International Journal of Web Services Research
Volume 19 • Issue 1

Recommendations for Crowdsourcing Services Based on Mobile Scenarios and User Trajectory Awareness

Jie Su, Hunan International Economics University, China*
Jun Li, Hunan International Economics University, China

ABSTRACT

With the rapid development of the mobile internet and the rapid popularization of smart terminal devices, types and content of services are changing with each passing day. These bring serious mobile information overload problems for mobile users. How to provide better service recommendations for users is an urgent problem. A crowdsourcing service recommendation strategy for mobile scenarios and user trajectory awareness is proposed. First, the location coordinates in the historical log are clustered into regions by clustering algorithms, and then the user's trajectory patterns are mined in different mobile scenarios to extract mobile rules. Furthermore, the mobile rules are extracted, and the scenario to which each rule belongs is judged. When performing crowdsourcing service recommendation, the location trajectory and mobile scenario information are perceived in real time. They are used to predict the location area where the user will soon arrive. Thereby, the crowdsourcing service in the area is pushed to the user.

KEYWORDS

Mobile Context, Mobile Crowdsourcing, Service Recommendation, Task Recommendation, Trajectory Prediction

INTRODUCTION

With the rapid development of wireless communication technology and mobile smart terminals, based on location service, mobile networks anytime, anywhere can be obtained with its unique features such as mobility, practicability, and portability (Jensen, C. S., et al. 2001). These services and information content are widely used in many fields. The so-called location-based service refers to the cooperation of mobile terminals, and wireless or satellite communication networks are used to determine the actual geographic location of mobile users, location-related information services are provided to users such as map navigation, logistics tracking, traffic monitoring, mobile crowdsourcing task recommendation. However, mobile internet services and information delivery are greatly affected by contextual information and mobile social networks. How to find services that users are interested in from the vast ocean of mobile information and improve user personalized service experience has become an urgent problem for mobile recommendation systems.

Compared with traditional Internet users, the biggest feature of users in mobile communication networks is the random change of user location over time. There is the change of location, it is possible to recommend services based on different locations of mobile users. The core of mobile crowdsourcing service is crowdsourcing task recommendation, which aims to push the spatio-temporal tasks to a

DOI: 10.4018/IJWSR.299020
*Corresponding Author

set of workers (Tong, Y. X., et al. 2016; Kazemi, L., et al. 2013; Kazemi, L., & Shahabi, C. 2012), and workers complete the same task independently or in a cooperative manner (for example, taking pictures / shooting video or check-in to a specified location), while the constraints of time, location and other tasks are meet (Amador, S., et al. 2014; To, H., et al. 2015). In the mobile crowdsourcing application scenario, crowdsourcing task recommendation faces two important challenges:

The first challenge is the uncertainty of the mobile user's travel trajectory and its intent. Under the crowdsourcing mode, the task is performed by a non-specific group of people on the internet. The acceptance and execution of tasks follow the voluntary principle, which cannot be forced by users based on their own interests or intentions. When task recommendations are made, each potential user's trajectory changes, behavioral intentions, and the impact of various mobile scenarios on them must be fully considered, it tries to match the mobile task with the user's will as much as possible, the success rate of task recommendation and their satisfaction are improved with service recommendation.

The second challenge lies in the dynamic nature of the mobile crowdsourcing service scenario. For example, O2O(Online To Offline) services, such as taxis and takeaways, are constantly emerging from task publishers or potential performers, or they exit at any time, and their positions and trajectories are also dynamically changing. The real-time effectiveness of the space-time task recommendation algorithm and the ability to adapt to dynamic scenarios all raise higher requirements.

The current strategy for recommending tasks to users is mainly based on the current location of the mobile user (task assignment time), a spatio-temporal task is pushed for the user or a task execution route is planed for him, and it is less attention to the user's own trajectory and its location change trend. It may be because the task recommended to him deviates from its trajectory direction and behavioral intention, and he refuses to accept this task, which leads to a low success rate of spatio-temporal task recommendation.

In this paper, because a lot of meaningful information can be provided in the historical trajectory data of workers, a user's movement trajectory pattern, behavior habits, preferences are analyzed for certain places. Based on this analysis, the user's mobile location area is predicted, the tasks are pushied in the area after the prediction, it will increase the probability that the user will accept the task, while the extra travel costs, time and other costs are reduced.

The main contributions in this article are as follows:

- Firstly, discrete locations are clustered into regions, and then the user's regional movement trajectories are mined, and the movement pattern set is used to describe the trajectory changes of specific users in different mobile scenarios.
- Based on the user's movement pattern set, a movement rule is constructed, and a method is proposed to predict the location of the worker, and the crowdsourcing tasks are pushed in the area to the user.
- Experiments on real data sets verify the effectiveness and accuracy of the user location prediction algorithm' task recommendation strategy in different mobile scenarios.

RELATED WORK

With the development of internet mobile communication technology and the widespread use of smart terminals, many researchers have focused their attention on the analysis and mining of historical trajectory data, which has resulted in many research results related to spatial location (Liu, T., et al. 1998; Chen, C., et al. 2015; Nanopoulos, A., et al. 2003; Deng, D. X., et al. 2013). However, most of them focus on how to specifically analyze the historical trajectory of moving objects, and find meaningful information from it, while the research on location prediction technology is relatively small. Jeung and Liu et al. proposed a novel method (Jeung, H., et al., 2008), a predefined motion formula is combined to predict the user's next position. In the pre-defined motion program, the movement behavior of moving objects is captured by using complex mathematical formulas, and frequent movement patterns are extracted in Apriori algorithm from user trajectories. The used motion program can be a linear model or a non-linear model. This method has huge time overhead and huge amount of calculation. Morzy used an improved version of the Apriori algorithm to generate association rules (Morzy, M., 2006). In his later research, he also uses an improved Prefix Span algorithm to find frequent user movement patterns, and then uses the found frequent patterns to generate prediction rules (Morzy, M., 2007). Although all methods of Morzy consider geographic information represented by time information and latitude and longitude, they do not consider the semantic information implicit in geographic location.

The rapid popularization of mobile positioning devices such as mobile phones has promoted the development of spatio-temporal crowdsourcing technology, and various spatial location-based services are getting closer and closer to people's lives. Space-time crowdsourcing task allocation has become one of the key issues in the field of space-time crowdsourcing. Unlike traditional crowdsourcing methods, people can complete crowdsourcing tasks online (Tong, Y., et al. 2018). Space-time crowdsourcing requires users (ie crowdsourcing workers) to complete location-based space-time crowdsourcing tasks online (such as meal delivery in Meituan, posting what they have seen at a certain moment in WeChat Moments, etc.). How to allocate space-time crowdsourcing tasks to appropriate crowdsourcing workers is still the focus of research in the field of space-time crowdsourcing. Cheng et al. divide spatio-temporal crowdsourcing systems into two categories (Cheng, P., et al. 2017): workers' motivation and distribution mode. There are two types of motivation for workers: reward type and interest type. Reward type motivates workers to complete tasks through remuneration (such as Didi Taxi, Meituan Takeaway), and interest type uses workers' willingness and interest to actively complete tasks. The distribution model also has two modes: spatio-temporal crowdsourcing workers actively select tasks, and spatio-temporal crowdsourcing platform assigns tasks to workers.

