Preference-aware Group Task Assignment in Spatial Crowdsourcing: A Mutual Information-based Approach

Yunchuan Li
University of Electronic Science and Technology of China
Chengdu, China
liyunchuan@std.uestc.edu.cn

Yan Zhao*
Department of Computer Science
Aalborg University
Aalborg, Denmark
yanz@cs.aau.dk

Kai Zheng
University of Electronic Science and Technology of China
Chengdu, China
zhengkai@uestc.edu.cn

***Abstract*—With the popularity of GPS-enable smart devices and the development of wireless network, Spatial Crowdsourcing (SC), as a framework for assigning location-sensitive tasks to moving workers, has received wide attention in recent years. In real-world scenarios, some complex tasks exist that may not be completed by a single worker. In this case, the tasks are often assigned to multiple workers, which is called group task assignment. However, the assignment of tasks that satisfy all group members in an even way remains a challenge. To this end, we propose a novel preference-aware group task assignment framework that includes two components: Mutual Information-based Preference Modeling (MIPM) and Preference-aware Group Task Assignment (PGTA). Specifically, MIPM learns the preferences of worker groups by maximizing the mutual information among workers based on the worker-task interaction data and the group-task interaction data, where an attention mechanism is used. PGTA adopts an optimal task assignment algorithm based on tree decomposition to assign tasks to appropriate worker groups, which aims to maximize the overall number of assigned tasks while giving priority to the groups of workers that are more interested in the tasks. Finally, extensive experiments are conducted, verifying the effectiveness and practicality of the proposed solutions.**

***Index Terms*—preference, group task assignment, mutual information, spatial crowdsourcing**

I. Introduction

With the continuous development of GPS-equipped smart devices and wireless networks, Spatial Crowdsourcing (SC), a recently proposed concept and framework, has attracted great attention from both academia and industry communities. SC platforms recruit a group of available workers to actually go to a specific location to complete tasks, e.g., taking photos, monitoring traffic conditions, etc. The way to allocate these spatial tasks to workers is called task assignment.

Most of the existing studies focus on individual task assignment [1]–[5], which means that a task is assigned to an individual worker. For example, Zheng et al. [6] take workers' rejection into consideration and design algorithms to maximize workers' acceptance in order to improve the system feasibility. The study [7] proposes a Flexible Two-sided Online task Assignment (FTOA) problem, which aims to guide idle workers to perform tasks by predicting the spatiotemporal distributions of subsequent tasks and workers, thereby increasing the total number of assigned worker-task pairs. However, SC applications exist in which a single worker cannot efficiently and independently perform a task, e.g., home improvement, major furniture installation, monitoring the traffic condition in an area, or holding a barbecue party [8]–[10], In such scenarios, each task needs to be assigned to multiple workers, which is called group task assignment.

Several group task assignment approaches have been developed in SC. For example, Gao et al. [11] propose a Top-k team recommendation problem in SC, called (TopkTR), and also propose its variant (i.e., TopkTRL), which aim to recommend suitable crowdsourced teams for tasks. Considering the cooperation of workers, Cheng et al. [8] propose a greedy approach and a game-theoretic approach such that tasks can be completed by multiple workers with high cooperation qualities. However, the above studies are based on the assumption that all workers in a worker group are willing to complete the assigned tasks. However, in actual situations, some workers may not be interested in the assigned tasks, which leads to workers rejecting tasks or completing tasks with low quality. Recently, Li et al. [12] propose a framework for group task assignment based on worker groups' preferences by taking workers' social impact into account. In this work we will go further in this direction and use the mutual information among workers to learn the group preference, and then optimize the global task assignment based on group preference.

Next, we will use a motivation example to illustrate the problem of group task assignment. In Figure 1, there are five workers $\{w_1, ..., w_5\}$, and two tasks $\{s_1, s_2\}$. Each worker has a current location and a reachable distance. Each task that requires two workers to complete is associated with a location in which it will be performed. Considering the spatio-temporal constraints between workers and tasks (i.e., an assigned task should be located in the reachable range of the corresponding worker, and the worker should arrive at the location of the

* Corresponding author: Yan Zhao.

assigned task before its deadline), s_1 and s_2 have three available worker groups, i.e., $\{\{w_1, w_2\}, \{w_2, w_5\}, \{w_1, w_5\}\}$ and $\{\{w_3, w_5\}, \{w_3, w_4\}, \{w_4, w_5\}\}$, respectively. Figure 1 also shows the preference values of the two tasks for their available worker groups. Without considering the group preference, a general group task assignment problem is to maximize the total number of assigned tasks, and we can get a task assignment $\{(s_1, \{w_1, w_2\}), (s_2, \{w_3, w_5\})\}$ that achieves the optimal task assignment. However, the total preference value of the task assignment is only 0.11, and the two assigned worker groups have extremely low preferences for tasks s_1 and s_2. Under such an assignment, workers may refuse to accept tasks or complete tasks with low quality.

To tackle this issue, we propose a data-driven framework that assigns tasks by considering the preference of worker groups. The framework consists of a Mutual Information-based Preference Modeling (MIPM) phase and a Preference-aware Group Task Assignment (PGTA) phase. The first phase aims to model the preferences of each worker group for different task categories. More specifically, we first maximize the mutual information between worker representations and group representations to train a discriminator, which aims to adjust preference representation vectors of workers and worker groups. Then, we use an attention mechanism to obtain highly relevant worker members in worker groups, and set different weights for each worker, so as to improve the preference representation vector of each worker group. Finally, group preferences for different task categories are obtained through a prediction layer. In the assignment phase, we first obtain available worker groups of each task without violating the spatio-temporal constraints. Then we use a tree-decomposition algorithm to assign a suitable worker group to each task while giving higher priorities to the worker groups that are more interested in the tasks. When applying our solutions in the example in Figure 1, a task assignment $\{(s_1, \{w_1, w_5\}), (s_2, \{w_3, w_4\})\}$ with a higher total preference value of 0.91 is obtained.

The contributions of this paper can be summarized as follows:

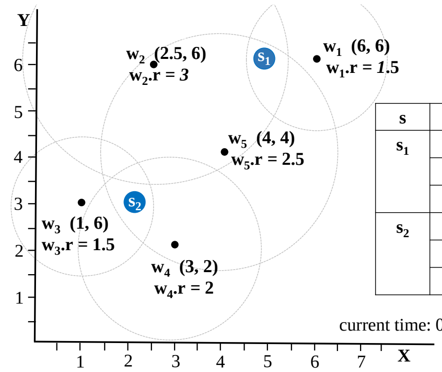
1) To the best of our knowledge, this is the first work in SC that considers the mutual information-based preference among workers.

2) We give a data-driven solution to learn workers' group preferences by maximizing the mutual information among workers based on an attention mechanism.

