

## **Consensus-Based Group Task Assignment with Social Impact in Spatial Crowdsourcing**

Li, Xiang; Zhao, Yan; Zhou, Xiaofang; Zheng, Kai

*Published in:*  
Data Science and Engineering

*DOI (link to publication from Publisher):*  
[10.1007/s41019-020-00142-0](https://doi.org/10.1007/s41019-020-00142-0)

*Creative Commons License*  
CC BY 4.0

*Publication date:*  
2020

*Document Version*  
Publisher's PDF, also known as Version of record

[Link to publication from Aalborg University](#)

*Citation for published version (APA):*  
Li, X., Zhao, Y., Zhou, X., & Zheng, K. (2020). Consensus-Based Group Task Assignment with Social Impact in Spatial Crowdsourcing. *Data Science and Engineering*, 5(4), 375-390. <https://doi.org/10.1007/s41019-020-00142-0>

### **General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

### **Take down policy**

If you believe that this document breaches copyright please contact us at [vbn@aub.aau.dk](mailto:vbn@aub.aau.dk) providing details, and we will remove access to the work immediately and investigate your claim.





# Consensus-Based Group Task Assignment with Social Impact in Spatial Crowdsourcing

Xiang Li<sup>1</sup> · Yan Zhao<sup>2</sup> · Xiaofang Zhou<sup>4</sup> · Kai Zheng<sup>3</sup>

Received: 18 May 2020 / Revised: 5 August 2020 / Accepted: 31 August 2020 / Published online: 15 September 2020  
© The Author(s) 2020

## Abstract

With the pervasiveness of GPS-enabled smart devices and increased wireless communication technologies, spatial crowdsourcing (SC) has drawn increasing attention in assigning location-sensitive tasks to moving workers. In real-world scenarios, for the complex tasks, SC is more likely to assign each task to more than one worker, called group task assignment (GTA), for the reason that an individual worker cannot complete the task well by herself. It is a challenging issue to assign worker groups the tasks that they are interested in and willing to perform. In this paper, we propose a novel framework for group task assignment based on worker groups' preferences, which includes two components: social impact-based preference modeling (SIPM) and preference-aware group task assignment (PGTA). SIPM employs a bipartite graph embedding model and the attention mechanism to learn the social impact-based preferences of different worker groups on different task categories. PGTA utilizes an optimal task assignment algorithm based on the tree decomposition technique to maximize the overall task assignments, in which we give higher priorities to the worker groups showing more interests in the tasks. We further optimize the original framework by proposing strategies to improve the effectiveness of group task assignment, wherein a deep learning method and the group consensus are taken into consideration. Extensive empirical studies verify that the proposed techniques and optimization strategies can settle the problem nicely.

**Keywords** Spatial crowdsourcing · Group task assignment · Social impact-based preference · Group consensus

## 1 Introduction

With the ubiquitous deployment of wireless networks and mobile devices (e.g., smart phones), spatial crowdsourcing (SC), an emerging paradigm utilizing the distributed mobile devices to monitor diverse phenomena about human activities, has attracted much attention from both academic and industry communities. The main idea of spatial crowdsourcing is recruiting a set of available workers to perform the location-specific tasks by physically traveling to these locations, called *task assignment*.

Most existing SC researches focus on single task assignment [20, 22], which assumes that tasks are simple and each task can only be assigned to a single worker. For example, Tong et al. [23] design several efficient greedy algorithms to solve the proposed global online micro-task allocation (GOMA) problem in spatial crowdsourcing. [12] considers task assignment and scheduling at the same time, in which an approximate approach is developed that iteratively improves the assignment and scheduling to achieve more completed tasks. However, in real-world scenarios, an

---

✉ Yan Zhao  
yanzhao2019612@gmail.com

Xiang Li  
xli96@stu.suda.edu.cn

Xiaofang Zhou  
zxf@itee.uq.edu.au

Kai Zheng  
zhengkai@uestc.edu.cn

<sup>1</sup> School of Computer Science and Technology, Soochow University, Suzhou, China

<sup>2</sup> Department of Computer Science, Aalborg University, Aalborg, Denmark

<sup>3</sup> University of Electronic Science and Technology of China, Chengdu, China

<sup>4</sup> University of Queensland, Brisbane, Australia

individual worker may not be able to perform a complex task (e.g., monitoring the traffic flow in an area or moving heavy stuff) independently since completing the task alone exceeds the capability of this worker. In such scenarios, each task should be assigned to a group of workers, which is named *Group Task Assignment*.

Group task assignment requires a group of workers to perform each task by physically traveling to the location of this task at a particular time. Some previous studies have explored the group task assignment problem in spatial crowdsourcing. For instance, [13] proposes a team-oriented task planning (TOTP) problem, which makes feasible plans for workers and satisfies the skill requirements of different tasks on workers. Gao et al. [14] develop a Top- $k$  team recommendation framework in spatial crowdsourcing, in which a team leader is appointed among each recommended team of workers in order to coordinate different workers conveniently. Cheng et al. [6] consider the collaboration in task assignment, in which workers are required to cooperate and accomplish the tasks jointly for achieving high total cooperation quality scores. Nevertheless, they fail to effectively incorporate the group preference, which is an essential factor for improving the quality of group task assignment in spatial crowdsourcing as the group members may not be willing to perform the task assigned to them when they are not interested in this task. We next illustrate the group task assignment problem through a motivation example.

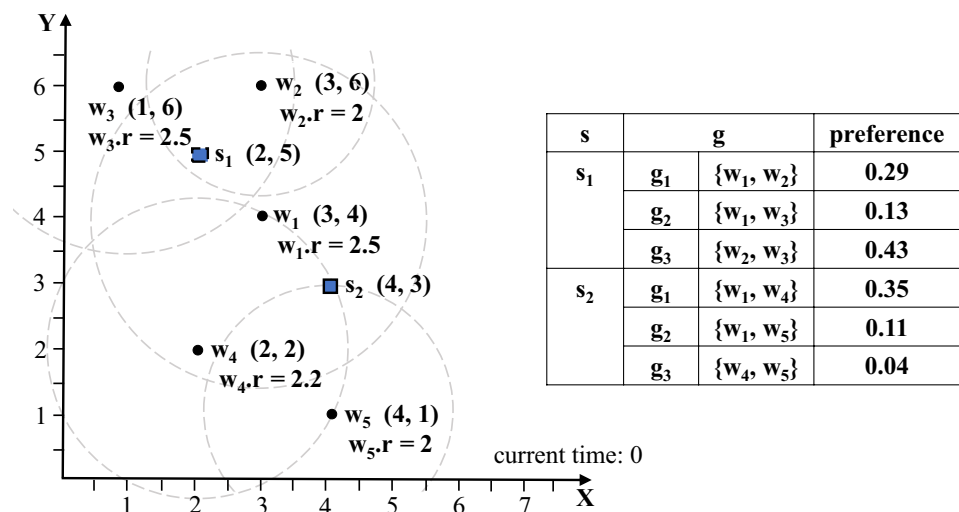
Figure 1 shows an example of the group task assignment problem, in which each task is required to be assigned to two workers. There exist five workers  $w_1, \dots, w_5$  and two tasks  $s_1, s_2$ . Each worker is associated with her current location, her reachable distance range and her movement speed. For the sake of simplicity, we set the movement speed of each worker to 1 in this running example. Each task is labeled with its location where it will be performed.

In addition, Fig. 1 also depicts the preferences of different available worker groups for each task. The problem is to assign tasks to suitable worker groups so as to maximize the total task assignments.

In SC, it is an intuitive move to assign the nearby tasks to workers without violating the spatiotemporal constraints (i.e., the assigned tasks should be located in the reachable ranges of the corresponding workers and workers can arrive in the locations of assigned tasks before the deadlines of tasks). Therefore, we can obtain a task assignment,  $\{<s_1, \{w_1, w_2\}>, <s_2, \{w_4, w_5\}>\}$ , with the overall group preference of 0.33. Nevertheless, when we assign the worker group,  $\{w_4, w_5\}$ , to task  $s_2$ , the group is likely to quit performing  $s_2$  as they show little interest in  $s_2$  (i.e., the group preference on  $s_2$  is only 0.04), which may leave  $s_2$  uncompleted. If we assign tasks by giving higher priorities to the worker groups who are more interested in the tasks, we can get the task assignment result,  $\{<s_1, \{w_2, w_3\}>, <s_2, \{w_1, w_4\}>\}$ , the total group preference of which is 0.78.

In this paper, we develop a group task assignment framework based on worker groups' preferences. The framework is comprised of two primary components. First, we utilize the powerful bipartite graph embedding model (BGEM) [31] and the attention mechanism to learn the embedding of task categories and worker groups in a low-dimensional space from group–task interaction data. In order to overcome the limitations of data sparsity problem, we integrate the worker–task interaction data and social network structure information (which is used for extracting the social impact of workers) during the process of preference modeling. Secondly, we apply the tree-decomposition-based algorithm [35] to assign tasks to worker groups to maximize the task assignments by giving higher priorities to the worker groups that show more interest in the tasks.

Fig. 1 Running example



Although our previous study [17] can exploit social network features by a linear approach, it fails to capture the nonlinear and complex network structures of social network features. Since the underlying network structure is complex and it is necessary to take into consideration the interactions among features in a nonlinear way, we apply an unsupervised deep learning model, called stacked denoising autoencoders (SDAE) [28], to learn the complex interactions among social network features, which is depicted in Sect. 3.3.

The second limitation is that, although our preliminary work [17] has already achieved the optimization goal of maximizing the overall task assignments by taking social impact-based preferences of worker groups into account, it fails to consider the members' disagreement (i.e., reflecting the level at which group members disagree with each other) among group members. In group activities, it is more desirable to participate in an activity that all the group members with high consensus are interested in than to attend an activity that polarizes group members even if the latter has higher preferences among them [19]. As a group activity, an effective group task assignment tends to ask a group of workers with high agreement to perform a task (such as moving a heavy stuff) simultaneously. Therefore, in Sect. 4.2, we combine the members' disagreement with the group preference in the phase of group task assignment. More specifically, we calculate a consensus score for each worker group, which includes two aspects. The first aspect is the worker group's preference (i.e., social impact-based preference), which reflects the degree to which the task is preferred by the worker group members. The second aspect is the group members' disagreement which reflects the level at which members disagree with each other. The aim of the group task assignment is to maximize the total task assignments by giving priority to the worker groups with higher consensus scores on tasks.