In the previous research on the task allocation problem of spatio-temporal crowdsourcing: For the taxi platform, Tong et al. proposed the LinUOTD model, the number of taxi calls is predicted per unit time and unit area (Tong, Y., et al. 2017). Considering the diversity of space-time crowdsourcing tasks, Cheng et al. proposed three effective approximation methods namely greedy algorithm, sampling algorithm and divide-and-conquer algorithm (Cheng, P., Lian, X., et al. 2015). Deng et al. proposed two methods based on dynamic programming and branching. Based on dynamic programming and branch and bound strategies, Deng et al. proposed two precise algorithms (Deng, D. X., Shahabi, C., & Demiryurek, U.,2013), the number of spatiotemporal crowdsourcing workers is maximized to choose tasks autonomously. For the task assignment problem of multiple spatiotemporal crowdsourcing workers, Deng et al. proposed an algorithm based on the dichotomy framework (Deng, D. X., Shahabi, C., & Zhu, L H., 2015). For the optimal allocation of spatio-temporal crowdsourcing tasks, Tong et al. carried out a unified implementation and clarified the advantages and disadvantages of each algorithm (Tong, Y., She, J., Ding, B., et al. 2016). Li Yang et al. proposed that the tree decomposition algorithm is used to solve the spatio-temporal crowdsourcing task assignment problem of crowdsourcing workers with the latest time constraint (Li, Y., Jia, M. D., et al. 2018). Song et al. studied the dynamic and real-time spatio-temporal crowdsourcing assignment problem (Song, T., Tong, Y., Wang, L., et al. 2017). However, these researches on spatiotemporal crowdsourcing distribution algorithms are all aimed at rewarded spatiotemporal crowdsourcing workers. They are all subject to the constraints of spatiotemporal crowdsourcing workers with paid rewards and must accept spatiotemporal crowdsourcing tasks to find more efficient spatiotemporal tasks. Crowdsourcing task allocation method. Ignore the crowdsourcing task assignment of those interested spatiotemporal crowdsourcing workers who have no incentive constraints and choose to complete crowdsourcing tasks. In order to improve the accuracy of mobile user location prediction, a mobile user location prediction method is proposed based on parallel pattern mining and path matching (Xu, X. Z., Tan, S. H., et al. 2020). The traditional FP-GROWTH algorithm is parallelized, and the node load distribution method is optimized. Frequent user movement patterns are mined under the Spark platform. The index-based path similarity algorithm is improved, and a repulsion algorithm is proposed based on the shortest distance of the path, the applicability of missing trajectory data is improved (Chen, M., Li, W. Z., et al., 2020).

Mobility prediction is one of the most essential issues that need to be explored for mobility management in mobile computing systems. A new algorithm was proposed for predicting the next inter-cell movement of a mobile user in a Personal Communication Systems network (Yavaş, G., Katsaros, D., Ulusoy, Ö., et al. 2005). In the first phase of our three-phase algorithm, user mobility patterns are mined from the history of mobile user trajectories. In the second phase, mobility rules are extracted from these patterns, and in the last phase, mobility predictions are accomplished by using these rules. The performance of the proposed algorithm is evaluated through simulation as compared to two other prediction methods. In this paper, the location prediction method is improved, and contextual influence factors are added to the prediction of mobile pattern generation, such as taking into account the differences in user movement behavior on weekends / holidays and weekdays, and mobile rules are extracted according to the context dependent mobile patterns, the accuracy of prediction is improved.

PROBLEM DEFINITION

The basic idea of the method in this paper is to predict the location area that the user will reach by mining the trajectory pattern of the mobile user, and then the tasks in the area is recommended to the workers to increase the probability of the workers successfully accepting the tasks. In order to facilitate understanding, the relevant definitions of the method in this paper are introduced.

Definition 1 (region *r*): The region *r* represents the geographic location range, which is the aggregation of the physical coordinates of the mobile user and the spatio-temporal task. It is represented by a number in this paper. For example, the Wanda Plaza business district can be used as a region *r*.

Definition 2 (Worker Actual Path, WAP): The worker actual path *WAP* is defined as $Wap(w) = <(r_1, t_1), (r_2, t_2), ..., (r_n, t_n)>$, where (r_i, t_i) indicates that the worker *w* reached the area r_i at time t_i, where r_i is the area number. A route indicates the location of the area where worker *w* has visited in a chronological order during the day.

Definition 3 (Worker Mobile Pattern, WMP): The mobile mode *WMP* is also a worker's trajectory. It appears more frequently in the trajectory of the worker, which means that the worker often goes to the place, which can well describe the daily movement trajectory of the worker. $Wmp(w) = (<(r_1, t_1), (r_2, t_2), ..., (r_n, t_n)>, supp)$, where $<(r_1, t_1), (r_2, t_2), ..., (r_n, t_n)>$ is same as the definition of route 2 above, *supp* indicates how frequently the route appears based on the historical trajectory of worker *w*, which is called support, and $supp \geq 0$. For the calculation of support refer to the Apriori algorithm (Cao, L., Sun, Y., et al. 2020; Zhang, L., Dong, W. F., et al. 2020).