3) We perform the task assignment based on group preferences by adopting a tree-decomposition algorithm.

4) Extensive experiments are conducted with a real-world dataset, where the results confirm the effectiveness of our solutions. Our proposed model can not only assign more worker groups to complete tasks, but also has a high completion rate for the assigned tasks.

The remainder of this paper is organized as follows. Section II introduces the related work and Section III provides notations and the proposed problem. In Section IV-B , we propose a mutual information-based approach for worker group preference modeling. The preference-based group task assignment algorithms are then presented in Section IV-C, followed by the experimental results in Section V. Finally, we conclude the paper in Section VI.



s	g		preference
s_1	g_1	$\{w_1, w_2\}$	0.03
	g_2	$\{w_2, w_5\}$	0.17
	g_3	$\{w_1, w_5\}$	0.55
s_2	g_1	$\{w_3, w_5\}$	0.08
	g_2	$\{w_3, w_4\}$	0.36
	g_3	$\{w_4, w_5\}$	0.18

Fig. 1. Running example

II. Related Work

Spatial crowdsourcing (SC) is a new framework that has emerged in recent years, requiring workers with GPS devices to physically travel to a specific location under certain restrictions to perform spatial tasks [2], [13]–[20]. Most of the existing studies focus on task assignment. Based on the task publishing models, Kazemi et al. [21] divides SC into worker selection task (WST) mode and server assignment task (SAT) mode. Most of the research in the two modes is devoted to finding ways to achieve a certain goal of task assignment, e.g., maximizing the number of assigned tasks [7], [21]–[24], maximizing the coverage of worker skills required by tasks [9], or maximizing the total profit of the platform [15], [25]. However, many studies on task assignment in SC put their focus on allocating individual workers to tasks without considering cooperation among workers. For a complex task, such as e.g., home improvement, major furniture installation, and monitoring the traffic condition in an area, a single worker may not be able to complete it independently.

In the task assignment problem of SC, compared with the single task assignment, group task assignment is more complex, and there are few researches in this field. The recent study [11] proposes a Top-k team recommendation problem in SC, where a method to recommend suitable crowdsourcing teams for each task. Cheng et al. [8] consider that the completion of complex tasks requires the cooperation of workers, and propose a greedy method and a game-theoretic method to assign multiple workers with high cooperative qualities to complete a task together. However, most previous studies on group task assignment did not consider whether the workers are interested in the task, which may lead to the workers' rejection of performing the task or low-quality completion of the task.

Recently, Li et al. [26] use a bipartite graph embedding model and the attention mechanism to learn the social impact-based preferences of the worker groups for different categories of tasks, and assign tasks to the groups according to their preferences. In the study [26], social networks between workers are used to reduce the sparseness of group-task interaction data. However, it is often difficult for us to obtain additional information such as social networks among workers. Therefore, the method of improving sparsity through additional information is no longer applicable. Unlike the above studies, our proposed approach does not dependent on the additional information, which uses the historical task-performing data to maximize the mutual information among workers in order to learn the informative representation vectors of groups and further learn the group preferences.

III. Problem Definition

In this section, we will briefly introduce a set of preliminary concepts and then give our problem statement. Table I summarizes the main symbols used in the paper.

TABLE I
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
$w.l$	Current 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

Definition 1 (Spatial Task): A spatial task, denoted by $s = (l, p, e, c, numW)$, has a location $s.l$, a publication time $s.p$, an expiration time $s.e$, a category $s.c$, and a number $s.numW$ that is the number of workers required to be assigned to perform s.

Definition 2 (Worker): A worker, denoted as $w = (l, r, on, off)$, includes a location $w.l$, a reachable distance $w.r$, an online time $w.on$, and an offline time $w.off$. The reachable area of the worker is a circular area with $w.l$ as the center and $w.r$ as the radius, where worker w can accept the task assignment. A worker can be in an online or offline mode. When a worker is ready for a task (from the online time $w.on$ to the next offline time $w.off$), the worker is online. In addition, a worker also has an offline time $w.off$, after which the worker cannot accept the task.

In group task assignment scenarios, a task s requires multiple workers to complete it cooperatively, which is also consistent with the actual reality. Workers can only handle one task at a specific time, which is reasonable in actual situations.

Definition 3 (Available Worker Set): The available worker set for a task s, denoted as $AWS(s)$, is a set of workers that satisfy the following conditions: $\forall w \in AWS(s)$:

1) worker w is in an online mode, i.e., $w.on \leq t_{now} \leq w.off$, and

2) task s is located in the reachable range of worker w, i.e., $d(w.l, s.l) \leq w.r$, and

3) worker w can travel from the origin to the location of task s directly before it expires, i.e., $t_{now} + t(w.l, s.l) \leq s.e$, and

4) worker w can travel from the origin to the location of task s directly before the offline time of w, i.e., $t_{now} + t(w.l, s.l) \leq w.off$,

where t_{now} is the current time, $d(w.l, s.l)$ is the travel distance between location $w.l$ and location $s.l$, and $t(w.l, s.l)$ is the travel time between location $w.l$ and location $s.l$.

For the sake of simplicity, we assume all workers share same speed,i.e., $t(w.l, s.l) = d(w.l, s.l)$. Because the time the worker arrives at the task is not directly related to whether the speed is uniform, the algorithm we proposed can also handle the situation where workers move at different speeds.

In Figure 1, task s_1 is located in the reachable ranges of the online workers w_1, w_2, and w_5. Further, w_1, w_2, and w_5 can arrive at the location of task s_1 before the expiration time of s and their offline time. Therefore, we can get an available worker set $\{w_1, w_2, w_5\}$ for task s_1, i.e., $AWS(s_1) = \{w_1, w_2, w_5\}$.

Definition 4 (Available Worker Group): Given a task s 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) all the workers in $AWG(s)$ are available workers for task s, i.e., $AWG(s) \subset AWS(s)$, and

2) the number of the workers in $AWG(s)$ is $s.numW$, i.e., $|AWG(s)| = s.numW$, and

3) each worker in $AWG(s)$ can arrive at the location of s before the offtimes of others in $AWG(s)$, i.e., $\forall w_i, w_j \in AWG(s), t_{now} + t(w_i.l, s.l) \leq w_j.off$.

For task s_1 in Figure 1, its available worker set is $\{w_1, w_2, w_5\}$. Assuming that $s_1.numW = 2$, we can obtain three available worker groups for s_1, i.e., $\{w_1, w_2\}$, $\{w_1, w_5\}$, and $\{w_2, w_5\}$. In the rest of this paper, we will use the terms *worker group* and *group* interchangeably.

Definition 5 (Spatial Task Assignment): Given a set of workers W and a set of tasks S, a spatial task assignment is denoted as A, which contains a set of pairs of a task and an AWG for the task: $(s_1, AWG(s_1))$, $(s_2, AWG(s_2))$,..., $(s_{|S|}, AWG(s_{|S|}))$, where $AWG(s_i) \cap AWG(s_j) = \emptyset$, and $1 \leq i \neq j \leq |S|$.

