2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)

A Task Recommendation Model for Mobile Crowdsourcing Systems based on Dwell-time

Yingjie Wang
and Xiangrong Tong
School of Computer and Control Engineering
Yantai University
Yantai, China
Email: wangyingjie@ytu.edu.cn

Zaobo He
Department of Computer Science
Georgia State University
Atlanta, GA

Yang Gao
and Kai Wang
School of Computer and Control Engineering
Yantai University
Yantai, China

***Abstract*—With the developments of mobile services, mobile crowdsourcing systems are attracting more and more attention. How to recommend user-preferred and trustful tasks for users is an important issue to improve efficiency of mobile crowdsourcing systems. This paper proposes a task recommendation model for mobile crowdsourcing systems based on dwell-time. Considering both user similarity and task similarity, the recommendation probabilities of tasks are derived. Based on dwell-time, the latent recommendation probability of tasks can be predicted. In addition, trust of tasks is obtained based on their reputations and participation frequencies. Finally, we perform comprehensive experiments towards the Amazon metadata and YOOCHOOSE data sets to verify the effectiveness of the proposed recommendation model.**

***Keywords*-component; formatting; style; styling; mobile crowdsourcing; recommendation model; dwell-time; reputation; participation frequencies**

I. Introduction

MSNs evolve from online social networks, and have extended universal and mobility features. MSNs can be regarded as communities composed of individuals with a similar feature through mobile devices (such as PADs and smart phones) [1]. It is convenient for users to share and update status anytime and anywhere. In particular, some MSN applications can share user's location information naturally, and broadcast the location information to friends [2]. At present, almost all the smart phones have GPS or triangulation functions to make user's location information to be more easily accessed.

The big data era provides new opportunities and challenges for network science. Therefore, crowdsourcing also becomes a new impetus to drive the developments of network science and engineering. Considering big data, the data flow in a network always dynamically changes at a high speed. Thus, it puts forward a stringent requirement for prompt response. In order to make MSNs to adapt to big data, the concept of *mobile crowdsourcing* emerges and brings new challenges [3]. The user communities equipped with mobile devices contain enormous potential, so many tasks can now be accomplished by users in MSNs. Utilizing mobile devices, users could collect sensing data and upload the data to a platform [4], [6]. After processing the uploaded data, the platform sends the integrated data to the requestor. A mobile crowdsourcing system can utilize the ample resources in MSNs to distribute, process, sense and integrate data. It can not only reduce the high cost for carrying out tasks but also improve the efficiency compared with traditional methods. Therefore, it is significant and promising to investigate mobile crowdsourcing systems in MSNs[5], [6]. However, how to recommend user-preferred and trustful tasks for users is very important for the efficiency of task sensing. Traditional mobile crowdsourcing systems fail to consider task recommendation, and all the users who satisfy the task requirements are assigned new tasks [7]. However, the assigned tasks may not be their interested tasks for some users. Thus, the efficiency of task sensing may be affected. In addition, for the tasks not interested by some users, they may be regarded as spam, which affects user's participation. In this paper, we investigate a preprocessing module on a cloud platform to recommend user-preferred and trustful tasks to users.

In order to accurately recommend user-preferred and trustful tasks for users, we not only need to consider users' interests, time, location and other complex information, but also need to consider trust degrees of tasks [8], [9], [10], [11]. Therefore, combining the relevant attributes of users and tasks, this paper designs a task recommendation model for mobile crowdsourcing systems based on dwell-time. The main contributions are as follows:

1) Based on dwell-time, we study users' latent interests to determine the corresponding task recommendation probability for users. This method considers the psychological aspects to investigate users' invisible characteristics in order to figure out users' latent interests.
2) To avoid recommending malicious tasks to users, the trust factor is considered to further improve the accu-

978-1-5090-3936-4/16 $31.00 © 2016 IEEE
DOI 10.1109/BDCloud-SocialCom-SustainCom.2016.36

racy of the recommendation model. Based on users' participation frequencies and reputations of tasks, we derive the trust degrees of tasks.
3) It is hard to determine the influences of different recommendation factors for the recommendation results. In order to resolve the uncertainty problem, we assign weights for different recommendation factors based on the information theory.
4) We perform comprehensive experiments towards the Amazon metadata and YOOCHOOSE data sets to verify the effectiveness and recommendation accuracy of the proposed recommendation model.

The rest of the paper is organized as follows. Section 2 presents the related works. Section 3 introduces the proposed task recommendation model for MSNs. Section 4 illustrates the experiment results. Finally, Section 5 concludes this paper.

II. Related Works

In the recent years, the study of recommendation models has become a research hotspot in online social networks. The existing models can be classified into four categories

i. *Content-based recommendation models*.