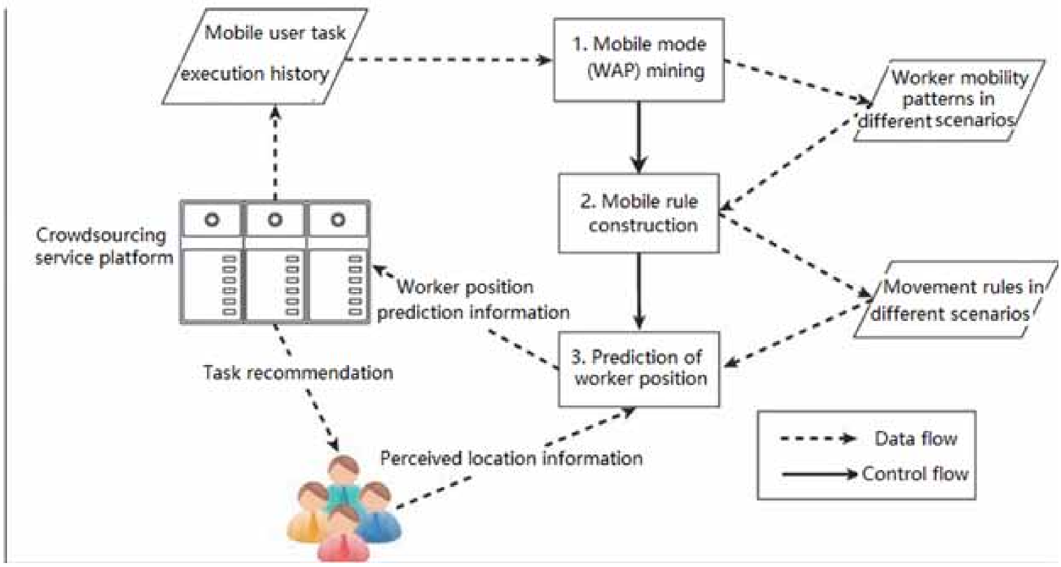
Definition 4 (Worker Mobile Rule, WMR): A mobile rule *WMR* describes the transfer relationship between workers in various regions, it is expressed as $Wmr = <r_1, r_2, ..., r_{k-1}> \rightarrow <r_k>$, where $< r1, r_2, ..., r_{k-1}>$ are regular heads, which indicate the current trajectory of the worker, and $<r_k>$ are regular tails, which indicate the area where the worker has the highest probability to go. The movement rules are obtained according to the movement mode $(<(r_1, t_1), (r_2, t_2), ..., (r_{k-1}, t_{k-1}), (r_k, t_k)>, supp)$. In a worker movement mode, he went to *k* regions according to time, the first *1,2, ..., k-1* regions can be used as the regular head $(k > 1)$, then the corresponding remaining *k-1, k-2, ..., 1* region is used as the predicted region. An example of a moving rule set is shown below.

$<r_1>\rightarrow<r_2,r_3,\ldots,r_k>$
$<r_1,r_2>\rightarrow<r_3,r_4,\ldots,r_k>$
$<r_1,r_2,\ldots,r_{k-1}>\rightarrow<r_k>$

PREDICTION OF WORKER POSITION BASED ON MOBILE PATTERN MINING

The mobile worker's location and area prediction and task recommendation process includes four stages: first, discrete location points are clustered to obtain the area, and the historical trajectory of the worker's historical area is mined to form a worker's movement pattern; the worker movement rules are constructed based on the obtained worker movement pattern. Based on the real-time perception of the worker's position information and movement rules, the location area where the worker will reach next is predicted; finally, a spatio-temporal task is recommended for the worker based on the prediction. The location prediction and task recommendation process is shown in Figure 1.

Figure 1. Location prediction and task recommendation process for mobile workers



Generate Area

Assuming that the positions of all tasks in this paper can be included in these regions, the most popular clustering algorithm K-means algorithm is used to aggregate the position points into regions to achieve point-to-region transformation. The steps of the K-means algorithm are as follows:

Step 1: K initial centroids are randomly selected;

Step 2: If the termination condition of the clustering algorithm is not met, continue to step 3, otherwise go to step 5;

Step 3: Calculate the Euclidean distance from each non-centroid point p to k centroids, and assign p to the nearest centroid;

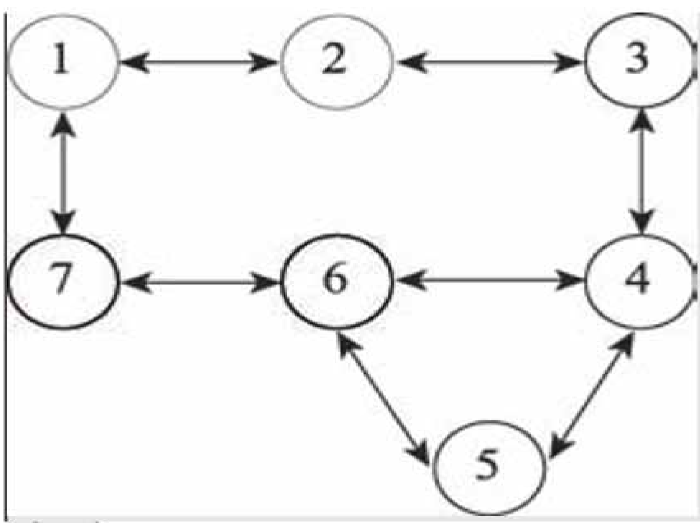
Step 4: Based on the k centroids in the previous step and their corresponding set of non-centroid points, recalculate the new centroid points, and then go to step 2;

Step 5: Output the clustering results. The algorithm can be executed multiple times, compare different clustering results, and choose a better one.

After clustering the position points by the K-means algorithm, the regions are obtained, which are further formalized as a directed graph G. Figure 2 is an example of an actual area map, and Figure 3 is a directed map of the area network. The two-way arrows represent direct reach between the two areas.

Figure 2. Actual area map

Figure 3. Digraph of regional network



Mining Worker Movement Pattern

This section details how to mine worker movement patterns, here are the detailed steps.

Knowing that the worker has multiple actual routes, firstly, a candidate pattern set C_1 with a length of 1 can be obtained, that is, a subpath with a length of 1, and their support degree is calculated. If it is greater than the threshold 1.33 in this paper, it is added to the mobile pattern set with a length of 1, it is represented by L_1; in the area of L_1, observe which area can be reached directly from the current area through the regional network directed graph, then add its area number to the set to form a candidate pattern set C_2 of length 2; then calculate the support degree, if it is greater than the threshold, it is added to the mobile pattern set L_2 of length 2. According to this rule, until the candidate pattern set is not found, the generation of the mobile pattern set is completed. The description of the worker movement mode *WMPMining()* mining algorithm is shown in Algorithm 1.

In terms of mobile scenarios, dividing workdays and weekends will generate their mobile modes L_w and L_p, respectively.