For example, $\{(s_1, \{w_1, w_2\}), (s_2, \{w_3, w_4\})\}$ and $\{(s_1, \{w_1, w_5\}), (s_2, \{w_3, w_4\})\}$ are two spatial task assignments in Figure 1.

Preference-aware Group Task Assignment (PGTA) Problem Statement. Given a set of workers W and a set of tasks S at the current time on a SC platform, our problem aims to find an optimal task assignment A_o that maximizes the number of assigned tasks (i.e., $\forall A_i \in \mathrm{A}$ ($|A_i.S| \leq |A_o.S|$)) while taking the preferences of worker groups into account, where $A_i.S$ denotes the set of tasks that are assigned to all the workers for A_i, and A denotes all the possible ways of assignments.

IV. Framework

In this section, we will explain the details of our proposed group task assignment framework. In reality, the groups in group task assignment are often contingent, so the group-task interaction data is often sparse. To overcome this problem, we use mutual information maximization to capture the internal information of workers and groups. At the same time, because different workers play different roles in different groups, their influence is also different. Therefore, we use an attention mechanism to learn the weight of each worker in a group. Finally, we learn the preferences of the worker groups for tasks, and adopt a tree-decomposition algorithm to obtain the optimal solution.

A. Framework Overview

As shown in Figure 2, the group task assignment framework is mainly composed of two parts: 1) Mutual Information-based Preference Modeling (MIPM) for worker groups, 2) Preference-based Group Task Assignment (PGTA).

In the MIPM part, we use a method of maximizing mutual information and an attention mechanism on worker-task interaction data and group-task interaction data to model worker groups' preferences. Considering that the preferences of workers and groups are closely related, for each group, we contrast the representations of the group members against those of non-members (that do not belong to the group) with similar task-performing history, and maximize the worker-group mutual information to train a discriminator to regularize representation vectors of workers and groups. In order to overcome the sparsity of the group-task interaction data, we propose a group-adaptive preference weighting technique. According to the discriminator, we obtain highly relevant worker members in each group and set different weights for each worker, so as to use the workers' personal preferences to improve group preference representation vectors. Finally, the group preferences for different task categories are obtained through a prediction layer.

In the PGTA part, given workers and tasks to be assigned, we first obtain the set of available workers $AWG(s)$ for each task s by considering the spatio-temporal restrictions (i.e., the reachable radius of the workers, the available time of the workers, and the expiration time of the tasks). Then we employ the optimal task assignment algorithm based on tree decomposition to assign tasks to the appropriate worker groups to maximize the total number of assigned tasks while giving higher priorities to worker groups with higher preferences on tasks.

B. Mutual Information-based Preference Modeling

In this section, we will introduce how to use historical interaction data to model worker groups' preferences. The main problem facing group preference modeling is the sparsity of the group-task interaction data. We use mutual information maximization for contrastive representation learning and utilize a group-adaptive preference weighting technique to relief the data sparsity. By aggregating the representation vectors of each group member, we can obtain the representation vectors of the worker groups. Then the representation vectors of the worker groups can be used to calculate the preferences of each worker group for all task categories, which will be used in the task assignment phase.

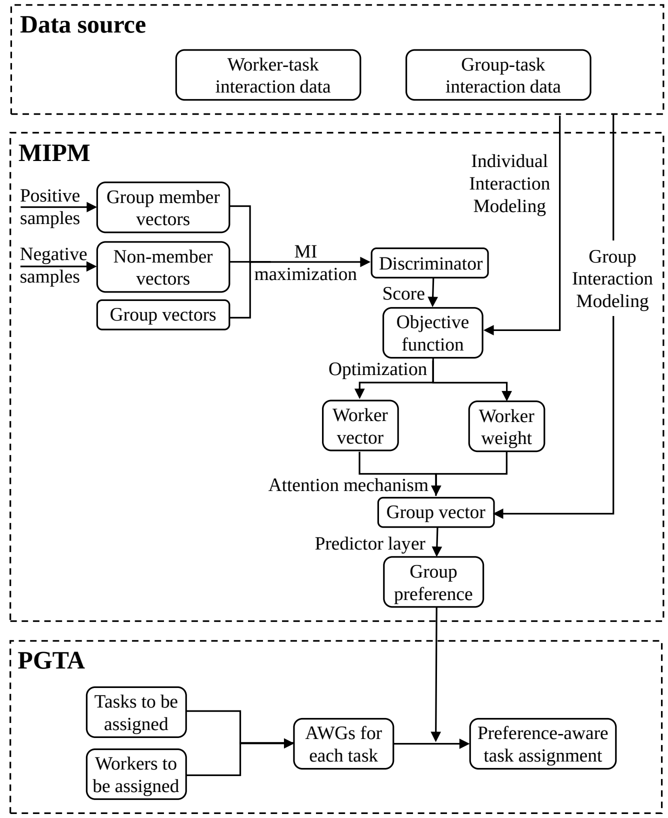


Fig. 2. Framework Overview

1) Worker and Group Representation Learning: We use W to represent the set of workers, C to represent the set of task categories, and G to represent the set of worker groups. Further, X_{WC} represents the interaction matrix formed by the interaction data of the worker-task category, and X_{GC} represents the interaction matrix formed by the interaction data of the group-task category. We use $\mathbf{w}_i$ and $\mathbf{c}_j$ to denote the latent representation vectors of worker w_i and task category c_j, respectively, where the worker representation vector denotes the worker's personal preference (stored in the worker-task category interaction matrix $\mathbf{X}_{WC}$).

Worker Representation Learning. In order to obtain a latent representation vector that can represent each worker's personal preference, we use a multi-layer perceptron with two fully connected layers, shown as follows:

$$\mathbf{w} = f_e(w, X_w) = \sigma(K_2^T(\sigma(K_1^T x_w + b_1)) + b_2) \quad \forall w \in W \tag{1}$$

where $\mathbf{w}$ denotes the worker' preference representation vector of worker w, f_e is a preference encoding function, X_w denotes the worker-task category interaction matrix, and x_w is is the row corresponding to worker w in the matrix. We use a nonlinear activation function $\sigma(x) == -\frac{1}{1+exp(-x)}$ to encode the preference, where K_1^T and K_2^T are two learnable weight matrices, and b_1 and b_2 are bias matrices.

