theta_{v_j} | 0, B_{(v_i,v_j)}^{-1})^{\lambda_B}, \tag{3}$$

where the product of normal distributions of the topic interests of crowdworker v_i and other crowdworkers v_j are used for regularization.

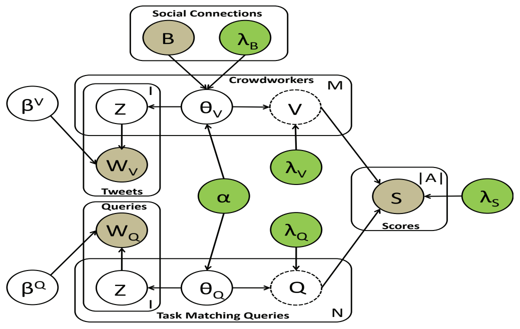


Figure 3: Graphical Representation of STC Model: grey nodes indicate observed variable and green nodes mean the prior; the dashed nodes are the variable of interest, while other nodes represent latent variable

3.3 Summary of STC Model

In this subsection, we present the generative process for STC model. The graphical representation of STC is given in Figure 3.

We notice that the contents of task matching queries and tweets are essentially represented by different kinds of languages. The study [8] shows that the words in task matching queries are formal, while the words in tweets are short, informal and abbreviated vocabularies. Particularly, tweets and task matching queries often have different words to describe the same topic. Therefore, we devise a dual language model for task matching queries and tweets based on different topic-word distributions β^q and β^v, respectively.

Referring to Figure 3, the generative process for the words in questions and tweets, latent query factor $\overrightarrow{Q}$, latent crowdworker factor $\overrightarrow{V}$, and the feedback score for task-solving activities S in STC is described as follows.

1. For each task matching query q:
   (a) Draw topic proportions $\theta_q \sim Dir(\alpha)$
   (b) For each word $w_{q,j} \in w_q$
      i. Draw a topic assignment $z_{q,j} \sim Mult(\theta_q)$
      ii. Draw a word $w_{q,j} \sim Mult(\beta^q_{z_{q,j}})$
   (c) Draw latent query factor $\overrightarrow{q} \sim N(\theta_q, \lambda_Q^{-1})$
2. For each crowdworker v:
   (a) Draw topic proportions θ_v by Equation 3
   (b) For each word $w_{v,j} \in w_v$
      i. Draw a topic assignment $z_{v,j} \sim Mult(\theta_v)$
      ii. Draw a word $w_{v,j} \sim Mult(\beta^v_{z_{v,j}})$
   (c) Draw latent crowdworker factor $\overrightarrow{v} \sim N(\theta_v, \lambda_V^{-1})$
3. For each task matching (q, v) in the processed task:
   (a) Draw feedback score $S_{(q,v)} \sim N(\overrightarrow{q}^T\overrightarrow{v}, \lambda_S^{-1})$

3.4 Inference Algorithm

In this section, we propose an inference algorithm that infers the latent query factor $\overrightarrow{Q}$, latent crowdworker factor $\overrightarrow{V}$, topic-word distributions β^q and β^v in STC.

Based on the generative process of STC, the joint distribution over feedback score S, topic proportions Θ, latent query factor $\overrightarrow{Q}$ and latent crowdworker factor $\overrightarrow{V}$ can be factorized, which is given by

$$\begin{aligned} & p(S, \overrightarrow{V}, \overrightarrow{Q}, \Theta, Z, W|\lambda_S^{-1}, \lambda_V^{-1}, \lambda_Q^{-1}, \lambda_B^{-1}, \alpha, \beta, A, B) \\ = & \; p(\theta_Q|\alpha)p(\theta_V|\alpha, B)p(Z|\Theta)p(W|Z, \beta)p(\overrightarrow{V}|\theta_V, \lambda_V^{-1}) \\ \times & \; p(\overrightarrow{Q}|\theta_Q, \lambda_Q^{-1})p(S|\overrightarrow{Q}, \overrightarrow{V}, A, \lambda_S^{-1}) \end{aligned}$$