Algorithm 1: WMP mining of worker movement patterns *WMPMining()*
Input: worker's actual route (WAP) *D* in the database, minimum support $supp_{min}$, network area graph *G*.
Output: Worker movement mode (WMP) L_w, L_p.

```
C_1¬the patterns which have a length of one
k=1
L_w=Æ //Initially workday patterns is empty
Lp=Æ //Initially weekend patterns is empty
While C_k¹Æ{
   For WAP aÎD {
         S ={s|sÎC_k and s is a subsequence of a} // S is the set of candidate length-k patterns which are also subsequence
         of WAP a
   for each sÎS {
               s.count = s.count + s.suppInc //increment the support of c and s
                     }
         }
               L_k ={s|sÎC_k, s.count³suppmin } //choose the candidates which have enough support
               IF(t_K==workday){
                     L_w=L_wÈL_k //add these length-k large patterns to the set of all large patterns
                     C_k+1¬CandidateGeneration(L_k,G), ∀cÎC_k+1 c.count = 0 //Generate length-(k+1) candidate patterns
                     k=k+1
         }
               Else{
               L_p=L_pÈL_k //add these length- k large patterns to the set of all large patterns
                     C_k+1¬CandidateGenaration(L_k,G), ∀cÎ C_k+1 c.count = 0 //Generate length-(k+1) candidate patterns
                     k=k+1
               }
   }
return L_w, L_p
```

To illustrate how *CandidateGeneration()* works, suppose there is a user trajectory of length k, $C = <(r_1, t_1), (r_2, t_2), ..., (r_k, t_k)>$ is as the input of the algorithm. In the directed graph of the network, all the regional vertices that can be reached from the vertex r_k are first added to the set $N^+(r_k)$. The regional vertices in this set represent the places that the workers can reach from r_k. Then, a certain vertex v is selected from $N^+(r_k)$, and it is added to the candidate sequence $C' = <(r_1, t_1), (r_2, t_2), ..., (r_k, t_k), (v, t)>$, C' is added to the candidate sequence of length $k + 1$. Table 1 gives an example of the worker's historical trajectory based on the directed graph of the regional network in Figure 3.

Table 1. Workers' actual path

WAP ID	WAP
1	$<(2,t_1),(1,t_2),(7,t_3),(6,t_4),(4,t_5),(8,t_6)>$
2	$<(7,t_1),(6,t_2),(4,t_3),(5,t_4)>$
3	$<(1,t_1),(3,t_2),(4,t_3),(5,t_4)>$
4	$<(7,t_1),(6,t_2),(5,t_3)>$

Suppose the pre-set threshold $supp_{min} = 1.33$. According to the worker's actual route in Table 1, the user's mobile pattern set L is mined by the *WMPMining()* algorithm, as shown in Table 2. For the convenience of reading, the time is omitted in the table display, but the time is actually taken into account.

Table 2. Worker mobile patterns set L

Cand	supp	Cand	supp	Cand	supp
<1>	2	<7>	3	<7,6>	3
<4>	3	<4,5>	3	<6,4,5>	2
<5>	4	<6,4>	2	<7,6,4>	2
<6>	3	<6,5>	2		

Generate Move Rule

On the basis of mining workers' movement patterns, extraction of movement rules is made. From the definition of movement rules in **Definition 4**, it is known that the workers' movement patterns WMP, $L = <(r_1, t_1), (r_2, t_2), ..., (r_k, t_k)>, k> 1$, to see its moving rules in **Definition 4**.

For a rule $R: <r_1, r_2,..., r_{i-1}> \ldots <r_i, r_{i+1},..., r_k>$, a confidence level is defined to represent the credibility of the rule, that is, the accuracy of the prediction. Confidence is defined in using the following formula (1):

$$confidence = \frac{< r_1, r_2, \cdots, r_k > \cdot \sup p}{< r_1, r_2, \cdots, r_{k-1} > \cdot \sup p} \times 100\% \quad (1)$$

WMP is obtained from the workers' movement pattern mining algorithm, all possible movement rules can be generated, and their confidence can be calculated. If the confidence level of a rule is higher than a preset confidence threshold ($coff_{min}$), they are selected for the next region prediction phase.

Because the mobile mode is extracted based on different scenarios (this article only considers the working day / holiday correlation), each mobile rule also needs to add different scenario tags to indicate the mobile rules in a specific scenario.

For example, according to the movement pattern set shown in Table 2, movement rules are generated and the confidence of each rule is calculated. If the confidence threshold ($coff_{min}$) is set to 50%, the mobile rule set is shown in Table 3 (WD stands for working day, PD stands for non-working day).

Table 3. Mobile rule set

Rule	context	Confidence/%
<4>→<5>	PD	100.0
<6> → <4>	WD	66.7
<6>→<5>	PD/WD	66.7
<7>→<6>	WD	100.0
<6>→<4,5>	PD/WD	66.7
<6,4>→<5>	WD	100.0
<7>→<6,4>	PD/WD	66.7
<7,6>→<4>	WD	66.7

Worker Area Forecast

Location area prediction is the last stage. The pseudo code of the algorithm is described as follows:

Algorithm 2: location area prediction *MobilityPrediction()*
Input: the user's current trajectory $P = <(r_1, t_1), (r_2, t_2), ..., (r_{i-1}, t_{i-1})>$, the movement rule set *R*, and the predicted maximum number of positions *m*.
Output: Prediction Region Set *PRegions*.
1. *PRegions* = ∅ //Initially the set of predicted cells is empty
2. *k=1*
3. sort(*r*) //Descending order according to the length of the rule's head
4. For rule $r{:}<a_1,a_2,…,a_j>\rightarrow<a_{j+1},…,a_i>\in R$ //check all the rules in R find the set of matching rules
5. {
6. *IFP.time* ∈*R.context and* $<a_1,a_2,…,a_j>$ is contained by $P=<r_1,r_2,…,r_{i-1}>$
7. {
8. MatchingRules←MatchingRulesÈ*r* //Add the rule into the set of matching rules
9. *TupleArray[k]=*$(a_{j+1}$*,r.confidence)* //Add the $(a_{j+1}$*, r. confidence)* tuple to the Tuples array
10. *k = k+1*

```
}
TupleArray ← sort (TupleArray) Now descending order the Tuples
array according to the second element of tuples, which is the
confidence of the corresponding rule }
index = 0
While(index<m&&index<TupleArray.length){
PRegions←PCells∪TupleArray[index]
index = index +1
}
return PRegions
```

The prediction process of the above algorithm can be summarized as follows: Assume that the worker's current trajectory is $P = <(r_1, t_1), (r_2, t_2), ..., (r_{i-1}, t_{i-1})>$. First, find all rules that fit the current path, which is summarized into the matching rule set. The matching rule has the following characteristics: the head of the rule is included in the current movement trajectory, which is equivalent to its sub-path. When matching, the first matching head is exactly the same as the trajectory, and then the principle of regular head length is gradually reduced to find. The mobile rule set is scaned once to find all the rules that meet the characteristics of the matching rule. The first area is numbered at the end of each rule. The confidence of this rule is composed of a primitive ancestor *(r, confidence)*, and the descending order of the confidence is sort. Another parameter *m* is defined in the algorithm as the number of predictable regions. Selecting the first *m* sorted tuples from the array means that the first *m* matching rules are used to predict the next position of the worker.