Group Representation Learning. Since the preference of the worker group depends on the preference of each group member, in order to better obtain the representation vector of the worker group, we use an attention mechanism [27]. The attention mechanism can learn the contribution of different group members to the group decision, as a result of which each group member's preference representation vector can be weighted. The equation is as follows:

$$\mathbf{g_i} = \sum_{w_j \in g_i} \alpha(j,i) K_{agg} \mathbf{w_j} \tag{2}$$

$$\alpha(j,i) = -\frac{exp(h^T K_{agg} \mathbf{w_j})}{\sum_{w_k \in g_i} exp(h^T K_{agg} \mathbf{w_k})} \tag{3}$$

where $\mathbf{g_i}$ and $\mathbf{w_i}$ denote the representation vectors of the worker group g_i and worker w_j, respectively. Next, $\alpha(j,i)$ is a learnable parameter representing the weight of worker w_j in the worker group g_i, which can be calculated in Equation 3, where h^T represents the hidden layer parameters of the attention network. A larger $\alpha(j,i)$ means that worker w_j can contribute more to the group decision.

2) Contrastive Representation Learning: We notice that group activities often reflect the following two phenomena: (1) the difference between different groups; (2) the connection between members in a group. To capture these features, we contrast the representation vectors of the members in a group against those of non-members (that do not belong to the group) with similar task-performing history, so as to effectively adjust the latent representation vectors of workers and groups. Moreover, inspired by the success of the Mutual Information (MI) for measuring the dependence between variables [28], we utilize MI for the contrastive representation learning. With MI, we can learn useful information only from the data itself without using additional data, which can reduce data sparsity to some extent.

The MINE framework [29] verifies the flexibility and effectiveness of maximizing MI methods using discriminator networks (i.e., classifiers), which can accurately distinguish between positive samples taken from a joint distribution and negative samples taken from a marg inal distribution. Through such a contrast method, using a scoring function can increase the score of positive cases and reduce the score of negative cases.

More specifically, we maximize the mutual information between workers and groups, i.e., the mutual information between the representation vector of the group members (obtained by Equation 1) and the representation vector of the group (obtained by Equation 2), by training a contrast discriminator network $\mathcal{D} : \mathbb{R}^F \times \mathbb{R}^F \mapsto \mathbb{R}^+$, where $\mathcal{D}(\mathbf{w}, \mathbf{g})$ represents a score function of the worker-group pair. For the members in group g, the higher the corresponding score.

Similar to the previous work [30], we use a simple bilinear function to calculate the score of the worker-group representation pair, i.e., $\mathcal{D}(\mathbf{w}, \mathbf{g}) = \sigma(\mathbf{w}^T W \mathbf{g})$, where $\sigma(\cdot)$ is a non-linear function and W is a learnable score matrix. The positive sample $(\mathbf{w}, \mathbf{g})$ in the network $\mathcal{D}$ represents the latent preference representation vector of the worker-group pair (w, g), where $w \in g$. The negative example is denoted as $(\tilde{\mathbf{w}}, \mathbf{g})$, where g represents the representation vector of the worker group, and $\tilde{\mathbf{w}}$ is the representation vector of the non-members that do not belong to the group, where the non-members are sampled from the negative sampling distribution $P_n(\tilde{w}|\mathbf{g})$. We train the discriminator $\mathcal{D}$ using a noise-contrast target, and calculate a binary cross-entropy (BCE) loss function between the positive and negative samples, where the positive and negative samples are sampled in the joint probability distribution and marginal probability distribution respectively. The optimization goal are shown in the following.

$$\begin{aligned} O_{MI} = -\frac{1}{|G|} \sum_{g \in G} \frac{1}{\mu_g} \big[\sum_{w \in g} log\mathcal{D}(\mathbf{w}, \mathbf{g}) \\ + \sum_{i=1}^{M_g} \mathbb{E}_{\tilde{\mathbf{w}} \sim P_n} log(1 - \mathcal{D}(\tilde{\mathbf{w}}, \mathbf{g})) \big] \end{aligned} \tag{4}$$

where G denote all the worker groups, and μ_g represents the sum of the number of workers in group g and the number of workers negatively sampled from g, i.e., $\mu_g = |g| + |M_g|$. Next, $\mathcal{D}(\mathbf{w}, \mathbf{g})$ denotes the score between worker w and group g, and $\mathbb{E}_{\tilde{\mathbf{w}} \sim P_n} log(1 - \mathcal{D}(\tilde{\mathbf{w}}, \mathbf{g}))$ is the mathematical expectation of $log(1 - \mathcal{D}(\tilde{\mathbf{w}}, \mathbf{g}))$. The objective function is based on the Jensen-Shannon (JS) divergence between the joint distribution and the marginal distribution, thereby effectively maximizing the mutual information between the worker representation vector $\mathbf{w}$ and the worker group representation vector $\mathbf{g}$ [30]. We do not use the random sampling in the negative sampling of workers, but prioritize the sampling of non-members who have performed the tasks with category x_g (x_g represents the row corresponding to worker group g in the group-task interaction data X_{GC}). When training the discriminator, by contrasting group members against non-members with similar task execution histories, we can learn the discriminative characteristics shared by group members, thereby improving the representation vectors of workers and groups. We define the negative sampling distribution as follows:

$$P_n(\tilde{\mathbf{w}}|\mathbf{g}) \propto \eta I(X_{\tilde{w}}^T \cdot x_g > 0) + (1 - \eta) \frac{1}{|W|} \tag{5}$$

where I is an indicator function, and η controls the sampling ratio.

3) Group-adaptive Preference Weighting: In this section, we describe the group adaptive preference weighting strategy, which overcomes the sparsity of interactive data by giving the relevant group members higher priority. First, we define the loss functions of workers and groups. In the group-task category interaction data X_{GC}, we use a polynomial likelihood equation to optimize the group loss function to obtain the group representation vector $\mathbf{g}$. The group representation vector $\mathbf{g}$ is used as the input of a fully connected layer, and then a softmax function is used to regularize the output of the fully connected layer and generates the probability vector $\pi(\mathbf{g})$ for the task category C. The loss function measures the KL distance between the regularized task-performing history and the predicted probability that the task assigned to the corresponding worker group, so the objective function of the group is as follows:

$$O_{group} = -\sum_{g \in G} \frac{1}{|x_g|} \sum_{i \in C} x_{gi} log \pi_i(\mathbf{g}) \tag{6}$$

where $\pi(\mathbf{g}) = softmax(K_C \mathbf{g})$, and K_C represents the weight matrix of the prediction layer. Similarly, based on the polynomial likelihood equation, using the worker-task interaction data X_{WC} to adjust the worker representation vector w, the worker's objective function is calculated as follows:

$$O_{worker} = -\sum_{w \in W} \frac{1}{|x_w|} \sum_{i \in C} x_{wi} log \pi_i(\mathbf{w}) \tag{7}$$