where

$$\begin{aligned} p(\theta_Q|\alpha) &= \prod_{\theta_q} Dir(\alpha) \\ p(\theta_V|\alpha, B) &= \prod_{\theta_v} Dir(\alpha) \\ &\times \prod_{v_i, v_j \in V} N(\theta_{v_i} - \theta_{v_j}|0, B^{-1}_{(v_i,v_j)})^{\lambda_B} \\ p(Z|\Theta) &= \prod_{l\in\{v,q\}} \prod_{z_{l,j}\in Z} Mult(\theta_l) \\ p(W|Z,\beta) &= \prod_{l\in\{v,q\}} \prod_{w_{l,i}\in W} \prod_{w_{l,j}\in w_l} \beta^l_{z_{l,j}, w_{l,j}} \\ p(\overrightarrow{Q}|\theta_Q, \lambda_Q^{-1}) &= \prod_{\overrightarrow{q_i}\in\overrightarrow{Q}} N(\overrightarrow{q_i}|\theta_{q_i}, \lambda_Q^{-1}) \\ p(\overrightarrow{V}|\theta_V, \lambda_V^{-1}) &= \prod_{\overrightarrow{v_i}\in\overrightarrow{V}} N(\overrightarrow{v_i}|\theta_{v_i}, \lambda_V^{-1}) \\ p(S|\overrightarrow{Q}, \overrightarrow{V}, A, \lambda_S^{-1}) &= \prod_{A_{(q_i,v_j)}=1} N(S_{(q_i,v_j)}|\overrightarrow{q_i}^T\overrightarrow{v_j}, \lambda_S^{-1}) \end{aligned}$$

We solve the inference problem, by finding a maximum a posterior (MAP) configuration of the latent query factor $\overrightarrow{Q}$ and latent crowdworker factor $\overrightarrow{V}$ conditioning on the feedback score S, and the words in queries and tweets W. That is, we aim to find

$$\begin{aligned} & (\overrightarrow{Q}^*, \overrightarrow{V}^*) \\ = & \arg\max_{\overrightarrow{Q}, \overrightarrow{V}} p(\overrightarrow{Q}, \overrightarrow{V}|S, W, A, B, \lambda_S^{-1}, \lambda_Q^{-1}, \lambda_V^{-1}, \lambda_B^{-1}, \alpha, \beta) \end{aligned} \quad (4)$$

Maximization a posterior configuration is equivalent to maximizing the complete log likelihood $\overrightarrow{Q}$ and $\overrightarrow{V}$, given by

$$\begin{aligned} \mathcal{L} = & -\frac{\lambda_V}{2} \sum_{v_i \in V} (\overrightarrow{v_i} - \theta_{v_i})^T(\overrightarrow{v_i} - \theta_{v_i}) \\ & - \frac{\lambda_Q}{2} \sum_{q_j\in Q} (\overrightarrow{q_j} - \theta_{q_j})^T(\overrightarrow{q_j} - \theta_{q_j}) \\ & + \sum_{l\in\{v,q\}} \sum_{w_l\in W} \sum_{z_{l,j}\in z_l, w_{l,j}\in w_l} \log(\sum_{k=1}^{K} \theta_{l,k}\beta_{z_{l,j}=k, w_{l,j}}) \\ & - \frac{\lambda_B}{2} \sum_{v_j\neq v_i} B_{(v_i,v_j)}(\theta_{v_i} - \theta_{v_j})^T(\theta_{v_i} - \theta_{v_j}) \\ & - \frac{\lambda_S}{2} \sum_{A_{(q_i,v_j)}=1} (S_{(q_i,v_j)} - \overrightarrow{q_i}^T\overrightarrow{v_j})^2 \end{aligned} \quad (5)$$

where we omit the constant terms and set the prior parameters α to a vector of 1.

Algorithm 1 Task Matching Query Processing Algorithm

```
Input: A task matching query $q$, word distribution $\beta^q$ and crowdworkers $V$
Output: A ranking of crowdworkers $V'$

1: Set $\theta_q \propto$ Uniform distribution
2: for $t : 1 \rightarrow \tau_{max}$ do
3:     for each word $w_{qj} \in w_q$ do
4:         for $k : 1 \rightarrow K$ do
5:             Estimate variational parameter $\phi_{qj,k} \propto \theta_q \beta^q_{k,w_{qj}}$
6:     Sample $\theta_q \propto \prod_{j=1}^{I} \sum_{k=1}^{K} \theta_{qk}\phi_{qj,k}$
7: Estimate the latent query factor $q \sim N(\theta_q, \lambda_Q^{-1})$
8: Rank crowdworkers based on the relevance
9: return A ranking of crowdworkers $V'$
```

Then, we infer the latent crowdworker factor $\overrightarrow{V}$, the latent query factor $\overrightarrow{Q}$, the latent topics of tweets, task matching queries Θ and word distributions β^V, β^Q. We estimate these parameters by taking the derivative of the complete log likelihood $\mathcal{L}$ in Equation 5 and set it to zero.