Content-based recommendation is one of the most typical recommendation methods. Utilizing users' historical task records, it computes similarities of users or extracts features from task information through statistics and machine learning methods to establish a recommendation model. There are many classical content-based recommendation methods, such as the TF-IDF method [13] and model-based recommendation methods (Bayesian classification [14], clustering, decision tree and neural network, *etc.*). Such recommendation methods work well for structured information (news, articles, *etc.*), but have poor performance for unstructured information [24].

ii. *Collaborative filtering recommendation models*.

Collaborative Filtering (CF) recommendation models recommend products to users based on similar evaluations between target users and other users [15], [32]. CF recommendation has become one of the most successful methods that can realize personalized service [16]. It was first applied to Typestry [18]. Then, it was applied to the music recommendation system Ringo [19], the joke recommendation system Jester [20], and the news CF system Grouplens. Chen *et al.* [21] proposed a generalized cross domain CF framework that seamlessly integrates social network information with cross domain data [17], [**?**]. Although as a typical recommendation technology, the CF method is employed in many applications, there are still some unsolved problems, such as sparsity and scalability.

iii. *User-product bigraph network structure based recommendation models*.

User-product bigraph network structure based recommendation models regard products and users as abstract nodes. The recommendation is based on the relationship of mutual selection between nodes [22]. Huang and Chen first proposed such a recommendation algorithm based on user-product network structure [23]. Using user-product bigraph, they established the correlation relationship between users and products. They solved the data sparsity problem by diffusion dynamics. In addition, Zhou *et al.* [25] proposed an algorithm for describing the relationships among different resources by resource allocation networks. Unfortunately, for such methods, it is difficult to extract rules, and the time complexity is high. Moreover, the personalization level is low.

iv. *Hybrid recommendation models*.

Hybrid recommendation models employ a new algorithm that combines the aforementioned recommendation algorithms. According to different combinations, the hybrid recommendation methods are divided into three groups: 1). An independent system with a recommendation model, such as the Dally Learner system. 2). A CF algorithm combining with a content-based algorithm, such as the recommendation model proposed in [26], which can improve the recommendation performance. 3). Other hybrid methods, *e.g.*, Basu *et al.* [27] established two-dimensional correlation between users and movies based on content-based and CF algorithms to predict users' preferences on movies; Jin *et al.* [28] proposed a maximum entropy web recommendation system combining collaborative and content features.

In addition to the above recommendation models, there are some other recommendation models based on correlation rules, community knowledge and so on. For example, the trust recommendation model based on colony was proposed by Bedi *et al.* [30], and the recommendation model about user cold start based on trust and distrust [31], *etc.* Yin *et al.* [33] proposed a recommendation model based on dwell time of browsing, which can reflect the opinion of a user better. Li *et al.* [34] proposed a blog recommendation mechanism that combines trust model, social relation, and semantic analysis. They illustrated how it can be applied to a prestigious online blogging system. These models have promoted the development of recommendation models.

Unfortunately, for MSNs, there are very few task recommendation models. This paper summarizes the main problems of the existing recommendation models considering the characteristics of task sensing in mobile crowdsourcing systems as follows.

1) When computing similarity of users' behaviors, most recommendation models fail to consider the recommendation attributes, so as to affect the accuracy of the recommendation results.
2) Most recommendation models fail to consider latent features of users, such as dwell-time related to users' clicks on different tasks.
3) Trust is an important factor when establishing recommendation models, especially for mobile crowdsourc-

ing systems. Therefore, it is necessary to consider the trust degree of tasks to guarantee the effectiveness of recommendation. However, it is not considered in the existing recommendation models.

4) Most recommendations models ignore the uncertainty issue caused by the many recommendation factors affecting the accuracy of the recommendation results.

III. The Proposed Recommendation Model

In this section, we introduce the proposed recommendation model. Through combining user similarity and task similarity, the task candidates are determined. Based on dwell-time in the psychology theory, we investigate the latent factor of users to establish the recommendation model. In addition, the trust degree of a task is calculated based on users' participation frequency and reputation of the task. Finally, when establishing the comprehensive task recommendation model, the weights of the recommendation factors are calculated based on information entropy in order to resolve the uncertainty problem of different recommendation factors.

In mobile crowdsourcing systems, users with mobile services play the most important roles for task sensing. Let $U = (u_1, u_2, ..., u_i, ..., u_m)$ be the user set where u_i represents user i. The task requester set is denoted as $Re = (r_1, r_2, ..., r_k, ..., r_s)$ where r_k represents requester k. The task set is denoted as $Ta = (ta_1, ta_2, ..., ta_j, ..., ta_n)$ where ta_j represents task j. Tasks belong to different categories in mobile crowdsourcing systems. Therefore, we use $Ct = (C_1, C_2, ..., C_l, ..., C_w)$ to denote the task category set, where C_l represents task category l. For all ta_j, it belongs to at least one task category.