Because the scenario tag (ie, whether it is a working day) is added when generating the mobile rule, first determine whether the current time is consistent with the content of the current tag when scanning. If it does not match the rule tag, skip directly to scan the next rule, which will greatly improve the matching effectiveness.

MOBILE CROWDSOURCING TASK RECOMMENDATION

In the mobile crowdsourcing application scenario, because the spatial tasks and the position and number of workers are dynamically changed and updated in real time, the target of the task recommendation is considered to achieve a local optimum that maximizes the number of task allocations over a period of time. This article is based on greedy strategy for task recommendation (Jeung, H., Liu, Q., Shen, H. T., et al. 2008). Given a set of workers $W_i = \{w_1, w_2, \ldots\}$ and a set of tasks $T_i = \{t_1, t_2, \ldots\}$ in area *D*, the goal is to recommend the tasks in T_i to the workers in W_i for a period of time. This article assumes that the quality of all workers is consistent.

Based on the proposed strategy (Jeung, H., Liu, Q., Shen, H. T., et al. 2008), the maximum flow problem of graphs is used to solve the task recommendation problem. Given a group of workers $W_i = \{w_1, w_2, \ldots\}$ and tasks $T_i = \{t_1, t_2, \ldots\}$ in multiple regions R_i, $G_i = (V, E)$ represents the vertices and edges of the network graph, which contains $| W_i | + | R_i | + | T_i | +2$ vertices. Each worker w_j corresponds to a vertex v_j, each region r_j is mapped to a vertex $r_{|wi|+j}$, and each task t_j is mapped to a vertex $VWI | + | RI | + J$. . Create a source point *src* and an end point *dest* at the same time, G_i contains $| W_i | + | R_i | + | T_i | + m$ edges, where $| W_i |$ edges are the source point *src* and each vertex v_j, (src, v_j) is used to indicate, the edge weight is the maximum number of tasks acceptable to each worker; there are $| T_i |$ edges that connect task T_i and end point *dest*, and the weight is set to 1, indicating that this task is recommended to a worker.

Figure 4 is a regional task distribution map, which is transformed into the process network diagram shown in Figure 5. The arrow between the worker and the area in Figure 4 indicates that the area is predicted to be reached by the worker. w_1, w_2, w_3, and w_4 in Figure 4 correspond to v_1, v_2, v_3, and v_4 in Figure 5, and areas r_1, r_2, and r_3 correspond to v_5, v_6, and v_7 in Figure 5. The tasks in the area correspond to v_8 to v_{16}. Based on the prediction of the workers' arrival location area by the algorithm described in the previous chapter, the recommended set of candidate workers for each spatial task can be determined, and then tasks in the area are recommended to the workers in the set of candidate workers. By solving the graph maximum flow problem, such as the Ford-Fulkerson algorithm (Tarjan, R. E., 1987), the above task recommendation problem can be solved.

Figure 4. Distribution map of workers, region and task

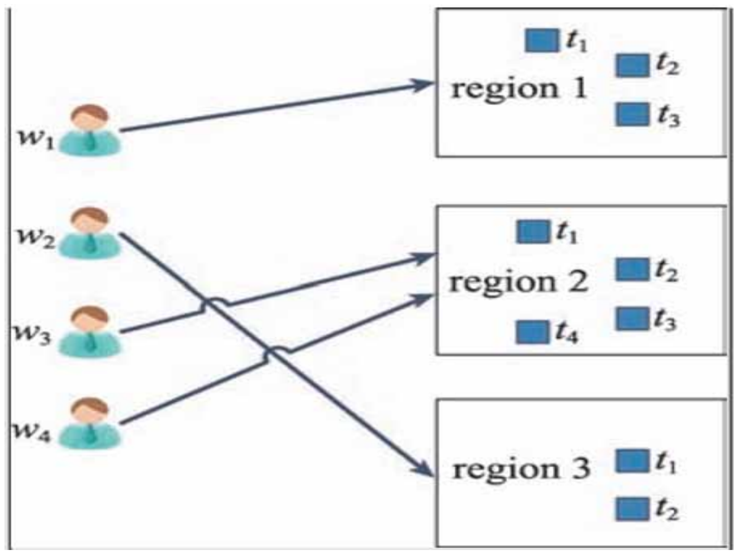
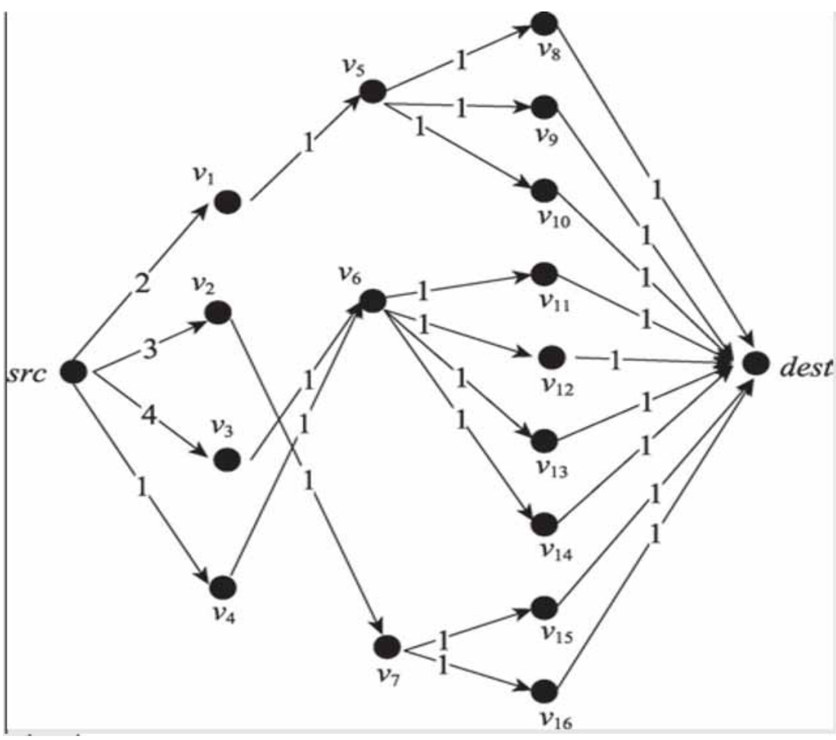