Equation 7 is to predict the preferences of all the groups that worker w belongs to based on w's representation vector $\mathbf{w}$, which will lead to the overfitting problem since the group-task interaction data is very sparse, and lack of flexibility in applying preferences among different groups of workers. In order to overcome this problem, we identify the group members who are highly relevant to the group based on contextual information, and then use the personal preferences of the group members to improve the representation vector of the group. In order to measure the contextual relevance, we introduce the group-adaptive preference weighting for each group member. Specifically, in the previous section, after maximizing the mutual information, the discriminator $\mathcal{D}$ can obtain the score, $\mathcal{D}(\mathbf{w}, \mathbf{g})$, which can quantify the contextual information of each worker-group pair (w, g). It means that for workers with more information will have higher scores. We use the discriminator score as the group adaptive preference weight of group member $w \in g$. Then for each group member, we use the weight $\mathcal{D}(\mathbf{w}, \mathbf{g})$ to adjust the worker group representation vector g. Equation 8 shows the objective function of the group-adaptive preference weighting technique.

$$O_{WG} = -\sum_{g \in G} \frac{1}{|x_g|} \sum_{i \in C} \sum_{w \in g} \mathcal{D}(\mathbf{w}, \mathbf{g}) x_{wi} log \pi_i(\mathbf{g}) \tag{8}$$

Then the objective function of the whole MIPM model is the sum of the three objective functions including the MI maximization objective function (cf. Equation 5), the worker group objective function (cf. Equation 6), and the group adaptive preference weight user objective function (cf. Equation 8), shown in Equation 9. We use a standard stochastic gradient descent(SGD) strategy to minimize the objective function [31].

$$O = O_{MI} + O_{group} + \lambda O_{WG} \tag{9}$$

By optimizing the above objective function, we can obtain the group representation vectors, and then obtain the group preferences for different task categories through the prediction layer, which will be introduced in Section IV-C1.

C. Preference-based Group Task Assignment

In this section, we first obtain the available worker group sets for each task, and then we calculate the group preferences for different task categories, based on which we adopt a tree-decomposition algorithm [5], [32] to assign tasks to appropriate worker groups.

1) Finding Available Worker Group Sets and Calculating Group Preferences: According to Definitions 3 and 4, we can obtain the available worker groups for each task s, denoted as $\mathbb{AWG}(s) = \{AWG_1(s), AWG_2(s), ..., AWG_{|\mathbb{AWG}(s)|}(s)\}$, where $|\mathbb{AWG}(s)|$ denotes the number of groups in $\mathbb{AWG}(s)$.

In the MIPM phase, for the worker group g, we can obtain its representation vector $\mathbf{g}$. Next, we input the representation vector of this worker group into a prediction layer to obtain the group preferences for different task categories. The prediction layer is shown in Equation 10.

$$\pi(\mathbf{g}) = softmax(K_C \mathbf{g}) \tag{10}$$

where K_C represents the weight matrix of the prediction layer. In the previous section, we define the worker worker objective function O_worker to regularizes the worker representations with worker-task category interactions $X_W C$, thus facilitating joint training with shared encoder f_e and predictor layers [33]. The calculated preferences of each worker group will be used in the next phase.

2) Assigning Tasks based on Group Preference: In order to assign suitable worker groups to tasks, we use an optimal assignment algorithm based on tree decomposition [5], [32]. More specifically, we first construct a task dependency graph $G(V, E)$ for all the tasks, where each vertex represents a task (i.e., $v \in V$ if and only if $s_v \in S$). If tasks s_u and s_v have the common available workers, then an edge $e(u, v)$ is added between vertices u and v. We use the Maximum Cardinality Search (MCS) algorithm to separate all tasks of the task dependency graph into clusters, each of which is a maximal clique. Then we use the recursive tree construction (RTC) algorithm to organize these clusters into a balanced tree structure, where the sibling nodes in the balanced tree do not share the common available workers. After getting the tree, we use the depth-first search method to independently solve the optimal assignment sub-problem on each sibling node to find the global optimal assignment. During the search process, we assign tasks to the available worker groups with high preferences, i.e., we choose the group with the highest preference for the current task when encountering different worker groups.

V. Experiment

A. Dataset

Our experiment uses the check-in dataset from Twitter, which provides check-in data in the United States from September 2010 to January 2011 except Hawaii and Alaska, including 62462 locations and 61412 user locations. The dataset is used widely in the experimental evaluation of SC platforms [34]–[36]. Since the dataset lacks the corresponding category information of the venue, we use FourSquare's API[1] to generate its category information. Since the dataset lacks information about the geographic location of workers and tasks, for each worker and task, we take the average value of the corresponding check-in location as its location information. For each check-in, we simulate that the user is the worker, and the venue accessed by the user is the task performed by the worker. The release time of the task is set to the earliest check-in time of the task in a day. Because the dataset does not contain explicit worker group information, we set the distance to be within a certain range (10km in the experiment), and workers visiting the same category of tasks within a certain period of time (1 hour in the experiment) are regarded as a worker group. We use the category information of the venue in 18 kinds of check-ins to simulate the category information of the task. A check-in record means that the worker has accepted and completed the task.

TABLE II
EXPERIMENT PARAMETERS

Parameter	Default value
Valid time of tasks, $e - p$	2.5 h
Available time of workers, $off - on$	3 h
Reachable radius of workers, r	10 km
Number of workers for each group, $numW$	2
Number of tasks, $\|S\|$	1000
Number of workers, $\|W\|$	3000

B. Baselines

In this experiment, we verify the efficiency and effectiveness of our model by comparing the following methods:

1) **OGTA**: Optimal Group Task Assignment based on tree decomposition without considering worker group preference.

2) **AVG-OGTA**: The OGTA algorithm with worker groups' preferences calculated by the Average Preference Calculation (AVG) method. In the AVG method, the average preference of each worker group g can be calculated by $\frac{N_g^c}{N_g}$, where N_g^c denotes the number of task categories c performed by the worker group g, and N_g denotes the number of all tasks performed by the worker group g.

3) **SIP-OGTA**: The OGTA algorithm with worker groups' social impact-based preference calculated by the SIP method. SIP is a social impact-based preference (SIP) modeling algorithm [12].

[1] https://developer.foursquare.com/

4) **MIPM-GGTA**: The Greedy Group Task Assignment (GGTA) algorithm with worker groups' preferences calculated by the Mutual Information-based Preference Modeling (MIPM).

5) **MIPM-OGTA**: The OGTA algorithm with the Mutual Information-based Preference Modeling (MIPM). That is the algorithm proposed in our paper.