As a summary, the major value-added extension over our preliminary work [17] is as follows:

- We identify and study in depth two limitations in our previous framework, which include failing to capture the nonlinear interactions among social network features and failing to consider the disagreement factor among group members.
- We employ the stacked denoising autoencoders (SDAE) method to learn the nonlinear interactions among social network features by exploring their complex inherent structures.
- We incorporate the consensus score into the group task assignment process, which tries to formalize the members' disagreement and group preference to weaken the polarization among group members.
- Extensive experiments are conducted to study the impact of the key parameters and effectiveness of our proposed

algorithms. In particular, compared with the original exact task assignment approach, our proposed task assignment method with the deep learning strategy (i.e., stacked denoising autoencoders) and group consensus strategy can improve the task assignment success rate by up to 31.58% in order to enhance the effectiveness of task completion.

The remainder of this paper is organized as follows. The preliminary concepts and framework are introduced in Sect. 2. We then present the preference modeling algorithm in Sect. 3, including the proposed deep learning method in Sect. 3.3. Next, the task assignment algorithm taking the consensus (in Sect. 4.2) into consideration is presented in Sect. 4, followed by the experimental results in Sect. 5. Section 6 surveys the related work, and Sect. 7 concludes this paper.

## 2 Problem Statement

In this section, we briefly introduce a set of preliminary concepts and then give an overview of our framework. Table 1 summarizes the major notations used in the rest of the paper.

### 2.1 Preliminary Concepts

**Definition 1 (Spatial Task)** A spatial task,  $s = \langle s.l, s.p, s.e, s.c, s.numW \rangle$ , is a task to be performed at location  $s.l$ , published at time  $s.p$ , and will expire at  $s.e$ ,

**Table 1** Summary of notations

Notation	Definition
$s$	Spatial task
$S$	A set of tasks
$s.l$	Location of spatial task $s$
$s.p$	Published time of spatial task $s$
$s.e$	Expiration time of spatial task $s$
$s.c$	Category of spatial task $s$
$s.numW$	Number of workers that $s$ requires to be assigned
$w$	Worker
$W$	A set of workers
$w.l$	Location of worker $w$
$w.r$	Reachable radius of worker $w$
$w.on$	Online time of worker $w$
$w.off$	Offline time of worker $w$
$w.speed$	Movement speed of worker $w$
$AWS(s)$	Available worker set of task $s$
$AWG(s)$	Available worker group of task $s$
$A$	A spatial task assignment

where  $s.l : (x, y)$  is a point in the 2D space. Each task  $s$  is also labeled with a category  $s.c$  (e.g., moving heavy stuff) and  $s.numW$  is the number of workers allowed to be assigned to perform  $s$  at the same time instance.

**Definition 2 (Worker)** A worker,  $w = \langle w.l, w.r, w.on, w.off, w.speed \rangle$ , is a carrier of a mobile device who volunteers to perform spatial tasks. A worker can be in an either online or offline mode. A worker is online when she is ready to accept tasks. An online worker is associated with her current location  $w.l$ , her movement speed  $w.speed$ , her reachable circular range with  $w.l$  as the center and  $w.r$  as the radius, where  $w$  can accept assignment of spatial tasks. Besides, a worker with her online time,  $w.on$ , is also associated with her offline time,  $w.off$ , before which the worker can be assigned tasks.

In our model, a worker can handle only one task at a certain time instance, which is reasonable in practice. Once the server assigns a task to a worker, the worker is considered being offline until she completes the assigned task.

**Definition 3 (Available Worker Set)** Given a task  $s$  to be assigned and a set of workers in the vicinity of  $s$ , the available worker set for task  $s$ , denoted as  $AWS(s)$ , should satisfy the following three conditions:  $\forall w \in AWS(s)$ :

1.  $t_{now} + t(w.l, s.l) \leq s.e$ , and
2.  $d(w.l, s.l) \leq w.r$ , and
3.  $t_{now} + t(w.l, s.l) \leq w.off$ ,

where  $t_{now}$  is the current time,  $t(w.l, s.l) = d(w.l, s.l)/w.speed$  is the travel time from  $w.l$  to  $s.l$  and  $d(w.l, s.l)$  is the travel distance (e.g., Euclidean distance) between  $w.l$  and  $s.l$ .

**Definition 4 (Available Worker Group)** Given a task  $s$  to be assigned and its available worker set  $AWS(s)$ , the available worker group for task  $s$ , denoted as  $AWG(s)$ , should satisfy the following three conditions:

1.  $AWG(s) \subset AWS(s)$ , and
2.  $|AWG(s)| = s.numW$ , and
3.  $\forall w_i, w_j \in AWG(s), t_{now} + t(w_i.l, s.l) \leq w_j.off$ ,

where  $|AWG(s)|$  denotes the number of workers in  $AWG(s)$ .

In the rest of the paper, we will use worker group and group interchangeably when the context is clear.

**Definition 5 (Spatial Task Assignment)** Given a set of workers  $W_i$  and a set of tasks  $S_i$  at time instance  $t_i$ , a spatial task assignment, denoted by  $A_i$ , consists of a set of  $\langle task, AWG \rangle$  pairs in the form of  $\langle s_1, AWG(s_1) \rangle, \langle s_2, AWG(s_2) \rangle, \dots$ . We use  $|A_i|$  to denote the number of task assignments.

**ProblemStatement:** Given a set of workers  $W_i$  and a set of tasks  $S_i$  at the current time instance  $t_i$  on a SC platform, the group task assignment (GTA) problem aims to find the optimal assignment with the maximum number of task assignments (i.e.,  $\max\{|A_i|\}$ ) by considering the consensus among group members.

## 2.2 Framework Overview

As shown in Fig. 2, our framework consists of two major components: (1) social impact-based preference modeling (SIPM) for worker groups; and (2) preference-based group task assignment (PGTA) based on worker groups' preferences.

In the SIPM procedure, inspired by the success of [2, 30] in learning (user) group preference based on both user-item and group-item interaction data, we utilize the bipartite graph embedding model (BGEM) and attention mechanism to obtain each worker group's preference on different categories of tasks by simultaneously leveraging both worker-task and group-task interaction data. Note that we say a worker interacts with a task if she has performed this task. More specifically, we utilize BGEM to model the individual interaction (i.e., worker-task interaction) and group interaction (i.e., group-task interaction) to learn the vector representation of workers and task categories in a low-dimensional space, respectively. Since the worker groups in spatial crowdsourcing are often formed in an ad hoc manner (called occasional groups) without any interaction with

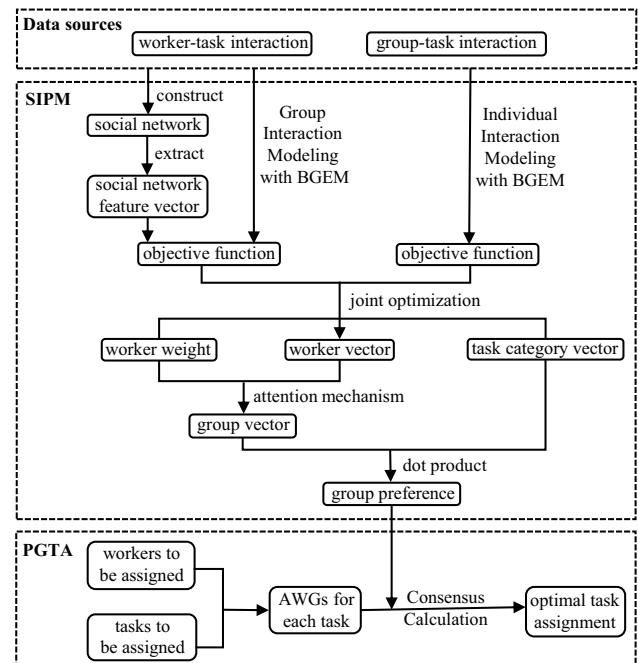


Fig. 2 Framework of our model



tasks, which means the group interaction data is sparse, we cannot effectively learn the vector representation of groups directly. To solve this problem, we introduce workers' social impact that represents workers' weights in a group when making decision about task selection.

In particular, we integrate the worker–task interaction data with group–task interaction data to construct a social network, based on which we extract the social network information by a deep learning method called stacked denoising autoencoders (SDAE). In order to alleviate the sparsity of group–task interaction data, we employ a joint optimization approach to combine group–task interaction data with worker–task interaction data, in which we can obtain the embedding vectors of workers and task categories as well as workers' weights (i.e., workers' social impact). At the same time, the group vector can be calculated by the attention mechanism, which assigns different weights to different workers. Finally, we can obtain the group preference on task categories by taking dot product between group vector and task category vector.

In the PGTA phase, given a set of workers and tasks to be assigned, we first obtain the available worker groups (AWGs) for each task by considering trip constraints, i.e., workers' reachable range, workers' available time and tasks' expiration time. Then, we calculate the consensus scores (including group preference and members' disagreement) of AWGs. Finally, we employ the optimal task assignment (OTA) algorithm based on tree decomposition to assign tasks to suitable worker groups in order to maximize the total task assignments and giving higher priorities to worker groups with higher consensus scores on tasks.

### 3 Social Impact-Based Preference Modeling

In this section, we first elaborate how the bipartite graph embedding model (BGEM) [31] learns each worker's embedding vector (representing her preference on different task categories) and each task category's embedding vector based on the historical worker–task interaction data (a.k.a. individual interaction data). Then, in the group interaction modeling, we extract workers' social impact from the social network, in which we utilize a deep learning method (i.e., SDAE), and employ the attention mechanism [2] to adapt the social impact to different worker groups. Finally, we design a joint optimization strategy, which can obtain the preference of each group on task categories by simultaneously leveraging both worker–task and group–task interaction data.