A. Task Recommendation Probability based on Similarity

We compute user similarity utilizing Pearson Correlation Coefficient. When considering similarity of users' behaviors, most traditional recommendation methods ignore the characteristics of the recommendation attributes, such as task reputation and users' participation frequency of a task. The recommendation attributes have different influences for similarity calculation, thus the accuracy of the recommendation results is constrained. We compute user similarity based on recommendation attributes. In our recommendation model, two recommendation attributes are considered: task reputation R and users' participation frequency of a task F. The similarity between u_a and u_i can be derived as follows.

$$W_{a,i} = \frac{\sum\limits_{ta_j \in S_{a,i}} \sqrt{\lambda \cdot \frac{1}{R_j^2} + (1-\lambda) \cdot \frac{1}{F_j^2}} \cdot (r_{a,j} - \overline{r_a}) \cdot (r_{i,j} - \overline{r_i})}{\sigma_a \cdot \sigma_i} \tag{1}$$

where R_j and F_j respectively represent the reputation and users' participation frequency for ta_j. λ and $(1-\lambda)$ represent the influence degrees of reputation R_j and frequency F_j for the similarity result respectively. $S_{a,i}$ is the set of tasks in which both u_a and u_i ever participated. The average rating of u_a for his participated tasks is $\overline{r_a}$. Similarly, $\overline{r_i}$ is the average rating of u_i for his participated tasks. $r_{a,j}$ is the rating of ta_j evaluated by u_a, and $r_{i,j}$ is the rating of ta_j evaluated by u_i. σ_a and σ_i are the standard deviations of u_a and u_i respectively, whose derivations are shown below.

$$\sigma_a = \sqrt{\sum_{ta_k \in S_{a,i}} (r_{a,k} - \overline{r_a})^2}, \sigma_i = \sqrt{\sum_{ta_k \in S_{a,i}} (r_{i,k} - \overline{r_i})^2} \tag{2}$$

where ta_k is the task involving both u_a and u_i.

The similar user set of target user u_i is $Sim_i = (su_1^i, su_2^i, ..., su_K^i)$, where K is the threshold of the number of the similar users. The similar user set includes the target user himself, and its similarity degree is set to be 1. Based on the historical records of the similar users, the recommendation probabilities for different task categories are derived as follows.

$$RC_h^i = \frac{\sum\limits_{a=1}^{K} \left(W_{a,i} \cdot \frac{|C_h^a|}{\sum\limits_{h=1}^{w} |C_h^a|} \right)}{K} \tag{3}$$

RC_h^i is the recommendation probability that the recommended task belongs to task category C_h for u_i. $|C_h^a|$ indicates the number of C_h tasks in which su_a ever participated according to the historical records of su_a. Thus, $\sum\limits_{h=1}^{w} |C_h^a|$ is the total number of the tasks in which su_a ever participated.

B. Task Recommendation Probability based on Dwell-time

When a user clicks the request web of a task, regardless whether the user participates in this task, there is a dwell-time. For example, when shopping for clothes on a website, assume a user's dwell-time is 1 minute on the first web page and 5 minutes on the second web page. If we recommend the second one to the user, the recommendation result should be better. Similar with online social networks, in mobile crowdsourcing systems, we can utilize the latent features of users to predict their latent interests. This paper studies the latent interests of users based on users' psychological characteristics in order to improve the accuracy of the task recommendation model.

Yin *et al.* [33] investigated user's dwell-time distribution for JokeBox which is a popular iPhone application. It is shown that user's dwell-time follows the log-Gaussian distribution. The distribution is shown in Fig.1. According to the historical information of user's dwell-time, we can compute the average dwell-time for different task categories, *i.e.*, the threshold of dwell-time. The calculation method is shown below.

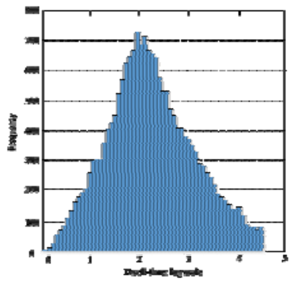

Figure 1. The distribution of dwell-time.

$$T_h^i = \frac{\sum_{j=1}^{s_h^i} t_j^i}{s_h^i} \quad (4)$$

where T_h^i is the dwell-time that u_i spent on C_h. t_j^i is the dwell-time that u_i spent on ta_j, $ta_j \in C_h$. s_h^i is the number of the tasks belonging to C_h that u_i ever browsed. Based on Eq.(4), we can obtain the average dwell-time for different categories of tasks. From Fig.1, it can be seen that dwell-time follows the log-Gaussian distribution. When a new task is distributed, the platform should first determine which category the task belongs to. Then the probability of the user-spent dwell-time is predicted based on the task category and average dwell-time of this task category, which is shown as follows.