We first report the inference of latent query factor $\overrightarrow{Q}$ and latent crowdworker factor $\overrightarrow{V}$, which is given by

$$\begin{aligned}
\overrightarrow{v_i} &\leftarrow (\lambda_S \overrightarrow{Q}\overrightarrow{Q}^T + \lambda_V I_K)^{-1}(\lambda_S \overrightarrow{Q} S_{v_i} + \lambda_V \theta_{v_i}) \\
&\sim \overrightarrow{Q} S_{v_i} + \frac{\lambda_V}{\lambda_S}\theta_{v_i} \qquad (6) \\
\overrightarrow{q_j} &\leftarrow (\lambda_S \overrightarrow{V}\overrightarrow{V}^T + \lambda_Q I_K)^{-1}(\lambda_S \overrightarrow{V} S_{q_j} + \lambda_Q \theta_{q_j}) \\
&\sim \overrightarrow{V} S_{q_j} + \frac{\lambda_Q}{\lambda_S}\theta_{q_j} \qquad (7)
\end{aligned}$$

where S_{q_i} is a diagonal matrix of the feedback scores for the workers on query q_i and S_{v_j} is also a diagonal matrix of the feedback scores for all tasks done by worker v_j.

We now justify the estimation of latent crowdworker factor and latent query factor as follows. The inference of latent crowdworker factor $\overrightarrow{v}$ is based on both its topical interests from the tweets θ_v and the latent factor of matched queries $\overrightarrow{Q}S_v$ for worker v. Similarly, the inference of latent query factor $\overrightarrow{q}$ is based on the latent topic from this query θ_q and the latent factor of the crowdworkers worked for this query $\overrightarrow{V}S_q$.

We then present the inference for the topical interests of crowdworkers θ_V, latent topics of the task matching queries θ_Q, word distributions β^V and β^Q, respectively. Unfortunately, we find that it is difficult to directly take the derivative for the complete log likelihood $\mathcal{L}$ with respect to θ_V and θ_Q, due to the decoupling between word distributions β^V, β^Q and topic assignment Z. To tackle this problem, we introduce a new variational parameter Φ for topic assignment Z and derive a lower bound, denoted by $\mathcal{L}'$. We then estimate the parameters θ_Q, θ_V, Φ, β^V and β^Q on $\mathcal{L}'$.

We derive the lower bound $\mathcal{L}'(\theta_{v_i})$ with respect to the topical interests of crowdworkers θ_{v_i} by Jensen's Inequality. The derivation for the lower bound $\mathcal{L}'(\theta_{q_j})$ with respect to the latent topic of queries θ_{q_j} is similar to the derivation of $\mathcal{L}'(\theta_{v_i})$. The derivation of $\mathcal{L}'(\theta_{v_i})$ is given by

$$\begin{aligned}
\mathcal{L}'(\theta_{v_i}) \geq & -\frac{\lambda_V}{2}(\overrightarrow{v_i} - \theta_{v_i})^T(\overrightarrow{v_i} - \theta_{v_i}) \\
& - \sum_{w_{v_i,j} \in w_{v_i}} \sum_{k=1}^{K} \phi_{v_i,j,k} \log \phi_{v_i,j,k} \\
& + \sum_{w_{v_i,j} \in w_{v_i}} \sum_{k=1}^{K} \phi_{v_i,j,k} \log(\theta_{v_i,k}\beta^V_{z_{v_i,j}=k, w_{v_i,j}}) \\
& - \frac{\lambda_B}{2} \sum_{v_j : v_j \neq v_i} B_{(v_i,v_j)}(\theta_{v_i} - \theta_{v_j})^T(\theta_{v_i} - \theta_{v_j}) = \mathcal{L}'(\theta_{v_i}).
\end{aligned}$$

We estimate the parameters ϕ and β^V by taking the derivative of $\mathcal{L}'$ with respect to them. Therefore, we report the inference of the parameters $\phi_{v_i,j,k}$ and $\beta^V_{k,w}$ by

$$\phi_{v_i,j,k} \propto \theta_{v_i,k}\beta_{k,w_{v_i,j}} \qquad (8)$$

$$\beta_{k,w} \propto \sum_{w_{v_i} \in W} \sum_{w_{v_i,j} \in w_{v_i}} \phi_{v_i,j,k} 1[w_{v_i,j} = w] \qquad (9)$$

We estimate the topical interests of crowdworkers θ_V and the latent topics of task matching queries θ_Q by using the root finding algorithm in numerical optimization tools[1].

3.5 Task Matching Algorithm on STC

We propose a task matching query processing algorithm based on our STC model in Algorithm 1. That is, given a new task matching query q, we aim to find the right crowdworkers to solve this task. Algorithm 1 first estimates the latent topic and topic assignments of query q alternatively. After that, Algorithm 1 samples the latent query factor $\overrightarrow{q}$ from its latent topic θ_q. Finally, we choose the crowdworkers based on the relevance between the latent crowdworker factor $\overrightarrow{V}$ and the latent query factor $\overrightarrow{q}$, (i.e. $\overrightarrow{V}^T\overrightarrow{q}$).