Figure 5. Process network diagram



EXPERIMENT AND DISCUSSION

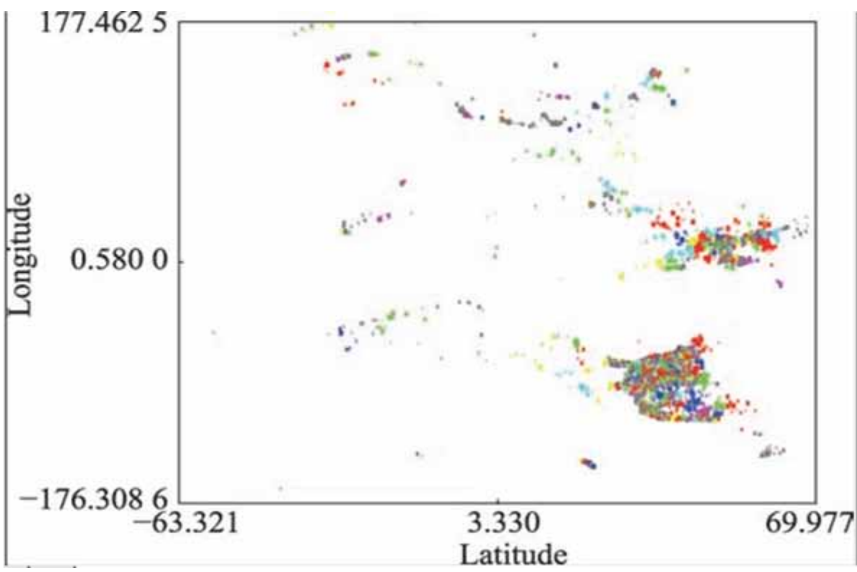
Data Collation

In order to test the proposed method in a real environment, the data set of the location social network *Gowalla* is used in this paper, including the user's time and latitude and longitude of its location, and the location ID. The check-in set included more than 6.44 million pieces of data which were collected from 2009 to October 2010, which were cleaned and processed, and finally a data set with 1 million pieces of data was extracted, which contains more than 4,000 users and 45,000 different locations. The location and user of the data set are used to represent the crowdsourcing task in space and the location of the worker. As long as the worker signs in to the designated location, it is regarded as accepting the crowdsourcing task and completing it. Although the data set is not directly crowdsourced from space, it provides different, real-world distributions of workers and task positions. Since the algorithm studied in this paper depends on position, their relative performance and some reasonable conclusions can be obtained by using this data set.

The data set itself is the check-in data of some users, which is relatively sparse. In this paper, the extracted data set is processed first, and all the damaged and blank data that does not help the experiment are cleared. After the data is cleaned, it is clustered using the *K*-means algorithm, and the discrete data is aggregated into multiple regions, which are regarded as the distribution points of the crowdsourcing task.

Figure 6 is the result of clustering the dataset. The number of categories is 2,000, and each category is marked with a different color. Each color represents a class, that is a region.

Figure 6. Clustering region graph



According to the time sequence, the trajectory path of each worker is divided into a training set and a test set, where the training set is used to obtain the worker movement rules introduced in ***Generate move rule***, and the test set is used to test the area predicted by the movement rules and the workers' actual going. Whether the area is consistent. If they agree, it indicates that the predicted area where the worker is about to go is the same as the distribution area of the crowdsourcing task, and the tasks in the area can be pushed to the worker, which means that the task recommendation is successful, otherwise the recommendation fails.

EXPERIMENT AND DISCUSSION

success represents the success rate of task recommendation. The calculation is in follow formula (2):

$$success = \frac{match}{all} \tag{2}$$

Where, *match* indicates that the task recommended amount of success, *all* means all of the number of tasks.

The following is a comparison of the method (WMP) in this paper with the prediction method (UMP) proposed by researchers such as Yavaş (Yavaş, G., Katsaros, D., Ulusoy, Ö., et al. 2005), and the comparison is based on the accuracy of the prediction region number of the test set, as shown in Figure 7 and Figure 8.

Figure 7. Comparison of accuracy of WMP and UMP on workdays

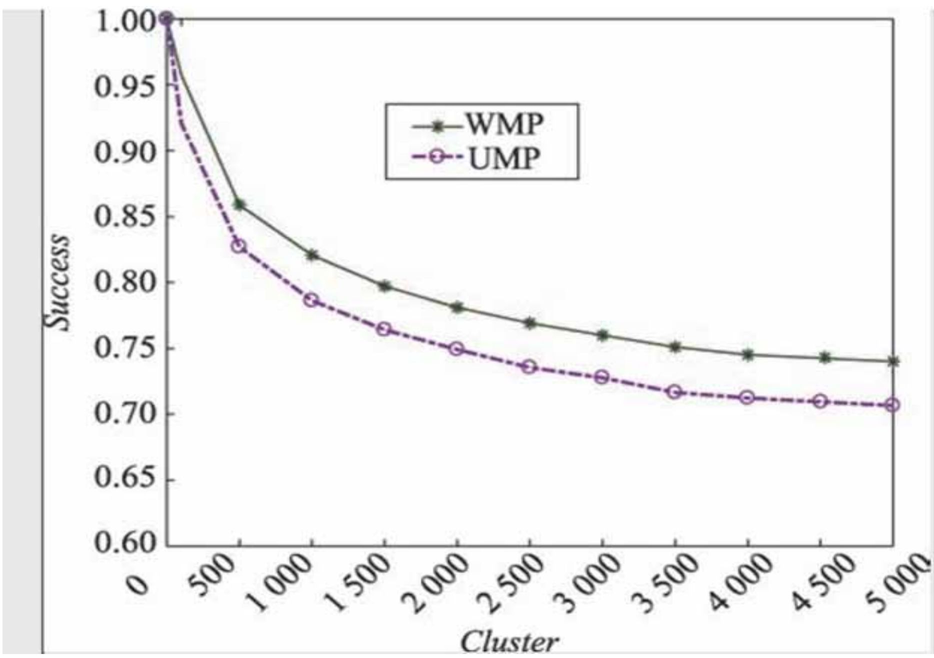
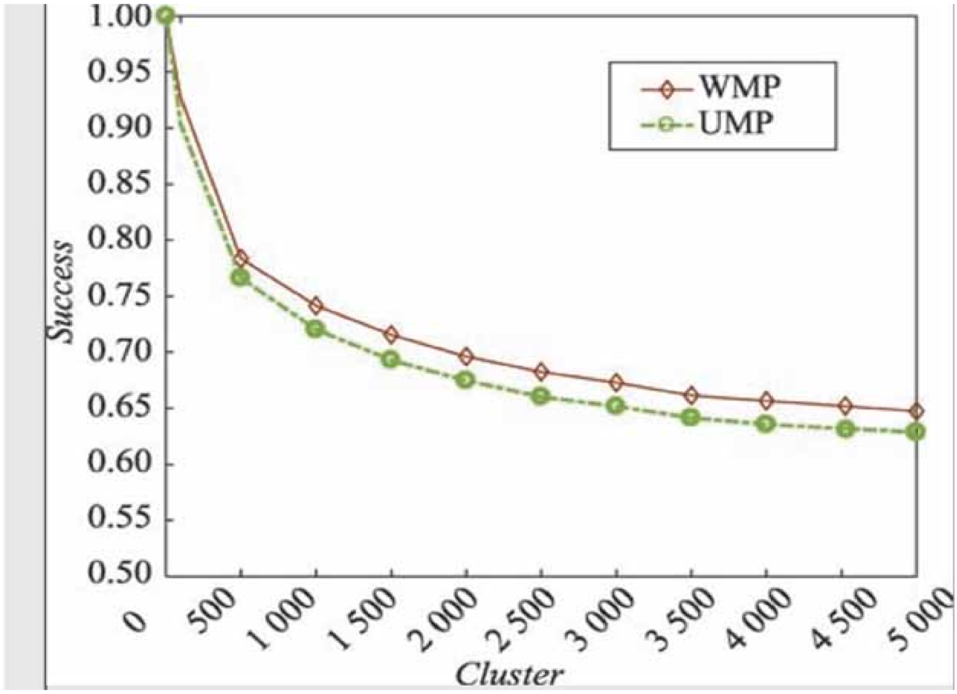