$$\begin{aligned} p_h^i\left(t\left|T_h^i, r^i, \alpha^i\right.\right) &= N\left(\log t; \mu\left(T_h^i, r^i, \alpha^i\right), \sigma^2\right) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(\log t - \mu)^2}{2\sigma^2}} \end{aligned} \quad (5)$$

where r^i is u_i's reading speed, α^i indicates the quality sensitivity of u_i for tasks. $p_h^i\left(t\left|T_h^i, r^i, \alpha^i\right.\right)$ is the probability that u_i will spend t dwell-time on C_h tasks under the condition of known T_h^i, r^i and α^i. We set $t = T_h^i$, thus the task recommendation model based on dwell-time can be derived by Eq.(5).

C. Trust of Tasks

Different tasks issued by different requesters have different trust degrees. Therefore, *trust* is an important factor for recommendation results. Because of the openness of MSNs, the reliability of tasks cannot be guaranteed. If a system recommends distrustful tasks to users, the recommendation results may deviate from users' real demand gravely. In order to solve this problem, we study how to measure the trust degree of a task based on the requester's reputation R' and users' participation frequency for the tasks posted by requester F'.

If the reputation and participation frequency are larger, the task is more trustful. Assume that a new task ta_j is posted by r_k, then the trust degree of ta_j is

$$Trust_j = \rho \cdot {R_k}' + (1-\rho) \cdot \frac{{F_k}'}{\overline{F}} \quad (6)$$

where ${R_k}'$ is the reputation of r_k, and ${F_k}'$ indicates users' participation frequency for the tasks posted by r_k. $\overline{F}$ is system dimensionless defined by the system in order to make $\frac{{F_k}'}{\overline{F}}$ to belong to interval $[0, 1]$. In addition, ρ is the influence factor of the requester's reputation, so $1-\rho$ is the influence factor of users' participation frequency.

Combining similarity of users, dwell-time and trust, the task recommendation model for mobile crowdsourcing systems is established. When a new task $ta_j \in C_h$ is issued, the recommendation probability of ta_j for u_i is

$$R_{i,j}^* = \omega_1 \cdot RC_h^i + \omega_2 \cdot p_h^i\left(t\left|T_h^i, r^i, \alpha^i\right.\right) + \omega_3 \cdot Trust_j \quad (7)$$

where ω_1, ω_2 and ω_3 are the weights of different recommendation factors, and $\omega_1 + \omega_2 + \omega_3 = 1$. In order to determine the weights, we need to solve the uncertainty problem of influence degrees.

D. How to Determine Recommendation Weights?

Through introducing thermodynamic entropy into information theory, Shnnon proposed the concept of information entropy. Information entropy is an important tool for solving the uncertainty problem. Before an event occurs, information entropy can be seen as the measurement for uncertainty of results. After the event, information entropy indicates the measurement of information quantity derived from this event. Therefore, information entropy represents a measurement of uncertainty or information quantity. Before discussing the calculation method of recommendation weights, we first introduce how to compute information entropies for different recommendation factors, which is shown below.

$$H(X_i) = -X_i \cdot \log_2 X_i - (1-X_i) \cdot \log_2(1-X_i) \quad (8)$$

where X_i $(i \in \{1,2,3\})$ represents the value of a recommendation factor. The corresponding relationships between X_i and different recommendation factors are shown as: $X_i \rightarrow RC_h^i$, $X_2 \rightarrow p_h^i\left(t\left|T_h^i, r^i, \alpha^i\right.\right)$, and $X_3 \rightarrow Trust_j$. $H(X_i)$ is the information entropy of the corresponding recommendation factor. If the information entropy is bigger, its uncertainty is more obvious. For example, if $X_1 = 0.9$ and $X_2 = 0.5$, then their information entropies are $H(X_1) = 0.47$ and $H(X_1) = 1$. From the results, we can see that the uncertainty of X_2 is more obvious than that of X_1. Therefore, for a task recommendation model, the recommendation factor with bigger uncertainty has less influence to the recommendation results. So the corresponding weight should be smaller. On the other hand, we assign a bigger weight for the recommendation factor that has smaller uncertainty.