4. EXPERIMENTAL EVALUATION

The main goal of this experimental evaluation is to validate the effectiveness of our proposed STC model. To show its competitive performance, STC is compared with other four state-of-the-art approaches for task matching in crowdsourcing systems, such as Vector Space Model (VSM) [31], AuthorityRank [3], Dual Role Model (DRM) [31] and Topic Sensitive Probabilistic Model (TSPM) [39]. We implemented our algorithm in C++ and tested on machines with Linux OS Intel(R) Core(TM2) Quad CPU 2.66Hz, and 32GB RAM.

4.1 Experimental Setup

4.1.1 Datasets

We first collect the data from a popular crowdsourcing system, Quora. Quora is a question-and-answer website where questions are posted and answered by its community of crowdworkers. Quora was launched to the public in June, 2010 and has become very successful in just a few years. We first crawled the questions posted between September 2012 and August 2013, and then crawled all the workers who answered these questions. In total, we collect 444,138 questions, 95,915 crowdworkers, 887,771 answers. We then collect the data of the social activities of all 95,915 users in

[1]`http://www.gnu.org/software/gsl/`

Table 2: Summary of Datasets

Dataset	#Questions	Average #Answers
Q_1	444k	2
Q_2	178k	3.5
Q_3	86k	5.0
Q_4	48k	6.7
Q_5	30k	8.3
Q_6	20k	10.0

Twitter, which are users' tweets and following relationship. In total, we collect 29 million following users and 20GB of tweets of the crowdworkers in Twitter.

We first split the already answered questions in Quora into a training dataset and a testing dataset. We split the answered questions into six groups: Q_1, Q_2, …, Q_6, based on the number of collected answers, i.e., group Q_1 contains the questions with at least one answer. For each group Q_i, we randomly sample 100 questions as testing dataset, denoted by Q'_i. In total, we have 600 testing questions. We then keep the remaining questions in groups Q_1, Q_2, …, Q_6 as training datasets. Thus, we have generated a pair of training and testing datasets. In this experimental study, we generate ten pairs of training and testing datasets to evaluate the performance of the algorithms. We take the average of the experimental results of these algorithms on the ten pairs of datasets. The summary of the datasets is given in Table 2.

4.1.2 Measurements

We assess the performance of the task matching algorithms based on three measurements: **Precision**, **Recall** and **Cold-Start Rate**.

Precision. We employ two measurements *Accu* and $Accu_{Top1}$ to evaluate the performance of different algorithms to rank the crowdworkers who answered the questions. For each question, we consider that the crowdworker whose answer receives the highest thumb-ups is the best answerer. Both *Accu* and $Accu_{Top1}$ evaluate the ranking quality of the best answerer by different algorithms (i.e. whether the best answerer can be ranked on top), which are also widely used in [31, 39] for task matching.

Given a question q, we denote by R_V^q the ranking of the crowdworkers who answered this question. We consider that $|R_V^q|$ is the number of the crowdworkers in the ranking R_V^q. We denote by r_{best}^q the rank of the best answerer for question q by an algorithm, which is given by

$$Accu = \sum_{q \in Q'} \frac{|R_V^q| - r_{best}^q - 1}{(|R_V^q| - 1)|Q'|},$$

where Q' is the set of testing questions. *Accu* illustrates the average ranking position of the best answerer by an algorithm, where $Accu = 1$ (best) means the best answerer returned by the algorithm always ranks first while $Accu = 0$ means the opposite.

We also propose $Accu_{Top1}$ to validate whether the best answerer is ranked on top, which is given by

$$Accu_{Top1} = \frac{|\{q \in Q' | r_{best}^q \leq 1\}|}{|Q'|}.$$

Recall. We employ the measurement $Recall_{TopK}$ to evaluate the ranking quality for all crowdworkers in crowdsourcing systems by an algorithm. Given a question q, we denote by R_{TopK}^q the set of crowdworkers ranked on $TopK$ by the algorithms. The measurement $Recall_{TopK}$ is given by

$$Recall_{TopK} = \frac{|\{q \in Q' | v \in R_{TopK}^q, A_{q,v} = 1\}|}{|Q'|}.$$

Cold-Start Rate. We also investigate the types of the crowdworkers returned by an algorithm (i.e. cold-start crowdworkers or warm-start crowdworkers). In this experimental study, we consider the crowdworker who answered less than $\tau = 25$ questions as cold-start crowdworkers. We propose the measurement *Cold-Start Rate* to illustrate the type of the crowdworkers ranked on top, which is given by

$$\text{Cold-Start Rate} = \frac{|\{q \in Q' | v \in R_{Top1}^q, \sum_{q \in Q} A_{q,v} \leq \tau\}|}{|Q'|},$$

where R_{Top1}^q is the set containing the crowdworker ranked the top.

4.2 Performance Results

We first compare the effectiveness of STC with the state-of-the-art methods on the six groups of testing questions. Next, we study the impact of various model parameters on STC model: dimension of latent space K, social regularization λ_B, and other model parameters λ_V and λ_Q, respectively. Then, we illustrate the time cost of building our STC model.