#### 3.1 Individual Interaction Modeling

Given the interactions between workers and tasks, i.e., worker–task interaction data, we first construct a bipartite

graph,  $\mathcal{G}_{WC} = (W \cup C, E_{WC})$ , where  $W$  denotes the worker set,  $C$  denotes all the categories of tasks,  $W \cup C$  is the node set of  $\mathcal{G}_{WC}$ , and  $E_{WC}$  is the set of edges between workers and task categories. An edge  $e_{ij} (\in E_{WC})$  exists when worker  $w_i (\in W)$  has performed the tasks with category  $c_j (\in C)$ . The weight  $h_{ij}$  of edge  $e_{ij}$  is set as  $h_{ij} = \frac{N_{w_i}^{c_j}}{N_{w_i}}$ , where  $N_{w_i}^{c_j}$  denotes the number of tasks (with category  $c_j$ ) worker  $w_i$  has performed and  $N_{w_i} = \sum_{c \in C} N_{w_i}^c$  denotes the total number of tasks  $w_i$  has performed.

Due to the success of BGEM [31] in learning the embedding of heterogeneous interaction entities, we employ it to model the individual worker–task interaction. For the given worker  $w_i$ , the probability of  $w_i$  interacting with the tasks with category  $c_j$  can be calculated in the following:

$$p(c_j|w_i) = \frac{\exp(\mathbf{w}_i \cdot \mathbf{c}_j)}{\sum_{c \in C} \exp(\mathbf{w}_i \cdot \mathbf{c})}, \quad (1)$$

where  $\mathbf{w}_i$  is the embedding vector of worker  $w_i$  representing her preference and  $\mathbf{c}_j$  is the embedding vector of task category  $c_j$ .

In the sequel, we define the objective function of the BGEM. As we all know from [18], the target of BGEM is to minimize the KL divergence between  $\hat{p}(\cdot|w_i)$  and  $p(\cdot|w_i)$ , which represent the empirical distribution and the estimated neighbor probability distribution for each worker  $w_i \in W$ , respectively.

We employ  $d_i$  to represent the outdegree of worker node  $w_i$ , which can be calculated as  $d_i = \sum_{c_j \in C} h_{ij}$  (where  $h_{ij}$  denotes the weight of the edge  $e_{ij}$ ). We define the empirical distribution  $\hat{p}(c_j|w_i) = h_{ij}/d_i$ . Thus, the objective function can be obtained as follows:

$$\begin{aligned} O_{WC} &= - \sum_{e_{ij} \in E_{WC}} h_{ij} \log p(c_j|w_i) \\ &= - \sum_{e_{ij} \in E_{WC}} \frac{N_{w_i}^{c_j}}{N_{w_i}} \log \frac{\exp(\mathbf{w}_i \cdot \mathbf{c}_j)}{\sum_{c \in C} \exp(\mathbf{w}_i \cdot \mathbf{c})}. \end{aligned} \quad (2)$$

#### 3.2 Group Interaction Modeling

In the similar way, we construct a bipartite graph, i.e.,  $\mathcal{G}_{GC} = (G \cup C, E_{GC})$ , to represent the interactions between groups and task categories, where  $G$  is a set of groups,  $G \cup C$  is the node set of  $\mathcal{G}_{GC}$ , and  $E_{GC}$  represents a set of edges between groups and task categories. There exists an edge  $e_{ij} (\in E_{GC})$  between group  $g_i (\in G)$  and task category  $c_j (\in C)$  if this group of workers has performed the tasks with category  $c_j (\in C)$ . Moreover, the weight  $h_{ij}$  of the edge  $e_{ij}$  is simply set as  $h_{ij} = \frac{N_{g_i}^{c_j}}{N_{g_i}}$ , where  $N_{g_i}^{c_j}$  denotes the number of tasks (with category  $c_j$ ) worker group  $g_i$  has

performed and  $N_{g_i}$  denotes the total number of tasks  $g_i$  has performed. Let  $\mathbf{g}_i$  be the embedding vector for group  $g_i$  and  $\mathbf{c}_j$  be the embedding vector for task category  $c_j$ . Our target is to obtain an embedding vector for each worker group to estimate the preference on all the task categories.

The objective function in group–task interaction data, which is similar to the worker–task interaction data, can be calculated in the following:

$$O_{GC} = - \sum_{e_{ij} \in E_{GC}} h_{ij} \log p(c_j | g_i) = - \sum_{e_{ij} \in E_{GC}} \frac{N_{g_i}^{c_j}}{N_{g_i}} \log \frac{\exp(\mathbf{g}_i \cdot \mathbf{c}_j)}{\sum_{c \in C} \exp(\mathbf{g}_i \cdot \mathbf{c})}. \quad (3)$$

Nevertheless, in reality, there are few persistent groups, while there are large amounts of occasional groups forming in an ad hoc manner to perform a task in spatial crowdsourcing. As a result, the group–task interaction data are over sparse with the cold-start nature (i.e., there is no or little group–task interaction) of occasional groups, which leads it difficult to directly learn the embedding vector of an occasional group. To tackle the sparsity and cold-start issue, we aggregate the embeddings of all the members in a group from the group–task interaction data. We observe that in decisions such as task selection, some group members may outspoke others in expressing their preference (due to prestige, authority, or other personality factors) and thus are more influential on the group’s choice on tasks. In addition, the same worker in different groups may have different contributions on group’s decision-making. Therefore, we introduce a coefficient  $\alpha(k, i)$  to learn the weight of worker  $w_k$  in group  $g_i$ , which represents the group-aware personal social impact of  $w_k$  in deciding the choice of group  $g_i$  on tasks. Specifically, given an occasional group  $g_i$ , we define the embedding vector  $\mathbf{g}_i$  as follows:

$$\mathbf{g}_i = \sum_{w_k \in g_i} \alpha(k, i) \mathbf{w}_k, \quad (4)$$

where  $\alpha(k, i)$  is a learnable parameter (where a higher value indicates greater impact on a group’s decision) and  $\mathbf{w}_k$  denotes the embedding of worker  $w_k$ .

However, occasional groups temporarily gather together to perform a task in a time instance. It is difficult to learn the coefficient  $\alpha(k, i)$  directly from the group–task interaction data because of the extreme data sparsity problem. Therefore, we introduce an additional positive numerical value  $\lambda_k$  for each worker  $w_k$  representing the global personal social impact, which does not depend on specific groups. We employ  $\exp(\lambda_k)$  to represent the relative impact on deciding a group choice on tasks. Thus,  $\alpha(k, i)$  can be calculated in Eq. 5, which is inspired by the attention mechanism [2].

$$\alpha(k, i) = \frac{\exp(\lambda_k)}{\sum_{w_k \in g_i} \exp(\lambda_k)}. \quad (5)$$

It is obvious that once we obtain the  $\lambda_k$  representing the global personal social impact for each worker  $w_k$ , we can easily obtain the  $\alpha(k, i)$ , which represents the group-aware personal social impact in a group. However, if a worker has only participated in very few group activities, it may suffer from over-fitting problems. Moreover, if a worker has never attended any group activities, we are not capable of learning the global personal social impact. As a result, we cannot learn the satisfying social impact only from the group–task interaction data.

In order to improve the accuracy of global personal social impact estimation, we construct a workers’ social network based on both worker–task and group–task interaction data, based on which we extract the social network information, which benefits workers’ global social impact estimation. In the social network, each worker maps to a node and an edge exists if two workers have cooperated with each other in the same group. The weight of the edge is set as the number of cooperations between the workers. Each worker (node) is associated with the number of tasks she has completed. Then, we extract the social network structure information by various measures (e.g., degree centrality and betweenness centrality) and integrate the social network structure information into the learning process of worker’s global social impact, which effectively alleviates the cold-start problem in group–task interaction data.

In particular, we can calculate a social network feature vector  $\boldsymbol{\beta}_k$  for worker  $w_k$  and employ a feature selector vector  $\mathbf{h}$  to assign different weights to different structure features [30]. We normalize all the feature values into the range [0, 1]. Then, we take dot product between the social network feature vector  $\boldsymbol{\beta}_k$  and the feature selector vector  $\mathbf{h}$  as the Gaussian prior for the global personal social impact of worker, i.e.,  $\lambda_k \sim (\boldsymbol{\beta}_k \cdot \mathbf{h} + b, \rho_V^2)$  ( $b$  is a bias term). Due to the fact that global personal social impact may be affected by other unknown factors, we assume that  $\lambda_k$  follows the normal distribution with the mean  $\boldsymbol{\beta}_k \cdot \mathbf{h} + b$  to learn the more robust personal global social impact.

In terms of the objective function, we should add a corresponding regularization term  $R_V$ , i.e.,  $\frac{1}{2\rho_V^2} \sum_{w_k \in W} (\lambda_k - (\boldsymbol{\beta}_k \cdot \mathbf{h} + b))^2$ , into the objective function since we introduce a Gaussian prior for the personal social impact parameter  $\lambda_k$ . The hyper-parameter  $\rho_V^2$  (i.e., variance) can control the weight of the regularization term. Therefore, the new objective function is as follows:

$$O_{VGC} = O_{GC} + R_V. \quad (6)$$

Considering the cold-start issue in group–task interaction data, we combine worker–task interaction data with



group–task interaction data during the optimization process. More specifically, we design a joint optimization approach, which can simultaneously learn the embedding vectors of workers and task categories from the worker–task interaction data and group–task interaction data. Besides, the global social impact of workers can be learned during the optimization process. Therefore, we combine  $O_{VGC}$  and  $O_{WC}$  to form a joint objective function, which is simply defined as follows:

$$O_{GWC} = O_{VGC} + O_{WC}. \quad (7)$$

Here, we adopt the standard stochastic gradient descent (SGD) strategy [4] to minimize the objective function  $O_{GWC}$  in Eq. 7, as a result of which each worker's embedding vector  $\mathbf{w}$ , each task category's embedding vector  $\mathbf{c}$  and the model parameters (i.e.,  $\lambda_k, \mathbf{h}$ ) can be learned. We can calculate the coefficient  $\alpha(k, i)$  representing the group-aware personal social impact according to Eq. 5. Then, each group's embedding vector  $\mathbf{g}$  can be correspondingly obtained based on Eq. 4. Finally, we take dot product between each group's embedding vector and each task category's embedding vector to achieve the preference of each group on each task category.