In this paper, we utilize classification distinctness $C(X_i)$ to represent the influence degree of a recommendation factor, which is shown below.

$$C(X_i) = \begin{cases} 1-\exp(-\pi)\cdot H(X_i), X_i \geq 0.5 \\ \exp\left(-\frac{1}{X_i}\right), \qquad X_i < 0.5 \end{cases}, \ i \in \{1,2,3\} \quad (9)$$

where $\exp(-\pi)$ is the system adjustment factor. $\pi = 1$ in this paper. Therefore, the values of different recommendation factors can be derived as follows.

$$\omega_i = \frac{C(X_i)}{\sum_{j=1}^{3} C(X_j)}, \ i \in \{1,2,3\} \quad (10)$$

Based on Eq.(10), we can determine the corresponding weights for different recommendation factors.

E. Dwell-time based Recommendation Algorithm

Our recommendation model employs the dwell-time based recommendation algorithm (DTRA) with the following four steps.

1) Compute recommendation probability based on user similarity and task similarity.
2) Compute recommendation probability based on dwell-time.
3) Compute trust degrees of tasks based on requester's trust attributes.
4) Combine the three recommendation factors to compute comprehensive recommendation results. In addition, the weights are computed based on information entropy.

In DTRA, we compute the recommendation probability of ta_j for u_i as an example. First, we give the specific description for the calculation of recommendation probability based on user similarity, which is shown in Algorithm 1.

Algorithm 1 Recommendation probability based on user similarity.

Input:
R_j, F_J, λ, $r_{a,j}$, $r_{i,j}$, $\overline{r_a}$, $\overline{r_i}$, $r_{a,k}$, $r_{i,k}$, K, $|C_h^a|$, where $k \in S_{a,i}$;
Output:
RC_h^i;
1: Compute the values of σ_a and σ_i through Eq.(2).
2: Compute $W_{a,i}$ between u_a and u_i through Eq.(1).
3: Compute recommendation probability RC_h^i based on user similarity through Eq.(3).
4: **return** RC_h^i.

Then, we give the specific description for the calculation of recommendation probability based on dwell-time, which is shown in Algorithm 2.

Algorithm 2 Recommendation probability based on dwell-time.

Input:
t_j^i, s_h^i, r^i, α^i;
Output:
$p_h^i\left(t\left|T_h^i, r^i, \alpha^i\right.\right)$;
1: Compute average dwell time T_h^i Through Eq.(4).
2: Let $t = T_h^i$.
3: Compute recommendation probability $p_h^i\left(t\left|T_h^i, r^i, \alpha^i\right.\right)$ based on dwell-time through Eq.(5).
4: **return** $p_h^i\left(t\left|T_h^i, r^i, \alpha^i\right.\right)$.

In order to guarantee the trust of the recommended tasks, we compute the trust degrees of the tasks, which is shown in Algorithm 3.

Algorithm 3 Trust degree of a task.

Input:
${R_k}'$, ${F_k}'$, $\overline{F}$, ρ;
Output:
$Trust_j$;
1: Compute $Trust_j$ of ta_j through Eq.(6).
2: **return** $Trust_j$.

Through combining the three recommendation factors, we can derive the comprehensive recommendation probability. In addition, the weights are computed based on information entropy in order to solve the uncertainty problem. The specific algorithm is shown in Algorithm 4.

Algorithm 4 Comprehensive recommendation probability.

Input:
RC_h^i, $p_h^i\left(t\left|T_h^i, r^i, \alpha^i\right.\right)$, $Trust_j$;
Output:
$R_{i,j}^*$;
1: Compute $H(X_1)$, $H(X_2)$ and $H(X_3)$ through Eq.(8).
2: Compute $C(X_1)$, $C(X_2)$ and $C(X_3)$ through Eq.(9).
3: Compute ω_1, ω_2 and ω_3 through Eq.(10).
4: Compute $R_{i,j}^*$ through Eq.(7).
5: **return** $R_{i,j}^*$.

The above algorithms describe the proposed task recommendation model based on dwell-time for mobile crowdsourcing systems. Based on DTRA, we can provide user-preferred and trustful tasks for users in mobile crowdsourcing systems.

IV. Experiments and analysis

In order to verify the effectiveness and recommendation accuracy of the proposed recommendation model, we perform some experiments utilizing Amazon metadata and YOOCHOOSE data sets and compare our model with a collaborative filtering recommendation (CFR) model [16] and a hybrid recommendation (HR) model [26].

A. Data Sets

The Amazon data set is collected by crawling the Amazon website and contains product metadata and review information about 548,552 different products (books, music CDs, DVDs and VHS video tapes). For each product, the following information is available: title, sales rank, list of similar products, detailed product category, and product reviews (time, customer, rating, number of votes, and number of people that found the review helpful). We verify the effectiveness of the proposed recommendation model in terms of user similarity and trust towards this data set.