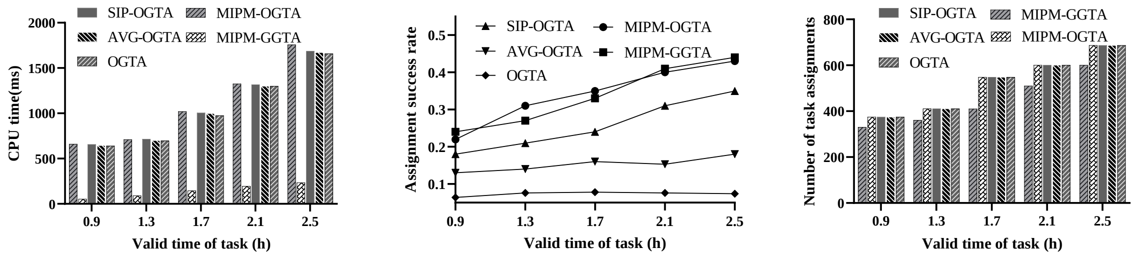
C. Experimental Setup

In order to verify the efficiency and effectiveness of baselines, three metrics including the CPU time, Assignment Success Rate (ASR), and the number of task assignments are compared among the baselines. The CPU time is the time cost of task assignment at a certain moment. ASR is the ratio of the number of successful assignments of all worker groups to the total number of assignments in a time instance. In our experiment, once a certain period of time (that is set to 1 hour in our experiments) when all members of a group actually perform (check in) tasks (locations) with the same category which are close to each other (e.g., in our experiment, the distance between tasks is required to be no more than 10 km), the assignment of this task can be considered as a successful assignment.

The default values of all parameters used in our experiments are shown in Table II. All algorithms are implemented on Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz with 32 GB RAM.

D. Experiment Results

Effect of $e - p$. First, we evaluate the effect of tasks' valid time $e-p$ on the performance of group task assignments (see Figure 3). It can be seen that the CPU time of all algorithms shows an increasing trend as the valid time of tasks increases. This is because as the valid time of tasks increases, there will be more groups of available workers, which leads to larger search space. The CPU time of OGTA-related algorithms (i.e., MIPM-OGTA, SIP-OGTA, AVG-OGTA, and OGTA) shows similar trends because these algorithms all use the optimal task assignment algorithms based on tree decomposition and have similar time complexity. As expected (see Figure 3(b)), in terms of the success rate of task assignment, preference-based task assignment algorithms (i.e., MIPM-GGTA, MIPM-OGTA, SIP-OGTA, and AVG-OGTA) all increase when the tasks' valid time gets larger. The reason behind the improvement is that as $e - p$ increases, the worker group will have more opportunities to be assigned tasks of interest, thereby improving the quality of task completion. The ASR values of MIPM-OGTA and MIPM-GGTA are higher than than those of AVG-OGTA and SIP-OGTA, which shows the advantage of mutual information-based preference modeling. As shown in Figure 3(c), the MIPM-GGTA algorithm has the least number of task assignments, while the OGTA-related algorithms (i.e., MIPM-OGTA, SIP-OGTA, AVG-OGTA, and OGTA) can generate more task assignments, which shows the superiority of the optimal task assignment algorithm.

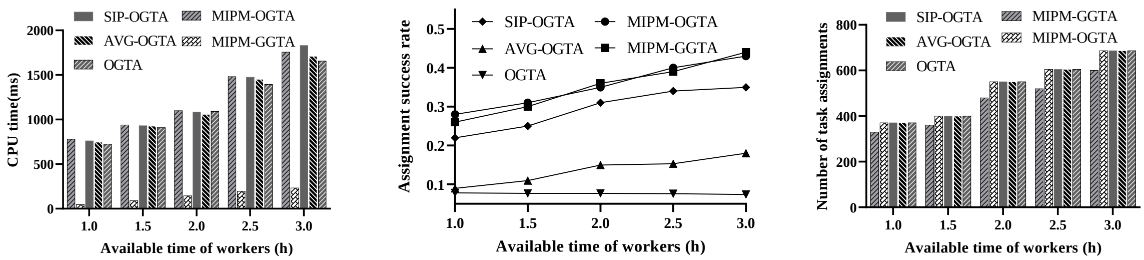


(a) CPU Time

(b) Assignment Success Rate

(c) Number of Task Assignments

Fig. 3. Effect of $e - p$

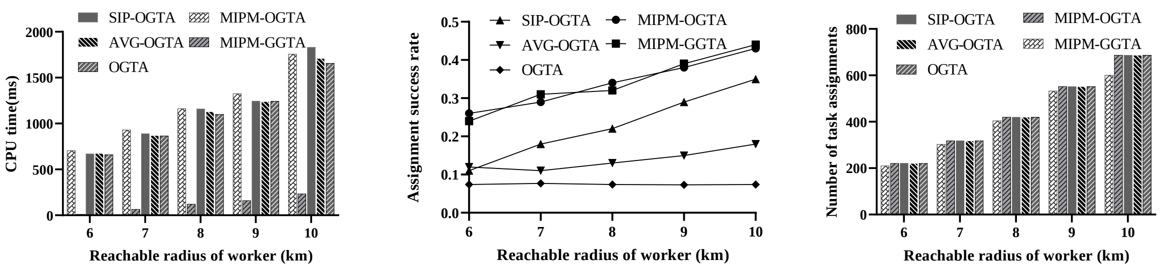


(a) CPU Time

(b) Assignment Success Rate

(c) Number of Task Assignments

Fig. 4. Effect of $off - on$



(a) CPU Time

(b) Assignment Success Rate

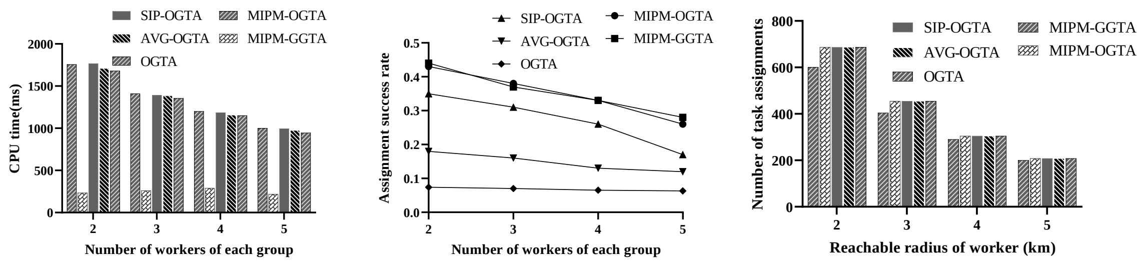
(c) Number of Task Assignments

Fig. 5. Effect of r

Therefore, it can be seen from Figure 3 that the MIPM-OGTA algorithm guarantees both the success rate of task assignment and the number of tasks assigned, which shows the effectiveness of the algorithms proposed in this paper.