### 3.3 Deep Social Impact Learning

Since the underlying network structures are complex, shallow models cannot capture the highly nonlinear network structures. In order to tackle this problem, we adopt the stacked denoising autoencoders (SDAE) [28] model to learn the latent representation of social network features for worker  $w_i$ , denoted as  $\mathbf{x}_i$ , from the original social network feature vector  $\beta_i$ . This is inspired by the success of deep learning, which has a powerful representation ability to learn complex structures of data [3]. As an unsupervised neural network model of codings, SDAE aims to learn the representation of the corrupted input data to predict the clean input itself in the output. SDAE consists of two parts, the encoder and the decoder, which contains multiple nonlinear functions (i.e., layers) for mapping the corrupted input to a latent representation space and then maps the representation into the representation space to reconstruct the original clean input. The composition of multiple layers of nonlinear functions can map the data into a highly nonlinear space, thereby being able to capture the highly nonlinear network structures.

More specifically, we take the initial social network features of all workers (denoted as  $\mathbf{X}_c$ ) as input and then corrupt the initial input to get a partially destroyed version (denoted as  $\mathbf{X}_0$ ) by randomly choosing some elements of  $\mathbf{X}_c$  to be forced to 0. Given the corrupted input  $\mathbf{X}_0$ , the output of each layer  $k$ , denoted as  $\mathbf{X}_k$ , can be generated by the means of  $\mathbf{X}_k \sim \mathcal{N}(\sigma(\mathbf{X}_{k-1} \mathbf{W}_k + \mathbf{b}_k), \rho_X^2 \mathbf{I})$ , where  $\mathbf{I}$  denotes the identity

matrix and  $\rho_X^2$  is a regularization hyperparameter. The weight and bias parameters of layer  $k$  can be represented as  $\mathbf{W}_k$  (i.e.,  $\mathbf{W}_k \sim \mathcal{N}(\mathbf{0}, \rho_W^2 \mathbf{I})$ ) and  $\mathbf{b}_k$  (i.e.,  $\mathbf{b}_k \sim \mathcal{N}(\mathbf{0}, \rho_b^2 \mathbf{I})$ ), respectively, where  $\rho_W^2$  and  $\rho_b^2$  are regularization hyperparameters. Following the process above, the optimization function of SDAE can be defined as follows:

$$O_{SDAE} = \frac{1}{2} \sum_k \left( \frac{\|\sigma(\mathbf{X}_{k-1} \mathbf{W}_k + \mathbf{b}_k) - \mathbf{X}_k\|_2^2}{\rho_X^2} + \frac{\|\mathbf{W}_k\|_2^2}{\rho_W^2} + \frac{\|\mathbf{b}_k\|_2^2}{\rho_b^2} \right), \quad (8)$$

where  $\|\cdot\|_2^2$  denotes the Euclidean norm and  $\sigma(\cdot)$  is a sigmoid function. The output of the hidden layer in the middle is the latent representation of the social network features. Then, we can take dot product between the latent vector  $\mathbf{x}_i$  and the feature selector vector  $\mathbf{h}$  as the Gaussian prior for the global personal social impact of worker  $w_i$ , i.e.,  $\lambda_i \sim \mathcal{N}(\mathbf{x}_i \cdot \mathbf{h} + b, \rho_V^2)$ , where  $\mathbf{x}_i$  is the  $i$ th row of the latent representation of the social network features,  $b$  is a bias term,  $\rho_V^2$  is a hyperparameter and  $\lambda_i$  denotes the global personal social impact.

Finally, we combine the SDAE model with our proposed group interaction learning framework and the optimization function in Eq. 6 can be converted into:

$$O_{VGC} = O_{GC} + O_{SDAE} + R_V, \quad (9)$$

where  $O_{GC}$  denotes the objective function in group–task interaction data and  $R_V$  is a regularization term. Then, we can employ the original method (i.e., SGD) to optimize the objective function in Eq. 7.

## 4 Group Task Assignment

In this section, we first generate the available worker groups for each task based on the trip constraints (i.e., workers' reachable range, workers' available time and tasks' expiration time). Then, we calculate the consensus scores of available worker groups, and finally a tree-decomposition-based algorithm [32, 35] is employed to achieve the optimal task assignment.

### 4.1 Available Worker Group Set Generation

#### 4.1.1 Finding the Reachable Workers for Each Task

Due to the constraint of workers' reachable distance, workers' available time and tasks' expiration time, each task can be completed by a small subset of workers in a time instance. Therefore, we firstly find the set of workers that can complete each task without violating the constraints.

The reachable worker subset for a task  $s$ , denoted as  $RW_s$ , should satisfy the following conditions:  $\forall w \in RW_s$ :

1.  $t_{\text{now}} + t(w.l, s.l) \leq s.e$ , and
2.  $d(w.l, s.l) \leq w.r$ , and
3.  $t_{\text{now}} + t(w.l, s.l) \leq w.off$ ,

where  $t_{\text{now}}$  denotes the current time,  $t(w.l, s.l)$  is the travel time from  $w.l$  to  $s.l$  and  $d(w.l, s.l)$  denotes the travel distance (e.g., Euclidean distance) between  $w.l$  and  $s.l$ . The above three conditions guarantee that a worker  $w$  can travel from her location  $w.l$  to a task  $s$  (which is located in her reachable range) directly before task  $s$  expires and during worker  $w$ 's available time.

#### 4.1.2 Finding the Available Worker Group Sets for Each Task

Given the reachable workers for each task  $s$ , we next find the set of available worker group, denoted as  $AWG(s)$ , under the constraints of workers' available time in a group and the number of workers allowed to be assigned to perform a task  $s$ . Each available worker group in  $AWG(s)$ , denoted as  $AWG(s)$ , should satisfy the following conditions:

1.  $|AWG(s)| = s.numW$ , and
2.  $\forall w_j, w_k \in AWG(s), t_{\text{now}} + t(w_j.l, s.l) \leq w_k.off$ ,

where  $|AWG(s)|$  is the number of workers in  $AWG(s)$ . The above two conditions guarantee that workers in a group can arrive at the location of task  $s$  without violating the available time of each other.

## 4.2 Consensus Calculation

The group task assignment aims to assign each task to a group of workers that show interests and preferences on the task. However, group members may not always share the same tastes. Therefore, the group task assignment process should manage the heterogeneity of worker groups. In this part, we design a consensus score for each available worker group, which contains two main aspects: group preference and group members' disagreement. In particular, the group preference reflects the degree to which the task is preferred by all the members. The more the group members prefer a task, the higher its consensus score should be for the group. The group members' disagreement reflects the level at which members disagree with each other. Group members who disagree with each other are not willing to conduct a task together, leading a low consensus score of the group.

We next calculate the consensus score for each available worker group that consists of two components, i.e., group preference and group members' disagreement. More

specifically, the group preference is the social impact-based preference, which is given in Sect. 3. The disagreement of a group  $g$  over a task category  $c$ , denoted as  $dis(g, c)$ , is the deviation of individual worker preference from her group's average preference. Therefore, the consensus score of  $g$  for task category  $c$ , denoted by  $con(g, c)$ , can be computed by the following formula:

$$con(g, c) = \mathbb{P}(g, c) \cdot \eta(g, c), \quad (10)$$

$$\mathbb{P}(g, c) = \mathbf{g} \cdot \mathbf{c}, \quad (11)$$

$$\eta(g, c) = 1 - \min\{1, dis(g, c)\}, \quad (12)$$

$$dis(g, c) = \frac{1}{|g|} \sum_{w \in g} (\mathbb{P}_{\text{indiv}}(w, c) - \mathbb{P}_{\text{mean}}(g, c))^2, \quad (13)$$

$$\mathbb{P}_{\text{indiv}}(w, c) = \mathbf{w} \cdot \mathbf{c}, \quad (14)$$

$$\mathbb{P}_{\text{mean}}(g, c) = \frac{1}{|g|} \sum_{w \in g} \mathbb{P}_{\text{indiv}}(w, c), \quad (15)$$

where  $\mathbb{P}(g, c)$  denotes the preference of group  $g$  for task category  $c$  by taking the dot product between group vector  $\mathbf{g}$  and task category vector  $\mathbf{c}$  (shown in Eq. 11).  $\eta(g, c)$  is a function calculating the discount to the group's preference on the basis of group members' disagreement, and  $dis(g, c)$  is the disagreement among group members that represents the deviation of group members' preferences from the group's average preference.  $\mathbb{P}_{\text{indiv}}(w, c)$  denotes the preference of individual worker  $w$  for task category  $c$ , which can be computed by taking the dot product between worker vector  $\mathbf{w}$  and task category  $\mathbf{c}$  (shown in Eq. 14) and  $\mathbb{P}_{\text{mean}}(g, c)$  denotes the mean of the preferences of all the members in group  $g$  for task category  $c$ .  $\mathbf{g}$ ,  $\mathbf{c}$  and  $\mathbf{w}$  are obtained in Sect. 3. Note that we normalize the preference values (e.g.,  $\mathbb{P}(g, c)$  and  $\mathbb{P}_{\text{indiv}}(w, c)$ ) to lie between 0 and 1, using a Min–Max normalization procedure.

## 4.3 Optimal Group Task Assignment Algorithm

It is easy to know that the global optimal result is the union of one possible available worker group (AWG) of all tasks. We introduce an algorithm, i.e., tree-decomposition-based strategy [32, 35], to achieve the optimal task assignment with the maximal task assignments, in which we give higher priority to the worker groups with higher consensus scores on tasks. More specifically, we first construct a task dependency graph,  $G(V, E)$ , according to the dependency relationship among tasks. (Two tasks are dependent with each other if they share the available workers; otherwise, they are independent.) We consider that each vertex  $v \in V$  represents a

task  $s_v \in S$ . There exists an edge  $e(u, v) \in E$  between  $u$  and  $v$  if two tasks  $s_u$  and  $s_v$  are dependent with each other. Subsequently, we utilize a tree decomposition strategy to separate all tasks into independent clusters, which are the maximal cliques of the task dependency graph. Then, we utilize a recursive tree construction (RTC) algorithm [32, 35] to organize them into a balance tree structure, such that the tasks in sibling nodes of the tree do not share the same available workers. Facilitated by such a tree structure, we can solve the optimal assignment subproblem on each sibling node independently. Finally, the optimal assignment result can be found by a depth-first search through the tree, during which we assign tasks to the available worker groups with higher consensus scores on tasks.