The YOOCHOOSE data set encapsulates users' *click events* and purchase records, thus we can verify the relationships between *click event* and purchase records. YOOCHOOSE is providing a collection of sequences of click events and click sessions. For some sessions, there are also buying events. The goal is hence to predict whether a user (a session) is going to buy something or not, and if he is buying, what would be the items he is going to buy. Therefore, based on the features of the YOOCHOOSE data set, we can verify the effectiveness of the proposed recommendation model after adding the dwell-time factor through comparison experiments.

B. Evaluation Metrics

Precision, *recall* and F_1*-measure* are three main evaluation metrics. They are standard measurements for the classification accuracy of a recommendation model. They can reflect the accuracy of recommendation results. The calculation method of precision is shown below.

$$precision = \frac{1}{M} \times \sum_{i=1}^{M} \frac{N_i}{L_i} \tag{11}$$

where N_i is the number of the products that u_i has purchased in the recommendation list. L_i is the length of the recommendation list, and M is the number of the users. Similarly, recall can be derived by Eq.(12), where B_i indicates the number of the products that u_i likes. Recall represents the probability that the recommended product will be liked by u_i.

$$recall = \frac{1}{M} \times \sum_{i=1}^{M} \frac{N_i}{B_i} \tag{12}$$

For comprehensive evaluation, we also utilize F_1-measure to evaluate the effectiveness of the proposed recommendation model. F_1-measure combines precision and recall. The calculation method of F_1-measure is shown as follows.

$$F_1 = \frac{2 \times precision \times recall}{precision + recall} \tag{13}$$

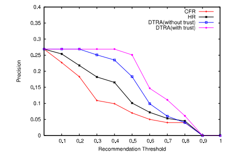

Figure 2. Precision comparison results for user similarity and trust.

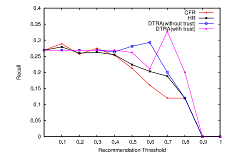

Figure 3. Recall comparison results for user similarity and trust.

C. Evaluation for User Similarity and Trust

In these experiments, we adopt the Amazon metadata to evaluate user similarity and trust. Fig.2 shows the precision comparison results of CFR, HR, DTRA without the trust factor and DTRA with the trust factor. From Fig.2, it can be seen that DTRA has better precision compared with CFR and HR. Compared with DTRA without the trust factor, the DTRA with the trust factor has better precision. In addition, we can see that when the recommendation threshold is smaller than 0.6, the recommendation results are better.

We also perform experiments to evaluate recall. Fig.3 shows the comparison results. It can be seen that DTRA has better recall compared with CFR and HR. When the recommendation threshold stays in $[0.6, 0.8]$, DTRA with the trust factor has some fluctuations. The reason is that the experiment data are extracted from real data, and there may be some noises and uncertain factors. However, from the overall trend, DTRA with the trust factor still has better recall compared with the other algorithms. In addition, when the recommendation threshold is smaller than 0.8, the recommendation results are better.

In order to comprehensively evaluate the effectiveness of the proposed recommendation model, we utilize F_1-measure to compare the recommendation results for CFR, HR, DTRA without the trust factor and DTRA with the trust factor. Fig.4 shows the comparison results, from where we can see that DTRA has better F_1-measure. Compared with other algorithms, DTRA with the trust factor has better F_1-measure. In addition, when the recommendation threshold is smaller than 0.6, the F_1-measure is better. Through the above comparison experiments, the effectiveness and recommendation accuracy of DTRA is verified comprehensively.

D. Evaluation for Dwell-time

In order to evaluate the dwell-time factor, we adopt the YOOCHOOSE data set with respect to precision, recall and F_1-measure. In the YOOCHOOSE data set, users' click events are collected during a period of time. Unfortunately,

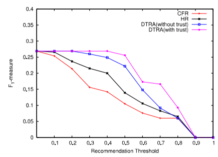

Figure 4. F_1-measure comparison results for user similarity and trust.

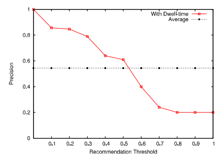

Figure 5. Precision comparison results for dwell-time.

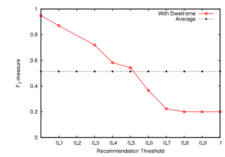

Figure 7. F_1-measure comparison results for dwell-time.

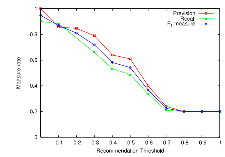

Figure 8. Effectiveness verification for dwell-time.

the dwell-time records are not collected. Thus, we map the number of the click events with dwell-time, *i.e.*, if the number of the click events is bigger, dwell-time is longer.

Fig.5 shows the precision result of DTRA with dwell-time. It can be seen that with the increase of the recommendation threshold, the recommendation precision decreases. The precision result stays in $[0.2, 1]$, and the average precision is 0.544. It can be seen that DTRA with dwell-time can perform well on the aspect of precision.