Figure 8. Comparison of accuracy of WMPand UMP on weekends



In Figure 7 and Figure 8, as the number of clustering categories increases, the accuracy of both methods decreases. If the number of categories is small, the range of the actual area will be large, and the trajectory area of the worker will be small. The worker will basically stay within a fixed area of one or two areas, and the accuracy will increase. As the position gets closer and closer, the trajectory of the worker starts to increase, and the accuracy will decrease slightly. When the number of categories is 100, the area is large, and workers stay in one area a day, and the prediction results will be accurate.

It can be seen from Figures 7 and 8 that the method in this paper performs better than the method proposed by Yavaş et al on both working and non-working days. This paper considers the time series of workers' historical trajectories. Studies show that the probability of users going to the first 100 places is 0.5 greater than the probability of hundreds of thousands of places behind (Wang, W. Q., Yin, H. Z., et al. 2016; Wang, H. T., et al. 2020). This shows that there is some potential link between regional locations, and not only the last region is considered for prediction. After the workday is divided, not only the success rate increases, but also the time complexity of the algorithm will be lower, because the rule label can be used to directly determine whether it is a workday, the time is reduced for the algorithm to scan the rules, and then the area is efficiently predicted, and the task is pushed.

CONCLUSION AND OUTLOOK

Crowdsourcing is a new form of production organization brought by the Internet. Wired magazine reporter Jeff Howe invented a term in 2006 to describe a new business model in which companies use the Internet to distribute work, discover ideas, or solve technical problems (Howe, J.,2006). Through Internet control, these organizations can take advantage of the creativity and abilities of the army of volunteer employees-these volunteer employees have the skills to complete tasks, are willing to work in their spare time, are satisfied with charging small payments for their services, or are temporarily unpaid, just satisfied the prospect of getting more rewards in the future. Especially for the software and service industries, this provides a new way to organize the workforce.

For mobile crowdsourcing service recommendation, a crowdsourcing service recommendation strategy is proposed in this paper based on mobile scenarios and user trajectory awareness. In this method, when tasks are distributed on the crowdsourcing platform, the real-time perceived trajectory information and movement rules predict the location area where the user is about to arrive, so as to push the space-time tasks in the area to the user. The task assignment method in this paper avoids additional costs such as the time, process, and cost of performing tasks. Workers are more willing to accept tasks, which increases the probability of task completion and improves the satisfaction of user experience recommendation services.

Crowdsourcing refers to the practice of a company or organization outsourcing tasks previously performed by employees to a non-specific (and usually large) mass network in a free and voluntary manner. The tasks of crowdsourcing are usually undertaken by individuals, but if it involves tasks that require multi-person collaboration, it may also appear in the form of individual production relying on open source.

Crowdsourcing is not only an upgrade to outsourcing, but also a kind of subversion. Its subversive significance is shown in:

- Crowdsourcing is not outsourcing. "People think this is outsourcing, but it must be a misunderstanding," commented Larry Huston, vice president of technology innovation at Procter & Gamble. "Outsourcing means that we hire people to provide services. The relationship is no different. But now our approach is to attract the participation of talents from the outside so that they can participate in this broad innovation and cooperation process. These are two completely different concepts." Outsourcing is generally done by a designated person or organization (usually an expert in a certain field) that has been acquainted with before, while crowdsourcing is to throw a certain task or problem to the unknown public-it can be any one who might contribute answers or solutions.
- The core of crowdsourcing actually contains the idea of creating value with users.
- In crowdsourcing, individual innovation becomes the mainstream. Outsourcing emphasizes a high degree of specialization, while crowdsourcing is the opposite. Cross-professional innovation often contains huge potential. There are numerous business cases that have been successful with the active participation of individual users.
- The most important thing is that with the rise of grassroots power, there are rules to turn grassroots power into commercial power.

Fast establishing itself as a legitimate business tool, the perceived benefits of crowdsourcing are obvious – problems can be explored quickly and at comparatively low cost, and businesses can tap a range of talent that may not exist within the confines of a particular organization. It's also useful for marketing purposes because the crowd may include potential customers – asking people for their response to a problem may also be a way to glean information about what they want from a product or service.

With the crowdsourcing logistics becomes more and more mature, There are chances that it will become the trend of high-profile. E-commerce and the rapid development of O2O (Online To Offline) lead to the increase demand of retail logistics services, but so far, the last mile of the delivery of many retailers and Electricity business platform is still quite bothering. Crowdsourcing logistics model may be able to help solve the problem of delivery in remote areas. However, given the potential risks and uncertainties involved, companies should weigh the pros and cons of testing this new model.

FUNDING AGENCY

Publisher has waived the Open Access publishing fee.

ACKNOWLEDGMENT

This work is sponsored by the Scientific Research Project (No.20A282, No:18B528, NO. 19B329) of Hunan Provincial Education Department, China.

REFERENCES

Amador, S., Okamoto, S., & Zivan, R. (2014). Dynamic multi-agent task allocation with spatial and temporal constraints. *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems*, 1495-1496.

Cao, L., & Sun, Y. (2020). Three-Dimensional Warehouse System Based on Apriori Algorithm. *Computer Systems & Applications*, *29*(9), 115–120. doi:10.15888/ j.cnki.csa.007644

Chen, C., Cheng, S. F., & Lau, H. C. (2015). Towards city-scale mobile crowdsourcing: task recommendations under trajectory uncertainties. *Proceedings of the International Conference on Artificial Intelligence*, 1113-1119.