4.2.1 Performance Comparison

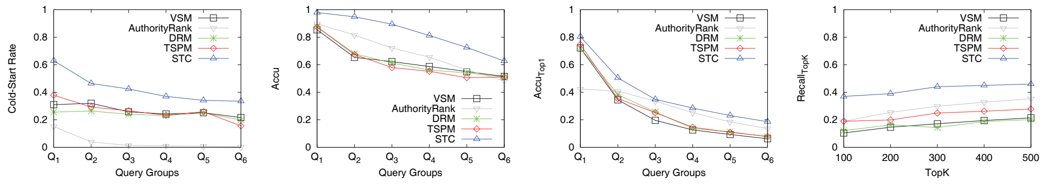
We compare STC with VSM, AuthorityRank, DRM and TSPM on six groups of queries and the results are presented in Figures 4(a) to 4(d).

We illustrate the Cold-Start Rate of the returned crowdworkers by all algorithms in Figure 4(a). The result shows that the rate of all the algorithms drops from Q'_1 to Q'_6. This is because the warm-start crowdworkers prefer solving common tasks to special tasks. We also find that the Cold-Start Rate of the crowdworkers found by STC is around 10% to 20% higher than other algorithms. The selection of the cold-start crowdworkers to solve the tasks is attributed to the effect of the knowledge transfer of STC.

Now, we show the benefits of the knowledge transfer of STC for cold-start crowdworkers using the measurements *Accu*, $Accu_{Top1}$ and $Recall_{TopK}$. We find that the performance of task matching can be greatly improved by STC.

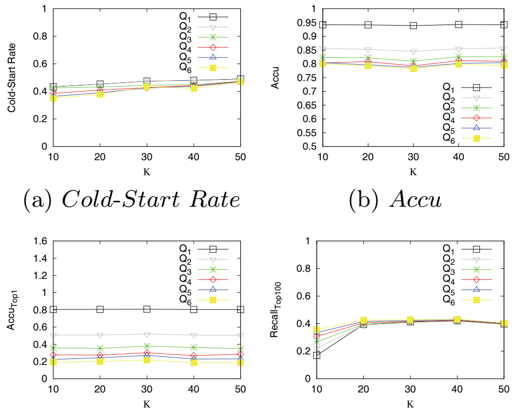
Our STC has the excellent performance on both precision measurements: *Accu* and $Accu_{Top1}$, for all the six groups of task matching queries. As shown in Figures 4(b) and 4(c), the performance of STC is 5% to 10% better than all the other algorithms. We notice that there is a number of cold-start crowdworkers having excellent performance on an extensive number of tasks in crowdsourcing systems. Using STC, the latent factor of cold-start crowdworkers can be inferred by knowledge transfer. Unlike the other classical algorithms, the high-quality cold-start crowdworkers can also be selected by STC to solve the tasks. Therefore, the precision of task matching is substantially improved.

The STC also greatly improves the performance of task matching in terms of $Recall_{TopK}$. Figure 4(d) shows that the recall of task matching is improved by 10% to 20% by STC. We find that the existing algorithms mostly select warm-start crowdworkers for solving the tasks. However, there is a significant number of the tasks that can be solved by



(a) *Cold-Start Rate* (b) *Accu* (c) $Accu_{Top1}$ (d) $Recall_{TopK}$

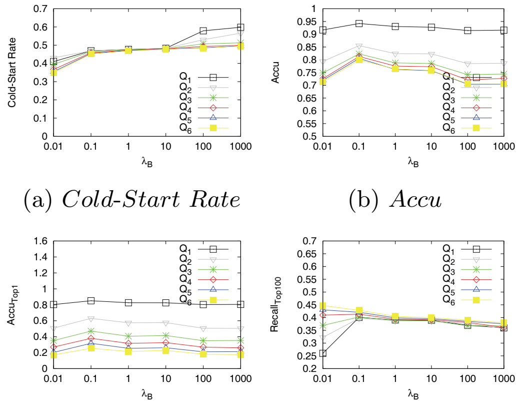
Figure 4: Performance Comparison



(a) *Cold-Start Rate* (b) *Accu*

(c) $Accu_{Top1}$ (d) $Recall_{Top100}$

Figure 5: Impact of Parameter K



(a) *Cold-Start Rate* (b) *Accu*

(c) $Accu_{Top1}$ (d) $Recall_{Top100}$

Figure 6: Impact of Parameter λ_B

cold-start crowdworkers. However, the existing algorithms do not select these cold-start crowdworkers who solve the tasks well. On the other hand, STC infers the latent factor of cold-start crowdworkers by knowledge transfer and can select both high-quality cold-start crowdworkers and warm-start crowdworkers to solve the tasks. Thus, the recall of the task matching is also improved.