## 5 Experiment

In this section, we conduct extensive experiments on a real-world dataset to evaluate the performance of our proposed algorithms. All the algorithms are implemented on an Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20 GHz with 256 GB RAM.

### 5.1 Experiment Setup

We conduct our experiments on a check-in dataset from Twitter, which provides check-in data across USA except Hawaii and Alaska from September 2010 to January 2011 including locations of 62,462 venues and 61,412 users. The dataset is used widely in evaluation of SC platform [11]. Due to the lack of category information of venues in dataset, we generate the category information (i.e., task category information) associated with each venue from Foursquare with the aid of its API. When using the dataset in our experimental research, we assume that the users in dataset are the workers of SC platform since users who check in to different spots may be good candidates to perform spatial tasks in the vicinity of those spots, and their locations are those of the most recent check-in points. We assume the spots are the tasks of SC platform and employ its location and earliest check-in time of the day as the location and publish time of a task, respectively. We consider each worker's average travel distance as the worker's movement speed. We extract 20 kinds of check-in categories to simulate the task categories, i.e., the categories of check-ins. Checking in a spot is equivalent to accepting a task.

As Twitter does not contain explicit group information, we extract implicit group task completion activities as follows: We assume if a set of users visit the same spot or different spots with the same category which are near to each other (e.g., the distance between any two spots is less than

10km in our experiments) in one hour, they are regarded as the members of a group.

The default values of all parameters used in our experiments are summarized in Table 2.

## 5.2 Experimental Results

### 5.2.1 Performance of Preference Modeling

In this experiment, we evaluate the efficiency of worker groups' preference modeling phase and its impact to the subsequent task assignment. Specifically, we compare the efficiency (i.e., CPU time) of the following algorithms:

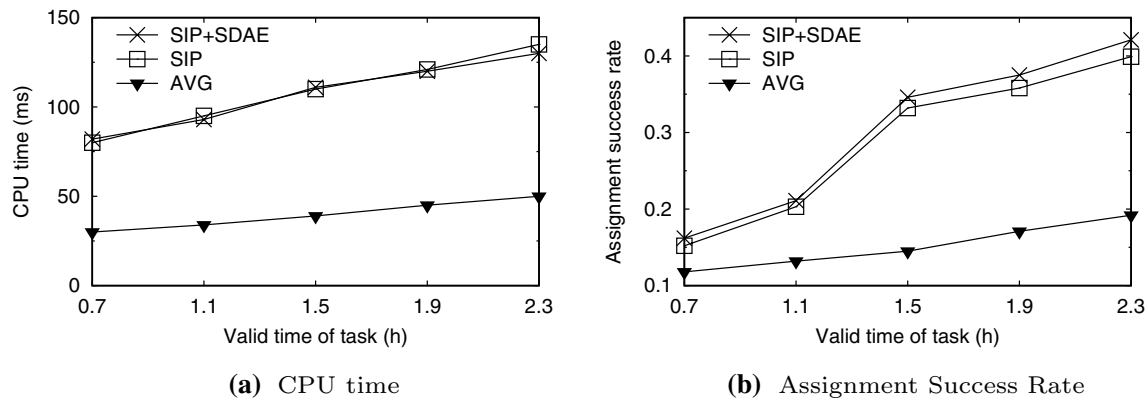
1. AVG: Average preference calculation (AVG) method, where the average preference of a group  $g$  is computed by  $\frac{N_g^c}{N_g}$ .  $N_g^c$  denotes the number of tasks (with category  $c$ ) worker group  $g$  has performed and  $N_g$  denotes the total number of tasks  $g$  has performed.
2. SIP: the social impact-based preference (SIP) modeling algorithm by employing the linear approach for exploiting social network features.
3. SIP+SDAE: the social impact-based preference (SIP) modeling algorithm by employing stacked denoising autoencoders (SDAE).

For effectiveness of task assignment based on the above preference modeling methods, we compare the assignment success rate (ASR) by applying the optimal group task assignment (OGTA) algorithm. ASR is the ratio of successful assignments to the total assignments for all workers in a time instance. Note that once all the group members actually perform (check in) the tasks (spots) with the same category which are near to each other (e.g., the distance between the tasks is less than 10 km in our experiments) in one hour, we regard this task assignment as a successful assignment.

*Effect of  $e - p$ .* First, we investigate how the valid time of tasks,  $e - p$ , affects the efficiency of preference modeling and the effectiveness of group task assignment. As we can see from Fig. 3a, the CPU time of all the methods increases gradually when  $e - p$  is longer, as there are more workers

**Table 2** Experiment parameters

Parameter	Default value
Valid time of tasks $e - p$	1.5 h
Number of workers of each group $numW$	2
Workers' reachable radius $r$	2 km
Workers' available time off-on	3 h
Number of tasks $ S $	3000
Number of workers $ W $	3000



**Fig. 3** Performance of preference modeling: effect of  $e - p$

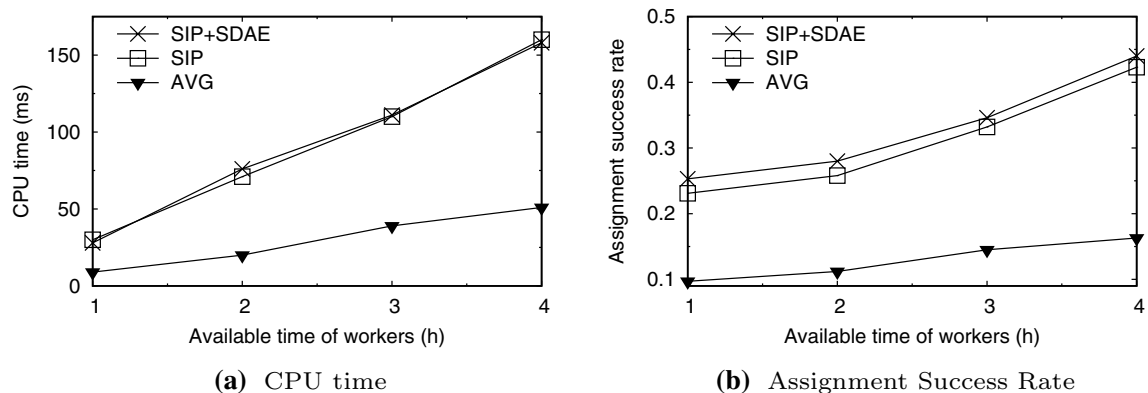
and tasks to be processed. SIP+SDAE and SIP have similar performance since both of them calculate the group preference by taking the dot product process between group vector and task category vector (that can be obtained by the SDAE method and the linear method in the training phase). Although AVG is the least-consuming, it performs worst in terms of ASR (Fig. 3b). In Fig. 3b, the optimal group task assignment algorithm based on SIP+SDAE achieves the highest accuracy, followed by SIP and AVG. Compared with SIP, the assignment accuracy of SIP+SDAE is higher, which confirms the superiority of taking SDAE into consideration. This is due to the fact that SDAE can better capture the nonlinear interactions among social network features to improve the estimation of workers' global personal social impact.

**Effect of off-on.** Figure 4 illustrates the effect of off-on on the performance of all algorithms. As expected in Fig. 4a, increasing workers' available time will incur more CPU time for all the algorithms. This may be due to the fact that, when the available time is more relaxed, more worker groups will be generated, which means we have to compute

the preferences for more worker groups. The ASR values of all the methods are enhanced with the increasing off-on, which is depicted in Fig. 4b, since a worker group has more chance to be assigned their interested tasks when off-on grows. SIP+SDAE performs better than SIP in terms of ASR, confirming the advantage of using SDAE during preference modeling. The reason behind it is that the workers' personal social impact learned by the deep learning model is more accurate.

**Effect of  $r$ .** Next, we study the effect of workers' reachable range  $r$ . The CPU time generated by all approaches has a growing tendency as  $r$  being enlarged (Fig. 5a), with the similar reason of the effect of tasks' valid time, i.e., the larger the workers' reachable regions are, the more available worker groups for each task need to be processed. From Fig. 5b, the ASR of all the approaches increases with the enlarged  $r$  and SIP+SDAE is better than other methods, since group members have more chance to be assigned their interested tasks when  $r$  is larger.

**Effect of  $numW$ .** We study the effect of  $numW$  in this set of experiments. As demonstrated in Fig. 6a, it is worth noting



**Fig. 4** Performance of preference modeling: effect of off-on

that the CPU time is gradually decreasing as the number of each group being enlarged, since the number of the available worker groups for each task decreases when  $numW$  gets larger. The ASR of all methods shows a decreasing trend (Fig. 6b), the reason behind which is that we cannot assign the tasks to the suitable groups because of the less available worker groups. However, the optimal group task assignment method based on SIP+SDAE still shows a higher superiority

than other methods, which confirms the effectiveness of our proposed methods.