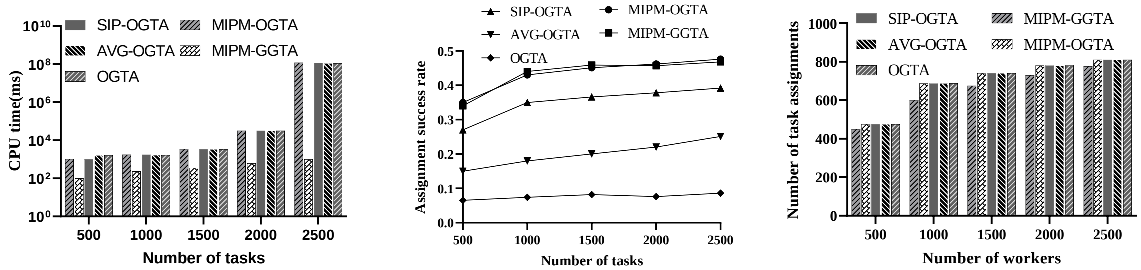
Effect of $\mathit{off} - on$**.** Next, we study how the available time of workers affects the performance of group task assignment. As shown in Figure 4(a), as the available time of workers increases, the CPU time of all algorithms also gradually increases. This is because the group of available workers for each task also increases, resulting in a larger search space. In terms of the success rate of task assignment, as shown in 4(b), the larger the $off - on$, the ASR values of preference-based algorithms show an increasing trend. The reason behind it is similar to that of $e - p$, i.e., an SC platform has a higher probability to assign tasks to workers who are interested in it. The MIPM-OGTA and MIPM-GGTA algorithms have similar task assignment success rates, while the number of tasks assigned by MIPM-OGTA is significantly higher than that of the MIPM-GGTA algorithm (see Figure 4(c)). This is because the optimal task assignment algorithm outperforms the greedy task assignment algorithm in terms of the number of assigned tasks. In addition, the number of tasks assigned to all algorithms gradually increases as the available time of workers increases since that the available worker groups corresponding to each task also increase.

Effect of r**.** In this part of the experiment, we further evaluate the effect of the reachable distance r of workers. It can be seen from Figure 5(a) that when r increases, the CPU time of all OGTA algorithms shows a similar growth trend. The reasons behind this are the following: 1) all methods are based on the tree-decomposition-based optimal task assignment; 2) when the reachable distance of workers increases, the number of the available worker groups for each task also increases, resulting in a larger search space. The MIPM-GGTA algorithm



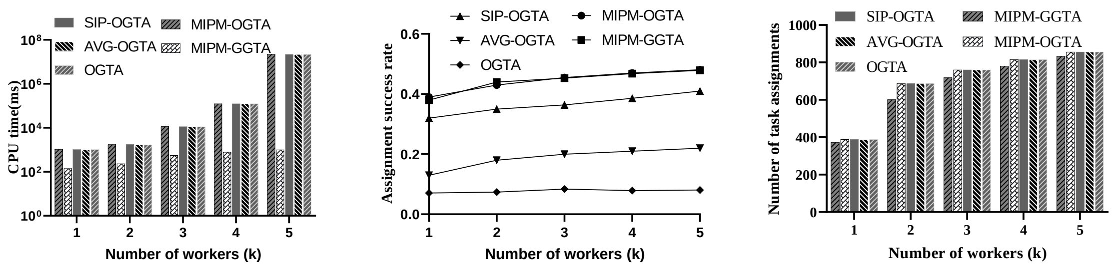
(a) CPU Time (b) Assignment Success Rate (c) Number of Task Assignments

Fig. 6. Effect of $numW$



(a) CPU Time (b) Assignment Success Rate (c) Number of Task Assignments

Fig. 7. Effect of $|S|$



(a) CPU Time (b) Assignment Success Rate (c) Number of Task Assignments

Fig. 8. Effect of $|W|$

still consumes the least CPU time, but its performance in the number of task assignments is obviously not as good as other OGTA-related algorithms (see Figure 5(c)). In addition, as shown in Figure 5(b), with r increasing, the ASR value also increases. This is because the group of workers can be assigned their favorite tasks with a greater probability.

Effect of $numW$. Figure 6 shows the effect of the number of workers in each group on task assignment performance. It can be seen from Figure 6(a) that as $numW$ increases, the number of available workers for each task decreases, reducing the search space in the task assignment process, so the CPU time shows a downward trend. Since the number available worker groups corresponding to each task decreases with the increase of $numW$, the task is less likely to be assigned to a suitable group where workers are interested in this task. As a result, the ASR value shows a downward trend (see Figure 6(b)). In addition, as shown in Figure 6(c), the OGTA-related algorithms perform similarly in terms of the number of tasks assigned, and the MIPM-GGTA algorithm assigns the fewest tasks.

Effect of $|S|$. We study the scalability of the algorithms by changing the size of the number of tasks $|S|$. It can be seen from Figure 7(a) that the CPU time is increasing. At the same time, we can conclude that the CPU time of the OGTA-related algorithms is higher than that of the GGTA-related algorithms. This is because as the number of tasks increases, the OGTA-related algorithms will cause additional tree construction, and searching these trees costs more CPU time. The total number of tasks assigned by MIPM-GGTA is less than those of OGTA-related algorithms (i.e., MIPM-OGTA, SIP-OGTA, AVG-OGTA, and OGTA). In addition, MIPM-OGTA performs better in terms of assignment success rate and task assignment number (see Figures 7(b) and 7(c). With the increase of $|S|$, a worker group can access the tasks that they are more interested in, so the ASR value increases. Moreover, each task corresponds to more available worker groups, so the number of assignments for the task also increases.

Effect of $|W|$. In the final set of experiments, we evaluate the effect of our proposed algorithms on group task assignment performance. As shown in the Figure 8(a), the larger the $|W|$, the longer the CPU time. This is because more and more available workers need to be assigned, which leads to longer time overhead. The reason behind it is similar to the effect of $|S|$, that is, more workers will cause more decomposition and search of the tree in the OGTA procedure, which will consume more CPU time. From the task assignment success rate in Figure 8(b), all preference-based algorithms show an increasing trend with the increase of $|W|$, and the number of task assignments is also increasing (see Figure 8(c)). In summary, it can be seen that MIPM-OGTA performs well in terms of the number of assigned tasks and the success rate of task assignment.

VI. Conclusion

In this paper we propose and offer solutions to a SC problem called Preference-aware Group Task Assignment (PGTA), which aims to find the optimal task assignment with maximal number of assigned tasks while considering worker groups' preferences. In order to relief the data sparsity, we give a Mutual Information-based Preference Modeling method, which learns the group preferences by maximizing the mutual information among workers and using an attention mechanism to model the contributions of different group members. Based on these group preferences, a tree-decomposition approach is adopted to achieve the optimal task assignment. To the best of our knowledge, this is the first study in SC that considers group preferences by exploring the mutual information among workers. An empirical study based on a real dataset confirms the superiority of our proposed algorithms.

Acknowledgment

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

References

[1] T. Song, Y. Tong, L. Wang, J. She, B. Yao, L. Chen, and K. Xu, "Trichromatic online matching in real-time spatial crowdsourcing," *ICDE*, pp. 1009–1020, 2017.