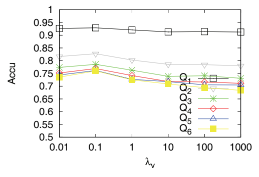
4.2.2 *Impact of Model Parameters*

We investigate the impact of various model parameters including dimension of latent space K, social regularization λ_B, parameters λ_V and λ_Q for latent crowdworker factor and latent query factor, respectively.

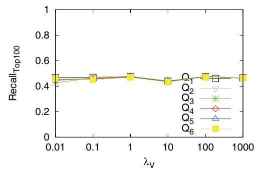
Dimension of latent space K. We study the impact of dimension of latent space on the performance of STC by varying K from 10 to 50, which is illustrated in Figures 5(a) to 5(d). Figure 5(a) shows the Cold-Start Rate of the returned crowdworkers by STC increases by 5% to 10% and then becomes convergent with respect to the dimension of latent space. By increasing the dimension of latent space, more knowledge is transferred from warm-start crowdworkers to cold-start crowdworkers. Figures 5(b) and 5(c) illustrate that the accuracy increases by 1% to 3% by varying the parameter K and then converges. We conclude that the setting $K = 50$ is good enough to represent the latent factor of crowdworkers and queries. We notice that the recall increases significantly by 10% to 20% with respect to parameter K in Figure 5(d). By transferring more knowledge to the cold-start crowdworkers, both cold-start crowdworkers and warm-start crowdworkers can be selected to solve the tasks such that the recall is greatly improved.

Social regularization λ_B. We investigate the impact of the social regularization λ_B on the performance of STC, which is illustrated in Figures 6(a) to 6(d). We vary the value of the regularization term λ_B from 0.01 to 1000. The Cold-Start Rate of the crowdworkers selected by STC first increases and then becomes convergent in Figure 6(a). The larger the parameter λ_B becomes, the more knowledge is transferred to the cold-start crowdworkers. Thus, more cold-start crowdworkers are selected for solving the tasks. The performance of STC is first improved but then declines with respect to λ_B on *Accu*, $Accu_{Top1}$ and $Recall_{Top100}$ in Figures 6(b) to 6(d). The inference of latent crowdworker factor by STC is based on both past task-solving activities in crowdsourcing systems and the activities of the crowdworkers in social networks. When the social regularization term λ_B is small, the knowledge transfer improves the performance of inferring the latent crowdworker factor. When the value of social regularization term λ_B becomes large, the inference of latent crowdworker factor is dominated by the activities of the crowdworkers in social networks. Thus, the performance of STC declines. To balance the inference from both past task-solving activities and social activities, we set the value of λ_B as a new regularization term.

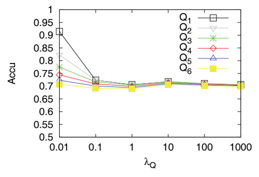
Impact of Parameters λ_V and λ_Q. We study the impact of parameters λ_V and λ_Q on the performance of STC by varying λ_V and λ_Q from 0.01 to 1000 and the results are presented in Figures 7(a) to 7(d). The Cold-Start Rate does not vary for the parameters λ_V and λ_Q and hence the result is omitted. We observe that the performance of STC declines with respect to the large value of both parameters in Figures 7(a) to 7(d), which is similar to the effect of pa-



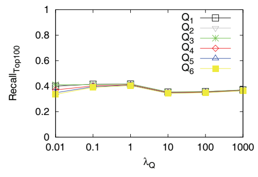
(a) *Accu*



(b) $Recall_{Top100}$

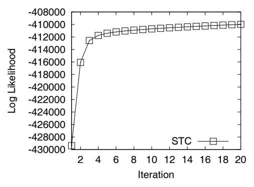


(c) *Accu*

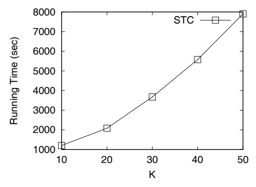


(d) $Recall_{TopK}$

Figure 7: Impact of Parameters λ_V and λ_Q



(a) Convergence



(b) Time Cost

Figure 8: Time Cost of Building STC Model

rameter λ_B. Therefore, we also set the value of λ_V and λ_Q as new regularization terms.

We remark that the overall performance of STC with three new regularization terms can also be improved by 5%, 5% and 10% on *Accu*, $Accu_{Top1}$ and $Recall_{Top100}$, respectively.

4.2.3 Running Time

We first study the running time of building our STC model on the convergence of model inference. We then investigate the time cost of model inference with respect to different dimensions of latent space K. We set the convergence threshold ϵ to 10^{-5} for inferring STC model. Figure 8(a) shows that the inference of STC model with latent space $K = 10$ converges after 20th iteration. Figure 8(b) illustrates that the time cost of inferring STC model scales linearly with respect to the dimensions of latent space K. Thus, it is efficient to build STC model on real crowdsourcing platforms such as Quora. After inferring the STC model, the time cost for matching the right crowdworkers can be computed in real time.