*Effect of  $lSI$ .* In this part of experiments, we evaluate the scalability of all the approaches by varying the number  $lSI$  of tasks from 1 to 5 k. As depicted in Fig. 7a, all the methods become time-consuming when  $lSI$  increases since more tasks and available worker groups need to be processed. The optimal group task assignment method based on SIP+SDAE

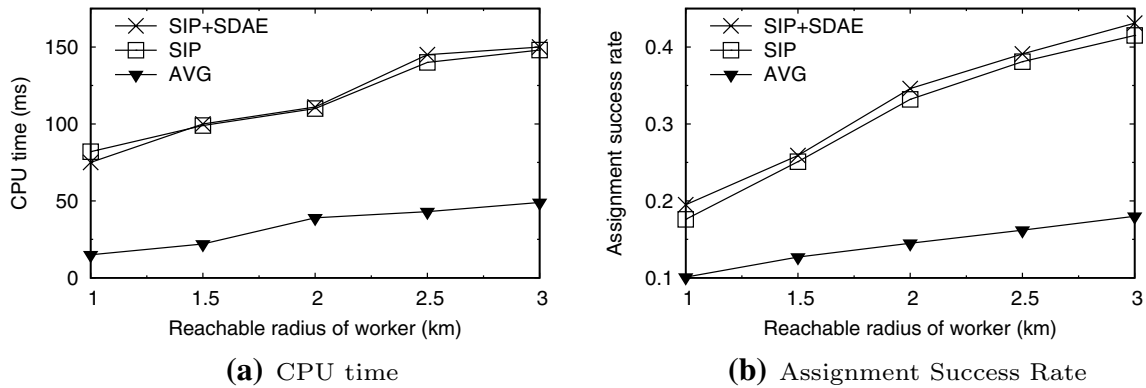


Fig. 5 Performance of preference modeling: effect of  $r$

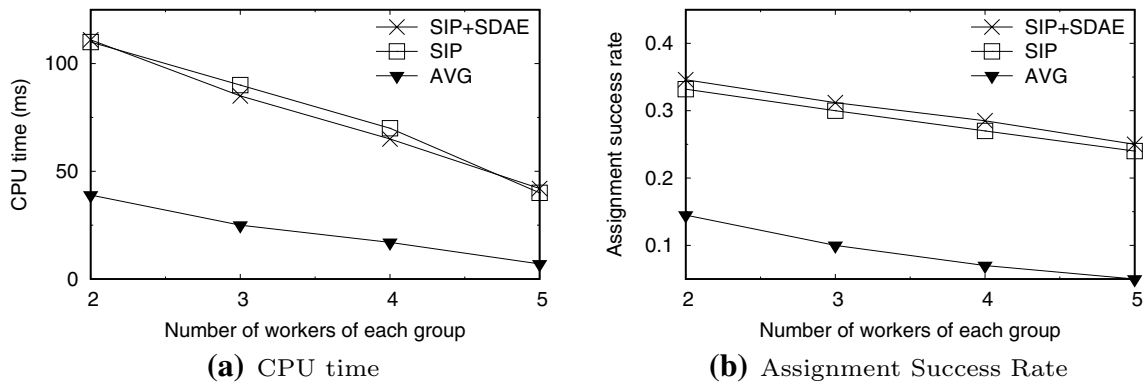


Fig. 6 Performance of preference modeling: effect of  $numW$

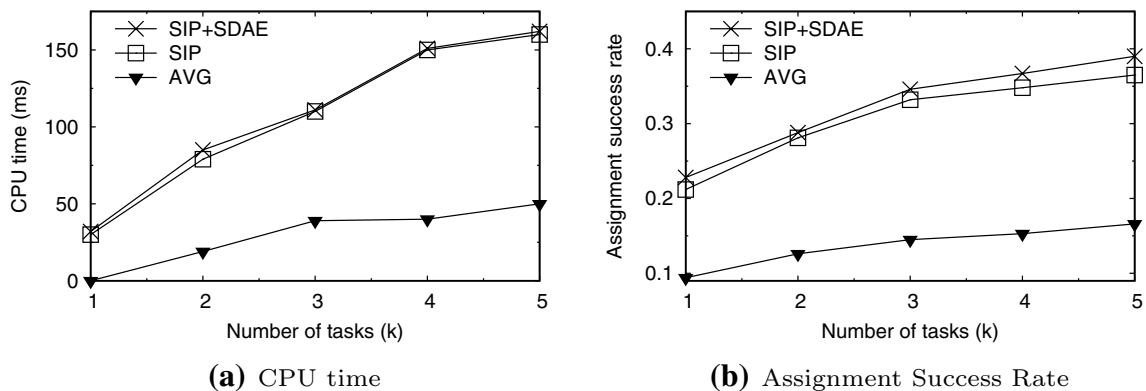


Fig. 7 Performance of preference modeling: effect of  $lSI$

still outperforms other methods (i.e., SIP and AVG) in ASR, which is shown in Fig. 7b.

**Effect of  $|W|$ .** Finally, we measure the performance of our methods by expanding the number of workers ( $|W|$ ) from 1 to 5 k. We can see from Fig. 8a naturally, the running time of all methods increases when the number of workers gets larger. The main reason behind it is that the number of available workers to be assigned grows when  $|W|$  gets larger, which in turn leads to longer time cost. The assignment success accuracy of all algorithms shows an increasing trend, which is shown in Fig. 8b. SIP+SDAE has the highest assignment success rate, demonstrating the superiority of our proposed method in modeling worker groups' preference.

### 5.2.2 Performance of Group Task Assignment

In this part, we evaluate the efficiency and effectiveness of the group task assignment approaches in terms of CPU time, assignment success rate (ASR) and the overall number of task assignments. Specially, the CPU time is given by the average time cost of performing task assignment at each time instance, the ASR and number of assigned tasks can measure

the quality of task assignment strategies. We compare and evaluate the performance of the following methods:

1. OGTA: optimal group task assignment (OGTA) algorithm based on tree decomposition algorithm without considering worker groups' preference.
2. SIP+SDAE-OGTA: OGTA algorithm with worker groups' social impact-based preference calculated by the SIP+SDAE method.
3. SIP+SDAE+DIS-GGTA: greedy group task assignment (GGTA) algorithm with both worker groups' social impact-based preference (calculated by the SIP+SDAE method) and group members' disagreement (DIS). A basic greedy task assignment algorithm is introduced to assign each task greedily to the worker groups with the maximal consensus until all the tasks are assigned or all the worker groups are exhausted.
4. SIP+SDAE+DIS-OGTA: OGTA algorithm with both worker groups' social impact-based preference (calculated by the SIP+SDAE method) and group members' disagreement (DIS).

**Effect of  $e - p$ .** We first study the effect of the valid time  $e - p$  of tasks. As illustrated in Fig. 9a, longer expiration

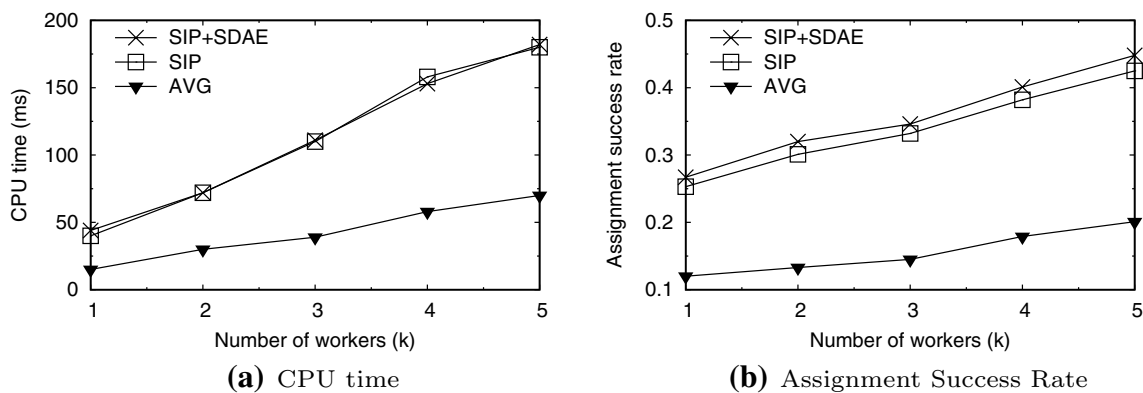


Fig. 8 Performance of preference modeling: effect of  $|W|$

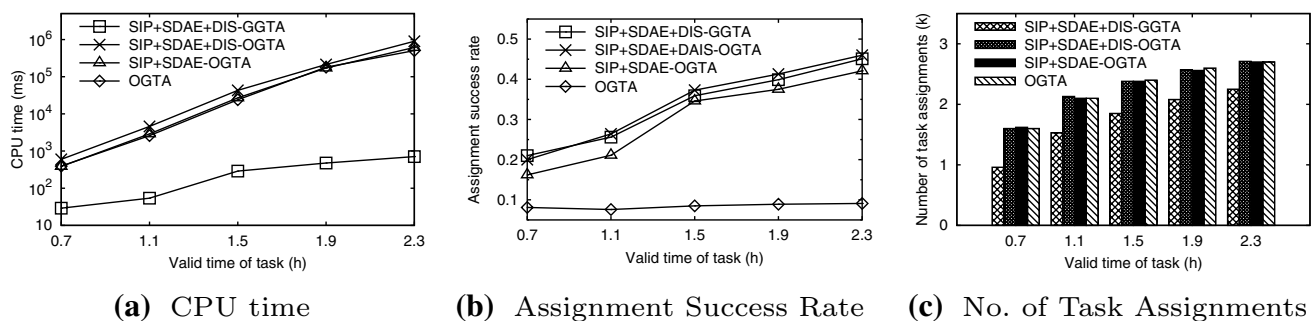


Fig. 9 Performance of group task assignment: effect of  $e - p$



time means on average each worker group has more freedom to choose tasks, which results in greater search space. SIP+SDAE+DIS-OGTA runs slower than other methods as it must compute the consensus score of worker group for the reachable tasks during the task assignment. On the other hand, as shown in Fig. 9b, the accuracy of all algorithms except OGTA has an increasing trend when  $e - p$  grows longer. This is due to the fact that a worker group has more chance to be assigned their interested tasks with the growing valid time of tasks. OGTA keeps almost constant as it does not consider worker group's preference. SIP+SDAE+DIS-OGTA and SIP+SDAE+DIS-GGTA outperform the SIP+SDAE-OGTA algorithm for all values of  $e - p$  in terms of ASR, which indicates that considering the consensus score is beneficial to assignment accuracy. The reason behind it is that it is more desirable to assign a task that each worker group member is interested in than to assign a task that polarizes group members. Although SIP+SDAE+DIS-GGTA is fastest among all the methods and has the similar ASR with SIP+SDAE+DIS-OGTA, it assigns less tasks compared with other methods (i.e., OGTA, SIP+SDAE+DIS-OGTA, SIP+SDAE-OGTA), as shown in Fig. 9c.