Chen, M., Li, W. Z., Qian, L., Lu, S.-L., & Chen, D.-X. (2020). Next POI Recommendation Based on Location Interest Mining with Recurrent Neural Networks. *Journal of Computer Science and Technology*, *35*(3), 603–616. doi:10.1007/s11390-020-9107-3

Cheng, P., Lian, X., Chen, L., & Shahabi, C. (2017). Prediction-based task assignment in spatial crowdsourcing. *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, 997-1008. DOI: doi:10.1109/ ICDE.2017.146

Cheng, P., Lian, X., Chen, Z., Fu, R., Chen, L., Han, J., & Zhao, J. (2015). Reliable diversity-based spatial crowdsourcing by moving workers. *Proceedings of the VLDB Endowment International Conference on Very Large Data Bases*, *8*(10), 1022–1033. doi:10.14778/2794367.2794372

Deng, D. X., Shahabi, C., & Demiryurek, U. (2013). Maximizing the number of worker's self-selected tasks in spatial crowdsourcing. *Proceedings of the 21st SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 314-323. DOI: doi:10.1145/2525314.2525370

Deng, D. X., Shahabi, C., & Zhu, L. H. (2015). Task matching and scheduling for multiple workers in spatial crowdsourcing. *Sigspatial International Conference on Advances in Geographic Information Systems*, 21, 1-10. doi:10.1145/2820783.2820831

Howe, J. (2006). Look Who's Crowdsourcing. *Revista Wired*.

Jensen, C. S., Friis-Christensen, A., & Pedersen, T. B. (2001). Locationbased services: a database perspective. *Proceedings of the 8th Scandinavian Research Conference on Geographical Information Science*, 59-68.

Jeung, H., Liu, Q., & Shen, H. T. (2008). A hybrid prediction model for moving objects. *Proceedings of the 24th International Conference on Data Engineering*, 70-79. doi:10.1109/ICDE.2008.4497415

Kazemi, L., & Shahabi, C. (2012). GeoCrowd: enabling query answering with spatial crowdsourcing. *Proceedings of the International Conference on Advances in Geographic Information Systems*, 189-198. doi:10.1145/2424321.2424346

Kazemi, L., Shahabi, C., & Chen, L. (2013). GeoTruCrowd: trustworthy query answering with spatial crowdsourcing. *Proceedings of the 21st SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 304-313. doi:10.1145/2525314.2525346

Li, Y., & Jia, M. D. (2018). Optimal Task Assignment Algorithm Based on Tree-Decouple in Spatial Crowdsourcing. *Journal of Software*, (3), 824–838. doi:10.13328/j.cnki.jos. 005453

Liu, T., Bahl, P., & Chlamtac, I. (1998). Mobility modeling, location tracking, and trajectory prediction in wireless ATM networks. *IEEE Journal on Selected Areas in Communications*, *16*(6), 922–936. doi:10.1109/49.709453

Morzy, M. (2006). Prediction of moving object location based on frequent trajectories. *LNCS 4263*: *Proceedings of the 21st International Symposium on Computer and Information Sciences*, 583-592.

Morzy, M. (2007). Mining frequent trajectories of moving objects for location prediction. *LNCS 4571*: *Proceedings of the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition*, 667-680. doi:10.1007/978-3-540-73499-4_50

Nanopoulos, A., Katsaros, D., & Manolopoulos, Y. (2003). A data mining algorithm for generalized Web prefetching. *IEEE Transactions on Knowledge and Data Engineering*, *15*(5), 1155–1169. doi:10.1109/ TKDE.2003.1232270

Song, T., Tong, Y., & Wang, L. (2017). Online task assignment for three types of objects under spatial crowdsourcing environment. *Journal of Software*, *28*(3), 611–630.

Tarjan, R. E. (1987). Algorithmic design. *Communications of the ACM*, *30*(3), 204–212. doi:10.1145/214748.214752

To, H., Shahabi, C., & Kazemi, L. (2015). A server-assigned spatial crowdsourcing framework. *ACM Transactions on Spatial Algorithms and Systems*, *1*(1), 1–28. doi:10.1145/2729713

Tong, Y., Chen, L., & Zhou, Z. (2018). Slade: a smart large-scale task decomposer in crowdsourcing. *IEEE Transactions on Knowledge and Data Engineering*, (99), 1588-1601. doi: .2018.279796210.1109/TKDE

Tong, Y., Chen, Y., & Zhou, Z. (2017). The simpler the better: a unified approach to predicting original taxi demands on large-scale online platforms. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1653-1662. doi:10.1145/3097983.3098018

Tong, Y., She, J., Ding, B., Chen, L., Wo, T., & Xu, K. (2016). Online minimum matching in real-time spatial data: Experiments and analysis. *Proceedings of the VLDB Endowment International Conference on Very Large Data Bases*, *9*(12), 1053–1064. doi:10.14778/2994509.2994523

Tong, Y. X., She, J. Y., & Ding, B. L. (2016). Online mobile microtask allocation in spatial crowdsourcing. *Proceedings of the 32nd IEEE International Conference on Data Engineering*, 49-60.

Wang, H. T., Chen, Y., Wang, J., Peng, H., & Liu, H. (2020). Mutual Inference Model for User Roles and Urban Functional Zones. *International Journal of Software Engineering and Knowledge Engineering*, *30*(3), 1–19. doi:10.1142/S0218194020400021

Wang, W. Q., & Yin, H. Z. (2016). SPORE: a sequential personalized spatial item recommender system. *Proceedings of the IEEE International Conference on Data Engineering*, 954-965. doi:10.1109/ICDE.2016.7498304

Xu, X. Z., & Tan, S. H. (2020). User Location Prediction Based on Parallel Pattern Mining and Path Matching. *Journal of Northeastern University*, *41*(6), 767–770. doi:10.12068/j.issn.1005-3026.2020.06.002

Yavaş, G., Katsaros, D., Ulusoy, Ö., & Manolopoulos, Y. (2005). A data mining approach for location prediction in mobile environments. *Data & Knowledge Engineering*, *54*(2), 121–146. doi:10.1016/j.datak.2004.09.004

Zhang, L., & Dong, W. F. (2020). Apriori algorithmCustomer Demand Data Analysis Based on Improved Apriori Algorithm. *Machinery Design & Manufacture*, (5), 185–188.

Jie Su (b. 1981) received the B.S degree in Tourism and Hospitality from Hunan Normal University in 2003, and received the M.S. degree in Ideological and Political Education from Changsha University of Science and Technology. Now, she is a researcher at Business School, Hunan University of International Economics, China. Her research interests include regional economics and direction of tourism information management.

Jun Li (b. 1980) received the B.S degree in Computer Science and Technology from Hunan Normal University in 2003, and received the M.S. degree in Computer Science and Technology from Hunan University. Now, he is a researcher at College of Computer, Hunan University of International Economics, China. Her research interests include image processing and information security.