We also design an experiment for verifying the effectiveness of DTRA with dwell-time on the aspect of recall. The result is shown in Fig.6. It can be seen that the trend of recall is similar with the trend of precision. With the increase of the recommendation threshold, the recommendation recall decreases. The recall result stays in $[0.2, 0.9]$, and the average recall is 0.493. Similarly, after considering user's dwell-time, the recommendation result is improved on the aspect of recall.

We verify the effectiveness of DTRA with dwell-time on the aspect of F_1-measure to comprehensively evaluate the effectiveness of dwell-time. Fig.7 shows the F_1-measure result, which comprehensively reflects precision and recall. The F_1-measure result has similar trend with the results of precision and recall. The F_1-measure result stays in $[0.2, 1]$, and the average F_1-measure is 0.514. Thus, after adding the dwell-time factor, the recommendation result is improved on the aspect of F_1-measure.

In order to clearly present the recommendation results considering the dwell-time factor, we give the comparison results on the aspects of precision, recall and F_1-measure,

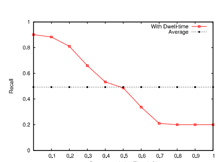

Figure 6. Recall comparison results for dwell-time.

which is shown in Fig.8. We can see that by considering the dwell-time factor, users' interests could be predicted more accurately.

V. Conclusions

To improve efficiency of mobile crowdsourcing systems, it is necessary to recommend user-preferred and trustful tasks for users. This paper proposes a task recommendation model for mobile crowdsourcing systems based on dwell-time. First, we determine the candidates for sensing a task through combining users' similarity and tasks' similarity from historical records, and compute the recommendation probability correspondingly. Then, in order to more accurately recommend user-preferred tasks for users, we compute the recommendation probability based on dwell-time. In addition, in order to recommend trustful tasks for users, we calculate the trust degrees of tasks according to historical records. The proposed task recommendation model is established through considering the above factors comprehensively. The uncertainty of weights is taken care of through the information entropy theory. Finally, through the comparison experiments towards the Amazon metadata and YOOCHOOSE data sets, the effectiveness and recommendation accuracy of the proposed recommendation model are verified.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grants No.61502410, No.61572418, No.61403329, No.61272186, No.61170224 and No.61472095, the Natural Science Foundation of Shandong Province under Grant No.ZR2013FQ020.

References

[1] M. Salehan and A. Negahban, *Social networking on smartphones: When mobile phones become addictive*, Computers in Human Behavior, 2013, 29(6): 2632-2639.

[2] X. Zheng, Z. Cai, J. Li and H. Gao, *An Application-aware Scheduling Policy for Real-time Traffic*, The 35th IEEE International Conference on Distributed Computing Systems, Columbus, Ohio, USA, 2015.

[3] S. Yang and J. Thormann, *Poster: crowdsourcing to smartphones: social network based human collaboration*, The 15th ACM international symposium on Mobile ad hoc networking and computing, ACM, New York, 2014.

[4] Z. Duan, M. Yan, Z. Cai, X. Wang, M. Han and Y. Li, *Truthful Incentive Mechanisms for Social Cost Minimization in Mobile Crowdsourcing Systems*, Sensor, 2016, 16(4).

[5] Y. Wang, Z. Cai, G. Yin, Y. Gao, X. Tong and G. Wu, *An incentive mechanism with privacy protection in mobile crowdsourcing systems* Computer Networks, 2016, 102: 157-171.

[6] J. Li, Z. Cai, M. Yan and Yingshu Li, *Using Crowdsourced Data in Location-based Social Networks to Explore Influence Maximization*, Proceedings of the 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 2016.

[7] Y. Wang, G. Yin, Z. Cai, Y. Dong and H. Dong, *A Trust-based Probabilistic Recommendation Model for Social Networks*, Journal of Network and Computer Applications, 2015, 55: 59-67.

[8] L. Zhang, Z. Cai and X. Wang, *FakeIt: Privately Releasing User Context Streams for Personalized Mobile Applications*, IEEE Transactions on Network and Service Management, 2016, 13(2): 335-348.

[9] Z. He, Z. Cai and X. Wang, *Modeling Propagation Dynamics and Developing Optimized Countermeasures for Rumor Spreading in Online Social Networks*, Proceedings of the 35th IEEE International Conference on Distributed Computing Systems, Columbus, Ohio, USA, 2015.

[10] Z. He, Z. Cai, Q. Han, W. Tong, L. Sun and Y. Li, *An Energy Efficient Privacy-Preserving Content Sharing Scheme in Mobile Social Networks*, Personal and Ubiquitous Computing, 2016.