5. RELATED WORK

In this section, we briefly review some related work on crowdsourcing, transfer learning in the literature.

Crowdsourcing. Crowdsourcing has been widely used to solve challenging problems by human intelligence in comprehensive areas. Some successful applications that appear include CrowdDB [5] and CDAS [16].

Recently, the crowdsourcing techniques have been applied in several research areas such as database management, machine learning and information retrieval. The crowdsourcing techniques on entity resolution were studied in [29, 30]. CrowdScreen [22] applied the crowdsourcing techniques in decision making. Guo et al. [7] studied the problem of finding maximum element in the crowdsourcing databases. In [4], Davidson et al proposed the top-k and group-by queries on crowdsourcing databases. Kaplan et al [9] aimed to select the right question for planing queries. Marcus et al [17] studied the count query with the crowd. Gao et al [6] proposed crowdsourcing based online algorithm to find the ground truth. Zhao et al [35, 36] proposed crowd-selection to find the right workers to answer the questions. The work [12, 33, 20] proposed the assignment algorithms for tagging based on crowdsourcing platform.

Transfer Learning. Transfer learning techniques tackle the problem of data sparsity in the target domain by transferring supervised knowledge from other related domains, which achieve great success in a lot of applications such as classification, clustering.

Recently, Zhong et al. developed techniques [37, 38] that borrow the knowledge from the auxiliary historical activities of users in multiple networks to predict the user behavior in a target network. Mo et al. proposed cross-task crowdsourcing method [19] that transfers the knowledge cross different tasks. However, their approaches are difficult to be applied to task matching for cold-start crowdsourcing since the total number of auxiliary crowdworker activities is limited. In this work, we enrich the profile of cold-start crowdworkers by external knowledge transfer.

6. CONCLUSIONS

We studied the problem of task matching for cold-start crowdsourcing by transferring knowledge from social networks to crowdsourcing systems. We developed a Social-Transfer graph to capture the relations between tasks, crowdworkers and social networks. We then proposed a Social-Transfer model for cold-start Crowdsourcing called STC. The model not only infers the latent factor of warm-start crowdowrkers from their past task-solving activities, but also transfers the knowledge from warm-start crowdworkers to cold-start crowdworkers via social networks. We evaluated the performance of STC on the popular crowdsourcing system Quora for question-answering, by transferring knowledge from Twitter. Our results show that compared with four state-of-the-art methods for task matching in crowdsourcing systems, STC selects more cold-start crowdworkers who are the best crowdworkers to solve the given tasks, and STC improves both the precision and recall of task matching. The results confirm that our approach effectively finds the high-quality cold-start as well as warm-start crowdworkers for crowdsourcing tasks.

Acknowledgments. We thank the reviewers for giving us many constructive comments, with which we have significantly improved our paper. This research is supported in part by SHIAE Grant No. 8115048, MSRA Grant No. 6903555, and HKUST Grant No. FSGRF13EG22.

7. REFERENCES

[1] Y. Baba and H. Kashima. Statistical quality estimation for general crowdsourcing tasks. In *KDD*, pages 554–562. ACM, 2013.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[3] M. Bouguessa, B. Dumoulin, and S. Wang. Identifying authoritative actors in question-answering forums: the case of yahoo! answers. In *SIGKDD*, pages 866–874. ACM, 2008.

[4] S. B. Davidson, S. Khanna, T. Milo, and S. Roy. Using the crowd for top-k and group-by queries. In *Proceedings of ICDT*, pages 225–236. ACM, 2013.

[5] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. Crowddb: answering queries with crowdsourcing. In *Proceedings of SIGMOD*, pages 61–72. ACM, 2011.

[6] J. Gao, X. Liu, B. C. Ooi, H. Wang, and G. Chen. An online cost sensitive decision-making method in crowdsourcing systems. In *SIGMOD*, 2013.

[7] S. Guo, A. Parameswaran, and H. Garcia-Molina. So who won?: dynamic max discovery with the crowd. In *Proceedings of SIGMOD*, pages 385–396. ACM, 2012.

[8] L. Hong, G. Convertino, and E. H. Chi. Language matters in twitter: A large scale study. In *ICWSM*, 2011.

[9] H. Kaplan, I. Lotosh, T. Milo, and S. Novgorodov. Answering planning queries with the crowd.

[10] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *SIGKDD*, pages 426–434. ACM, 2008.

[11] Labelme. `http://labelme.csail.mit.edu/`.

[12] S. Lei, X. S. Yang, L. Mo, S. Maniu, and R. Cheng. itag: Incentive-based tagging. In *ICDE*, pages 1186–1189. IEEE, 2014.

[13] W.-J. Li and D.-Y. Yeung. Relation regularized matrix factorization. In *IJCAI*, pages 1126–1131, 2009.