*Effect of off-on.* In this set of experiments, we evaluate the effect of the workers' available time. Obviously, from Fig. 10a, the running time of all algorithms increases with the longer workers' available time, since there are more available worker groups for each task to be searched. SIP+SDAE+DIS-OGTA is more time-consuming than other algorithms. The ASR of SIP+SDAE+DIS-OGTA and SIP+SDAE+DIS-GGTA methods consistently outperform other methods, and SIP+SDAE+DIS-OGTA is slightly higher than SIP+SDAE+DIS-GGTA (Fig. 10b). The ASR of all the methods except OGTA has a similar tendency with  $e - p$  when off-on grows, with the similar reason that worker groups have more chance to obtain their interested tasks with the increasing available time off-on. From Fig. 10c, we can see that the number of task assignments grows quickly when workers' available time is longer, the reason behind which is that there are more available worker groups for each task as workers' available time gets longer.

*Effect of  $r$ .* Next, we evaluate the effect of  $r$ , the range of workers' reachable radius. Not surprisingly, as we can see in Fig. 11a, the running time of all methods increases when  $r$  grows, while the growth of computational cost for SIP+SDAE+DIS-OGTA is faster. The ASR of SIP+SDAE+DIS-OGTA and SIP+SDAE+DIS-GGTA is higher than SIP+SDAE-OGTA and OGTA methods,

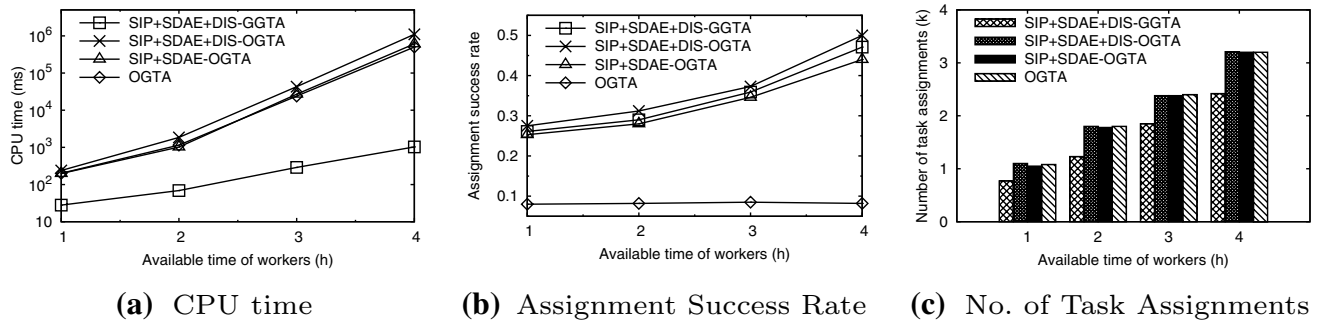


Fig. 10 Performance of group task assignment: effect of off-on

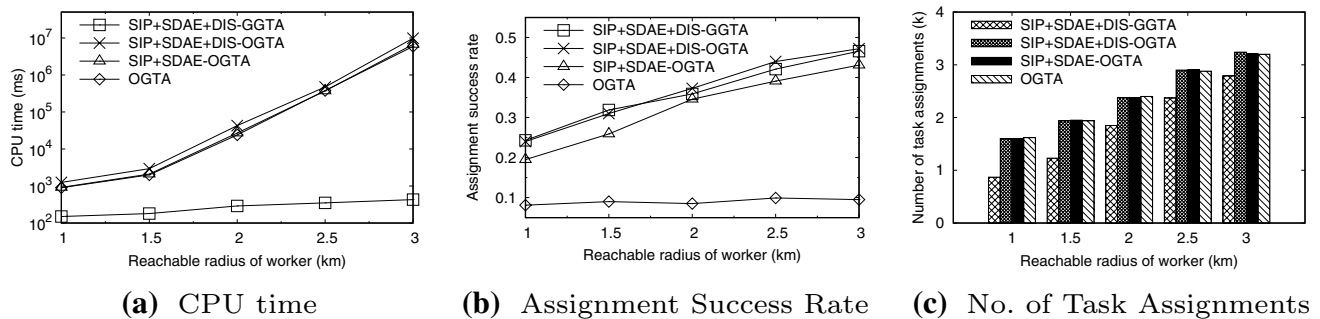


Fig. 11 Performance of group task assignment: effect of  $r$

which confirms the superiority of considering the consensus score for each worker group (Fig. 11b). Although SIP+SDAE+DIS-OGTA and SIP+SDAE+DIS-GGTA show a similar performance in ASR, SIP+SDAE+DIS-OGTA assigns more tasks than SIP+SDAE+DIS-GGTA, which is depicted in Fig. 11c. The number of task assignments generated by all approaches has a growing tendency as  $r$  being enlarged, since the larger the workers' reachable regions are, the more chance the SC server can assign more tasks to the suitable worker groups.

**Effect of  $numW$ .** Figure 12 shows the effect of  $numW$ . Figure 12a illustrates the CPU time of different methods. As expected, the running time shows a downward tendency when the number of workers of each group (i.e.,  $numW$ ) gets larger. It is due to the fact that there are less available worker groups for each task as  $numW$  gets larger, which reduces the search space. SIP+SDAE+DIS-OGTA is still the most time-consuming method. The assignment success rate of all algorithms declines with  $numW$ , as shown in Fig. 12b. However, the method based on SIP+SDAE+DIS still shows a higher superiority than other algorithms, demonstrating the advantage of taking group members' disagreement into account. In addition, Fig. 12c demonstrates that the number of task assignments of SIP+SDAE+DIS-GGTA has no advantage compared with other methods.

**Effect of  $|SI|$ .** In this set of experiments, we evaluate the scalability of all the proposed algorithms by changing the number  $|SI|$  of tasks from 1 to 5 k. As expected, although the CPU time increases as  $|SI|$  increases, SIP+SDAE+DIS-OGTA performs well in improving the assignment success rate and the number of task assignments, which is demonstrated in Fig. 13b, c. Figure 13a indicates that SIP+SDAE+DIS-GGTA is the most efficient algorithm, while other algorithms based on OGTA run much more slower, which is mainly due to the extra time cost for building the tree to be searched and searching the tree during the OGTA procedure. In terms of assignment success rate, the accuracy of SIP+SDAE+DIS-OGTA is a bit higher than SIP+SDAE+DIS-GGTA and SIP+SDAE-OGTA, while OGTA still keeps almost constant as  $|SI|$  grows, which is shown in Fig. 13b. Similar to the previous results, the OGTA-related algorithms (i.e., SIP+SDAE+DIS-OGTA, SIP+SDAE-OGTA, OGTA) outperform the SIP+SDAE+DIS-GGTA method for all the values of  $|SI|$  in the number of task assignments, which is depicted in Fig. 13c.

**Effect of  $|W|$ .** In our final set of experiments, we investigate how the number of workers affects the efficiency and effectiveness of group task assignment. As Fig. 14a shows, the CPU time increases with a larger number of workers with the similar reason of the effect of  $|SI|$ , i.e., the extra