[11] Z. He, Z. Cai, J. Yu, X. Wang, Y. Sun and Y. Li, *Cost-efficient Strategies for Restraining Rumor Spreading in Mobile Social Networks*, IEEE Transactions on Vehicular Technology, 2016.

[12] M. Han, M. Yan, et al., *Neighborhood-based uncertainty generation in social networks.* Journal of Combinatorial Optimization, 2014. 28(3): p. 561-576.

[13] O. Chum, J. Philbin and A. Zisserman, *Near Duplicate Image Detection: min-Hash and tf-idf Weighting*, BMVC, 2008, 810: 812-815.

[14] X. Yang, Y. Guo and Y. Liu, *Bayesian-Inference-Based Recommendation in Online Social Networks*, IEEE Transactions on Parallel and Distributed Systems, 2013, 24(4): 642-651.

[15] Y. M. Li, H. W. Hsiao and Y. L. Lee, *Recommending social network applications via social filtering mechanisms*, Information Sciences, 2013, 239: 18-30.

[16] I. Konstas, V. Stathopoulos and J. M. Jose, *On social networks and collaborative recommendation*, Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, Boston, MA, USA, 2009.

[17] M. Han, M. Yan, Z. Cai, and Y. Li, *An exploration of broader influence maximization in timeliness networks with opportunistic selection*. Journal of Network and Computer Applications, vol. 63, pp. 39–49, 2016.

[18] D. Goldberg, D. Nichols and B. M. Oki, et al, *Using collaborative filtering to weave an information tapestry*, Communications of the ACM, 1992, 35(12): 61-70.

[19] H. C. Chen and A. L. P. Chen, *A music recommendation system based on music data grouping and user interests*, Proceedings of the tenth international conference on Information and knowledge management. ACM, 2001: 231-238.

[20] K. Goldberg, T. Roeder and D. Gupta, *Eigentaste: A constant time collaborative filtering algorithm*, Information Retrieval, 2001, 4(2): 133-151.

[21] W. Chen, W. Hsu and M. L. Lee, *Making recommendations from multiple domains*, Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, 2013.

[22] L. Guo, J. Ma, Z. Chen and H. Jiang, *Incorporating item relations for social recommendation*, Chinese Journal of Computers, 2014, 37(1): 219-228.

[23] Z. Huang, W. Chung and H. Chen, *A graph model for E-commerce recommender systems*, Journal of the American Society for information science and technology, 2004, 55(3): 259-274.

[24] M. Han, W. Zhang and J. Li, *RAKING: An efficient K-maximal frequent pattern mining algorithm on uncertain graph database*, Jisuanji Xuebao (Chinese Journal of Computers), vol. 33, pp. 1387–1395, 2010.

[25] T. Zhou, J. Ren, M. Medo and Y. C. Zhang, *Bipartite network projection and personal recommendation*, Physical Review E, 2007, 76(4): 046115.

[26] Ekkawut Rojsattarat and Nuanwan Soonthornphisaj, *Hybrid Recommendation: Combining Content-Based Prediction and Collaborative Filtering*, Proceedings of the 4th International Conference, Hong Kong, China, 2003.

[27] C. Basu, H. Hirsh and W. Cohen, *Recommendation as classification: Using social and content-based information in recommendation*, AAAI/IAAI, 1998: 714-720.

[28] X. Jin, Y. Zhou and B. Mobasher, *A maximum entropy web recommendation system: combining collaborative and content features*, Proceedings of the 11th ACM SIGKDD international conference on Knowledge discovery in data mining, Chicago, IL, USA, 2005.

[29] M. Han, M. Yan, Z. Cai, Y. Li, X. Cai, and J. Yu, Influence maximization by probing partial communities in dynamic online social networks. Transactions on Emerging Telecommunications Technologies, 2016, DOI: 10.1002/ett.3054.

[30] P. Bedi and R. Sharma, *Trust based recommender system using ant colony for trust computation*, Expert Systems with Applications, 2012, 39(1): 1183-1190.

[31] M. Jamali and M. Ester. *A matrix factorization technique with trust propagation for recommendation in social networks*, Proceedings of the fourth ACM conference on Recommender systems. ACM, 2010: 135-142.

[32] M. Han, M. Yan, et al., *Neighborhood-based uncertainty generation in social networks*. Journal of Combinatorial Optimization, 2014. 28(3): pp. 561-576.

[33] P. Yin, P. Luo, W. C. Lee and M. Wang, *Silence is also evidence: interpreting dwell time for recommendation from psychological perspective*, Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, 2013.

[34] Y. M. Li and C. W. Chen, *A synthetical approach for blog recommendation: Combining trust, social relation, and semantic analysis*, Expert Systems with Applications, 2009, 36(3): 6536-6547.