[2] Y. Tong, J. She, B. Ding, L. Chen, T. Wo, and K. Xu, "Online minimum matching in real-time spatial data: Experiments and analysis," *PVLDB*, vol. 9, pp. 1053–1064, 2016.

[3] P. Cheng, X. Lian, Z. Chen, L. Chen, J. Han, and J. Zhao, "Reliable diversity-based spatial crowdsourcing by moving workers," *PVLDB*, vol. 8, pp. 1022–1033, 2015.

[4] H. To, G. Ghinita, and C. Shahabi, "A framework for protecting worker location privacy in spatial crowdsourcing," *PVLDB*, vol. 7, pp. 919–930, 2014.

[5] Y. Zhao, Y. Li, Y. Wang, H. Su, and K. Zheng, "Destination-aware task assignment in spatial crowdsourcing," *CIKM*, 2017.

[6] L. Zheng and L. Chen, "Maximizing acceptance in rejection-aware spatial crowdsourcing," *TKDE*, vol. 29, pp. 1943–1956, 2017.

[7] Y. Tong, L. Wang, Z. Zhou, B. Ding, L. Chen, J. Ye, and K. Xu, "Flexible online task assignment in real-time spatial data," *PVLDB*, vol. 10, pp. 1334–1345, 2017.

[8] P. Cheng, L. Chen, and J. Ye, "Cooperation-aware task assignment in spatial crowdsourcing," *ICDE*, pp. 1442–1453, 2019.

[9] P. Cheng, X. Lian, L. Chen, J. Han, and J. Zhao, "Task assignment on multi-skill oriented spatial crowdsourcing," *TKDE*, vol. 28, pp. 2201–2215, 2016.

[10] Y. Zhao, J. Guo, X. Chen, J. Hao, X. Zhou, and K. Zheng, "Coalition-based task assignment in spatial crowdsourcing," in *ICDE*, 2021.

[11] D. Gao, Y. Tong, J. She, T. Song, L. Chen, and K. Xu, "Top-k team recommendation and its variants in spatial crowdsourcing," *DSE*, vol. 2, pp. 136–150, 2017.

[12] X. Li, Y. Zhao, J. Guo, and K. Zheng, "Group task assignment with social impact-based preference in spatial crowdsourcing," in *DASFAA*, 2020, pp. 677–693.

[13] Y. Cui, L. Deng, Y. Zhao, B. Yao, V. Zheng, and K. Zheng, "Hidden poi ranking with spatial crowdsourcing," *SIGKDD*, 2019.

[14] Y. Tong, Y. Zeng, Z. Zhou, L. Chen, J. Ye, and K. Xu, "A unified approach to route planning for shared mobility," *PVLDB*, vol. 11, pp. 1633–1646, 2018.

[15] J. Xia, Y. Zhao, G. Liu, J. Xu, M. Zhang, and K. Zheng, "Profit-driven task assignment in spatial crowdsourcing," in *IJCAI*, 2019, pp. 1914–1920.

[16] J. Tu, P. Cheng, and L. Chen, "Quality-assured synchronized task assignment in crowdsourcing," *TKDE*, vol. 33, pp. 1156–1168, 2021.

[17] Y. Zhao, K. Zheng, J. Guo, B. Yang, T. B. Pedersen, and C. S. Jensen, "Fairness-aware task assignment in spatial crowdsourcing: Game-theoretic approaches," in *ICDE*, 2021, pp. 265–276.

[18] Y. Zhao, K. Zheng, Y. Cui, H. Su, F. Zhu, and X. Zhou, "Predictive task assignment in spatial crowdsourcing: a data-driven approach," in *ICDE*, 2020, pp. 13–24.

[19] G. Ye, Y. Zhao, X. Chen, and K. Zheng, "Task allocation with geographic partition in spatial crowdsourcing," in *CIKM*, 2021.

[20] Z. Wang, Y. Zhao, X. Chen, and K. Zheng, "Task assignment with worker churn prediction in spatial crowdsourcing," in *CIKM*, 2021.

[21] L. Kazemi and C. Shahabi, "Geocrowd: enabling query answering with spatial crowdsourcing," in *SIGSPATIAL*, 2012, pp. 189–198.

[22] L. Kazemi, C. Shahabi, and L. Chen, "Geotrucrowd: trustworthy query answering with spatial crowdsourcing," *SIGSPATIAL*, pp. 314–323, 2013.

[23] H. To, C. Shahabi, and L. Kazemi, "A server-assigned spatial crowdsourcing framework," *TSAS*, vol. 1, pp. 201–228, 2015.

[24] Y. Tong, L. Chen, Z. Zhou, H. V. Jagadish, L. Shou, and W. Lv, "Slade: A smart large-scale task decomposer in crowdsourcing," *TKDE*, vol. 30, pp. 1588–1601, 2018.

[25] L. Zheng, L. Chen, and J. Ye, "Order dispatch in price-aware ridesharing," *PVLDB*, vol. 11, pp. 853–865, 2018.

[26] X. Li, Y. Zhao, X. Zhou, and K. Zheng, "Consensus-based group task assignment with social impact in spatial crowdsourcing," *DSE*, vol. 5, pp. 375–390, 2020.

[27] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2015.

[28] B. C. Ross, "Mutual information between discrete and continuous data sets," *PLoS ONE*, vol. 9, no. 2, p. e87357, 2014.

[29] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, R. D. Hjelm, and A. C. Courville, "Mutual information neural estimation," in *ICML*, 2018, pp. 531–540.

[30] P. Velickovic, W. Fedus, W. L. Hamilton, P. Lio', Y. Bengio, and R. D. Hjelm, "Deep graph infomax," *ArXiv*, vol. abs/1809.10341, 2019.

[31] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *COMPSTAT*, 2010.

[32] Y. Zhao, K. Zheng, Y. Li, H. Su, J. Liu, and X. Zhou, "Destination-aware task assignment in spatial crowdsourcing: A worker decomposition approach," *TKDE*, vol. 32, pp. 2336–2350, 2020.

[33] D. Cao, X. He, L. Miao, Y. An, C. Yang, and R. Hong, "Attentive group recommendation," *SIGIR*, 2018.

[34] D. Deng, C. Shahabi, and U. Demiryurek, "Maximizing the number of worker's self-selected tasks in spatial crowdsourcing," *SIGSPATIAL*, 2013.

[35] H. Dang, T. A. Nguyen, and H. To, "Maximum complex task assignment: Towards tasks correlation in spatial crowdsourcing," in *IIWAS*, 2013.

[36] P. Cheng, X. Lian, L. Chen, and C. Shahabi, "Prediction-based task assignment in spatial crowdsourcing," *ICDE*, pp. 997–1008, 2017.