[14] J. Lin, K. Sugiyama, M.-Y. Kan, and T.-S. Chua. Addressing cold-start in app recommendation: Latent user models constructed from twitter followers. 2013.

[15] N. N. Liu, X. Meng, C. Liu, and Q. Yang. Wisdom of the better few: cold start recommendation via representative based rating elicitation. In *RecSys*, pages 37–44. ACM, 2011.

[16] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang. Cdas: a crowdsourcing data analytics system. *Proceedings of PVLDB*, 5(10):1040–1051, 2012.

[17] A. Marcus, D. Karger, S. Madden, R. Miller, and S. Oh. Counting with the crowd. In *VLDB*, pages 109–120. VLDB Endowment, 2012.

[18] D. Mimno and A. McCallum. Expertise modeling for matching papers with reviewers. In *KDD*, pages 500–509. ACM, 2007.

[19] K. Mo, E. Zhong, and Q. Yang. Cross-task crowdsourcing. 2013.

[20] L. Mo, R. Cheng, B. Kao, X. S. Yang, C. Ren, S. Lei, D. W. Cheung, and E. Lo. Optimizing plurality for human intelligence tasks. In *CIKM*, pages 1929–1938. ACM, 2013.

[21] Mturk. `https://www.mturk.com/`.

[22] A. G. Parameswaran, H. Garcia-Molina, H. Park, N. Polyzotis, A. Ramesh, and J. Widom. Crowdscreen: Algorithms for filtering data with humans. In *Proceedings of SIGMOD*, pages 361–372. ACM, 2012.

[23] S.-T. Park and W. Chu. Pairwise preference regression for cold-start recommendation. In *RecSys*, pages 21–28. ACM, 2009.

[24] S. Purushotham, Y. Liu, and C.-C. J. Kuo. Collaborative topic regression with social matrix factorization for recommendation systems. *arXiv preprint arXiv:1206.4684*, 2012.

[25] F. Riahi, Z. Zolaktaf, M. Shafiei, and E. Milios. Finding expert users in community question answering. In *WWW*, pages 791–798. ACM, 2012.

[26] V. S. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *KDD*, pages 614–622. ACM, 2008.

[27] Y. Tian and J. Zhu. Learning from crowds in the presence of schools of thought. In *Proceedings of SIGKDD*, pages 226–234. ACM, 2012.

[28] G. Wang, K. Gill, M. Mohanlal, H. Zheng, and B. Y. Zhao. Wisdom in the social crowd: an analysis of quora. In *WWW*, pages 1341–1352. International World Wide Web Conferences Steering Committee, 2013.

[29] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. Crowder: Crowdsourcing entity resolution. *Proceedings of PVLDB*, 5(11):1483–1494, 2012.

[30] S. E. Whang, P. Lofgren, and H. Garcia-Molina. Question selection for crowd entity resolution. 2013.

[31] F. Xu, Z. Ji, and B. Wang. Dual role model for question recommendation in community question answering. In *Proceedings of SIGIR*, pages 771–780. ACM, 2012.

[32] L. Yang, M. Qiu, S. Gottipati, F. Zhu, J. Jiang, H. Sun, and Z. Chen. Cqarank: jointly model topics and expertise in community question answering. In *CIKM*, pages 99–108. ACM, 2013.

[33] X. S. Yang, R. Cheng, L. Mo, B. Kao, and D. W. Cheung. On incentive-based tagging. In *ICDE*, pages 685–696. IEEE, 2013.

[34] H. Yin, B. Cui, J. Li, J. Yao, and C. Chen. Challenging the long tail recommendation. *VLDB*, 5(9):896–907, 2012.

[35] Z. Zhao, W. Ng, and Z. Zhang. Crowdseed: query processing on microblogs. In *Proceedings of EDBT*, pages 729–732. ACM, 2013.

[36] Z. Zhao, D. Yan, W. Ng, and S. Gao. A transfer learning based framework of crowd-selection on twitter. In *SIGKDD*, pages 1514–1517. ACM, 2013.

[37] E. Zhong, W. Fan, J. Wang, L. Xiao, and Y. Li. Comsoc: adaptive transfer of user behaviors over composite social network. In *KDD*, pages 696–704. ACM, 2012.

[38] E. Zhong, W. Fan, Y. Zhu, and Q. Yang. Modeling the dynamics of composite social networks. 2013.

[39] G. Zhou, S. Lai, K. Liu, and J. Zhao. Topic-sensitive probabilistic model for expert finding in question answer communities. In *Proceedings of CIKM*, pages 1662–1666. ACM, 2012.

[40] K. Zhou, S.-H. Yang, and H. Zha. Functional matrix factorizations for cold-start recommendation. In *SIGIR*, pages 315–324. ACM, 2011.