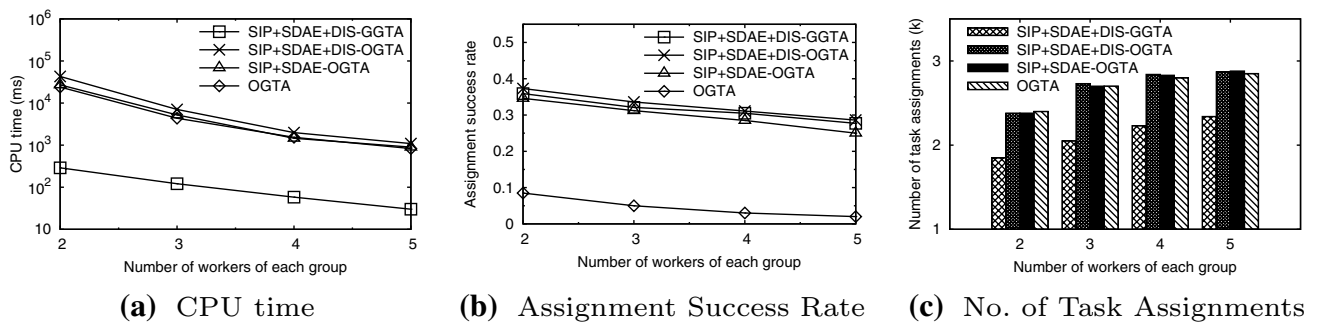


Fig. 12 Performance of group task assignment: effect of  $numW$

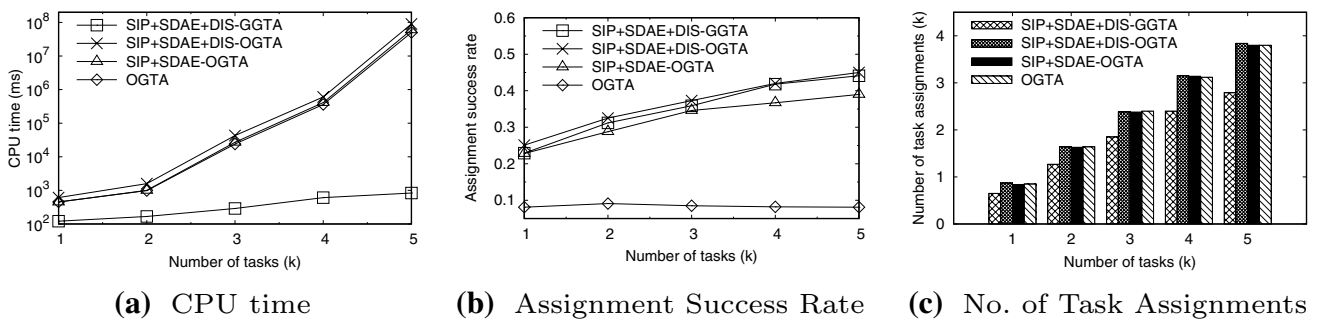


Fig. 13 Performance of group task assignment: effect of  $|SI|$

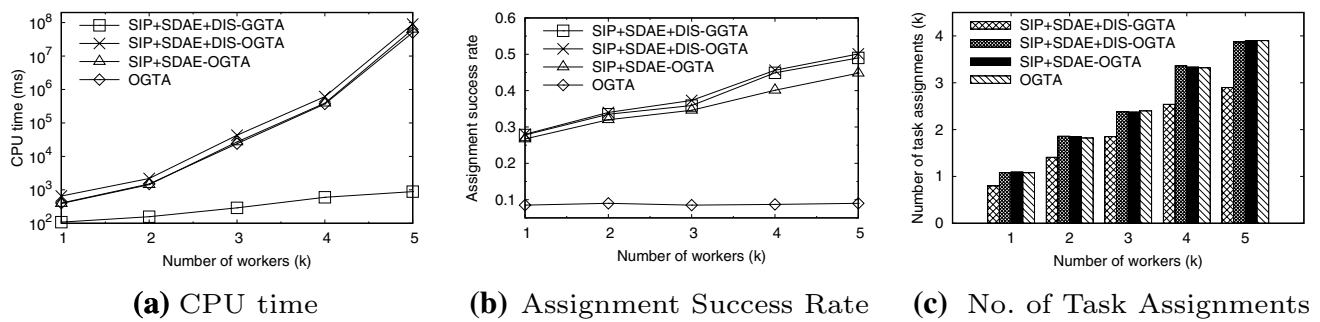


Fig. 14 Performance of group task assignment: effect of IW

time cost for building the tree and searching the tree during the OGTA procedure with more workers. The optimal group task assignment algorithm considering both worker group's preference and group members' disagreement (SIP+SDAE+DIS-OGTA) performs best in assignment success rate, which is shown in Fig. 14b. In Fig. 14c, the number of task assignments increases with IW getting larger due to the fact that more spatial tasks can be conducted by more workers. Although SIP+SDAE+DIS-OGTA and SIP+SDAE+DIS-GGTA show a similar performance in ASR, the SIP+SDAE+DIS-OGTA performs better in number of task assignments, which demonstrates the effectiveness of optimal task assignment method.

## 6 Related Work

Spatial crowdsourcing (SC) is a new concept of online crowdsourcing, which employs smart device carriers as workers to physically travel to specified locations and perform the requested spatial tasks with various constraints [10, 27, 29, 33, 34]. Most existing research focuses on the task assignment [24–26]. Kazemi et al. [15] classify SC into two categories, namely server assigned tasks (SAT) and worker selected tasks (WST) based on the task publishing modes. In particular, for the SAT mode which is popular in existing researches, SC server is responsible for directly assigning proper tasks to nearby workers, which aims to maximize the number of assigned tasks after collecting all the locations of workers/tasks on the server side [8, 15, 16, 21] or maximize the reliability-and-diversity score of assignments [9], or maximize the number of accomplished tasks for a worker with an optimal schedule on the client side [12], or maximize the coverage of required skills of workers [7]. For the WST mode, spatial tasks are published online and then broadcast to all workers, such that workers can choose any task according to their personal preferences by themselves [11].

Meanwhile, quality assurance is an intractable problem needing to be solved during the process of spatial

task assignment. Workers are more likely to honestly and promptly complete the assigned tasks if the quality control strategy is considered, e.g., giving higher priority to workers who are more interested in tasks. Although a few existing approaches consider workers' preferences for tasks in crowdsourcing [1, 5], they just infer workers' preferences from historical task-performing records or explicit feedbacks without taking workers' social impact into consideration.

Most of the previous studies in spatial crowdsourcing mainly focus on assigning tasks to the individual worker, which unfortunately cannot be effectively applied for group task assignment. In recent years, a few researches [6, 13] are studied for group task assignment (also called collaborative task assignment), i.e., assigning tasks to a group of multiple workers. The groups are formed by workers in an ad hoc way, also called occasional groups, who have a shared purpose only in a certain time. Cheng et al. [6] propose a framework called cooperation-aware spatial crowdsourcing (CA-SC), in which they design both task-priority greedy approach and game-theoretic approach to solve the CA-SC problem, aiming to achieve high cooperation quality scores. Different from the algorithm proposed by Cheng et al. [6], our proposed algorithm aims to maximize the total number of task assignments and gives higher priorities to worker groups with higher consensus scores (including groups' preference based on social impact and group members' disagreement) on tasks.

## 7 Conclusion

In this paper, we propose a novel task assignment problem, called group task assignment (GTA), in spatial crowdsourcing. In order to achieve effective task assignment, we address a few challenges by proposing different strategies to obtain the social impact-based preferences of different worker groups for each task category and adopting an optimal algorithm to assign tasks. Moreover, we further optimize the original solution by proposing several strategies to improve the effectiveness of group task assignment, wherein a deep

learning method is adopted to better learn workers' social impact-based preferences and the group consensus is taken into consideration. Extensive empirical study demonstrates the effectiveness of our proposed solution.

**Acknowledgements** This work is partially supported by Natural Science Foundation of China (No. 61972069, 61532018, 61836007 and 61832017).

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Ambati V, Vogel S, Carbonell JG (2011) Towards task recommendation in micro-task markets. In: AAAI, pp 80–83
2. Bahdanau D, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. In: ICLR
3. Bengio Y (2009) Learning deep architectures for AI. *Found Trends Mach Learn* 2(1):1–127
4. Bottou L (2010) Large-scale machine learning with stochastic gradient descent. In: *Proceedings of COMPSTAT*, pp 177–186
5. Buchholz S, Latorre J (2011) Crowdsourcing preference tests, and how to detect cheating. In: ISCA, pp 3053–3056
6. Cheng P, Chen L, Ye J (2019) Cooperation-aware task assignment in spatial crowdsourcing. In: ICDE, pp 1442–1453
7. Cheng P, Lian X, Chen L, Han J, Zhao J (2016) Task assignment on multi-skill oriented spatial crowdsourcing. *TKDE* 28(8):2201–2215
8. Cheng P, Lian X, Chen L, Shahabi C (2017) Prediction-based task assignment in spatial crowdsourcing. In: ICDE, pp 997–1008
9. Cheng P, Lian X, Chen Z, Fu R, Chen L, Han J, Zhao J (2015) Reliable diversity-based spatial crowdsourcing by moving workers. *PVLDB* 8(10):1022–1033
10. Cui Y, Deng L, Zhao Y, Yao B, Zheng VW, Zheng K (2019) Hidden poi ranking with spatial crowdsourcing. In: SIGKDD, pp 814–824
11. Deng D, Shahabi C, Demiryurek U (2013) Maximizing the number of workers' self-selected tasks in spatial crowdsourcing. In: SIGSPATIAL, pp 314–323
12. Deng D, Shahabi C, Zhu L (2015) Task matching and scheduling for multiple workers in spatial crowdsourcing. In: SIGSPATIAL, p 21
13. Gao D, Tong Y, Ji Y, Ke X (2017) Team-oriented task planning in spatial crowdsourcing. In: APWeb, pp 41–56
14. Gao D, Tong Y, She J, Song T, Chen L, Xu K (2017) Top-k team recommendation and its variants in spatial crowdsourcing. *DSE* 2(2):136–150
15. Kazemi L, Shahabi C (2012) Geocrowd: enabling query answering with spatial crowdsourcing. In: SIGSPATIAL, pp 189–198
16. Kazemi L, Shahabi C, Chen L (2013) Geotrucrowd: trustworthy query answering with spatial crowdsourcing. In: SIGSPATIAL, pp 304–313
17. Li X, Zhao Y, Guo J, Zheng K (2020) Group task assignment with social impact-based preference in spatial crowdsourcing. In: DASFAA
18. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: NIPS, pp 3111–3119
19. Roy SB, Amer-Yahia S, Chawla A, Das G, Cong Y (2010) Space efficiency in group recommendation. *VLDB* 19(6):877–900
20. Song T, Tong Y, Wang L, She J, Yao B, Chen L, Xu K (2017) Trichromatic online matching in real-time spatial crowdsourcing. In: ICDE, pp 1009–1020
21. Tong Y, Chen L, Zhou Z, Jagadish HV, Shou L, Weifeng L (2019) Slade: a smart large-scale task decomposer in crowdsourcing. In: ICDE, pp 2133–2134
22. Tong Y, She J, Ding B, Chen L, Wo T, Xu K (2016) Online minimum matching in real-time spatial data: experiments and analysis. *VLDB* 9(12):1053–1064
23. Tong Y, She J, Ding B, Wang L (2016) Online mobile micro-task allocation in spatial crowdsourcing. In: ICDE, pp 49–60
24. Tong Y, Wang L, Zhou Z, Chen L, Du B, Ye J (2018) Dynamic pricing in spatial crowdsourcing: a matching-based approach. In: SIGMOD, pp 773–788
25. Tong Y, Wang L, Zhou Z, Ding B, Chen L, Ye J, Xu K (2017) Flexible online task assignment in real-time spatial data. *VLDB* 10(11):1334–1345
26. Tong Y, Zeng Y, Zhou Z, Chen L, Ye J, Xu K (2018) A unified approach to route planning for shared mobility. *PVLDB* 11(11):1633–1646
27. Tong Y, Zhou Z, Zeng Y, Chen L, Shahabi C (2020) Spatial crowdsourcing: a survey. *VLDB* 29(1):217–250
28. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA (2010) Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J Mach Learn Res* 11:3371–3408
29. Xia J, Zhao Y, Liu G, Xu J, Zhang M, Zheng K (2019) Profit-driven task assignment in spatial crowdsourcing. In: IJCAI, pp 1914–1920
30. Yin H, Wang Q, Zheng K, Li Z, Yang J, Zhou X (2019) Social influence-based group representation learning for group recommendation. In: ICDE, pp 566–577
31. Yin H, Zou L, Nguyen QVH, Huang z, Zhou X (2018) Joint event-partner recommendation in event-based social networks. In: ICDE, pp 929–940
32. Zhao Y, Li Y, Wang Y, Su H, Zheng K (2017) Destination-aware task assignment in spatial crowdsourcing. In: CIKM, pp 297–306
33. Zhao Y, Xia J, Liu G, Han S, Lian D, Shang S, Zheng K (2019) Preference-aware task assignment in spatial crowdsourcing. In: AAAI, pp 2629–2636
34. Zhao Y, Zheng K, Cui Yue, Su H, Zhu F, Zhou X (2019) Predictive task assignment in spatial crowdsourcing: a data-driven approach
35. Zhao Y, Zheng K, Li Y, Su H, Liu J, Zhou X (2019) Destination-aware task assignment in spatial crowdsourcing: a worker decomposition approach. In: TKDE