Preference-aware Group Task Assignment in Spatial Crowdsourcing: Effectiveness and Efficiency

Yan Zhao, Jiaxin Liu, Yunchuan Li, Dalin Zhang*, Christian S. Jensen, *Fellow, IEEE*, and Kai Zheng*, *Senior Member, IEEE*

Abstract—With the diffusion of online mobile devices with geo-location capabilities, the infrastructure necessary for real-world deployment of Spatial Crowdsourcing (SC), where so-called mobile workers are assigned location-sensitive tasks, is in place. Some SC tasks cannot be completed by a single worker due to their complexity, but rather must be assigned to and completed by a group of users. Achieving such group assignments that satisfy all group members evenly is an open challenge. To address this challenge, we propose a novel preference-aware group task assignment framework encompassing two components: Mutual Information-based Preference Modeling (MIPM) and Preference-aware Group Task Assignment (PGTA). The MIPM component learns the preferences of groups contrastively by maximizing the mutual information between workers and worker groups based on worker-task and group-task interaction data and by using an attention mechanism to weight group members adaptively. In addition, curriculum negative sampling is adopted to generate a small number of negative workers for each worker group, following the principles of curriculum learning. Next, the PGTA component offers an optimal task assignment algorithm that employs tree decomposition to assign tasks to appropriate worker groups, with the aim of maximizing the number of task assignments while prioritizing more interested groups when assigning tasks. The task assignment framework also features preference-constrained pruning of unpromising worker groups to speed up the assignment process. Finally, we report extensive experiments that offer evidence of the effectiveness and practicality of the paper's proposal.

Index Terms—Preference, Group Task Assignment, Mutual Information, Spatial Crowdsourcing

1 Introduction

With the widespread diffusion of online, geo-located mobile devices such as smartphones, the notion of Spatial Crowdsourcing (SC), where location-sensitive tasks are assigned to and completed by mobile workers, is attracting substantial interest in both academia and industry. SC platforms assign tasks to workers that the workers must perform at specific locations, e.g., taking photos or monitoring traffic conditions. Most of existing studies focus on assigning tasks to individual workers [8], [20], [21], [24], [38]. For example, Zheng et al. [42] take the rejection of assigned tasks into consideration and focus on the problem of maximizing the acceptance by workers of assigned tasks. Another study [25] considers a Flexible Two-sided Online task Assignment (FTOA) problem, where prediction of the spatio-temporal distributions of future tasks and workers is used to guide idle workers to locations with work so that the total number of assigned task is increased. However, in some SC applications, a single worker cannot perform a task efficiently and independently. Examples occur in home improvement, furniture installation, monitoring of traffic conditions in an area, organization of an event, or performing entertainment at an event [5], [6], [37]. Rather, in these examples, it takes multiple workers to perform a task, thus calling for group task assignments.

Several group task assignment proposals exist for SC. Gao et al. [13] study a Top-k team recommendation problem called TopkTR and a variant called TopkTRL, both of which aim to recommend suitable crowdsourced teams for tasks. Considering the cooperation of workers, Cheng et al. [5] propose greedy and game-theoretic approaches with the goal of assigning tasks to groups of workers with high cooperation qualities. However, these studies assume that all workers in a worker group are willing to complete the assigned tasks. In practice, some workers may not be interested in an assigned task, resulting in workers rejecting tasks or completing tasks with low quality. Li et al. [17] recently propose a framework for group task assignment that considers the preferences of worker groups by taking into the account the social impact of workers. Our study goes further in this direction by using mutual information among workers to learn group preferences and by optimizing task assignments based on the group preferences.

The example in Figure 1 illustrates the problem of group task assignment. Five workers $\{w_1, ..., w_5\}$, and two tasks $\{s_1, s_2\}$ exist. Each worker has a current location and a reachable distance. Each task requires two workers for its completion and has a location in which it must be performed. Considering the spatio-temporal constraints between workers and tasks, i.e., a task assigned to a worker should be in the reachable range of the worker, and the worker should be

- *Y. Zhao, D. Zhang and C. S. Jensen are with the Department of Computer Science, Aalborg University, Aalborg 9220, Denmark. Email: {yanz, dalinz, csj}@cs.aau.dk.*
- *J. Liu, Y. Li and K. Zheng are with the Yangtze Delta Region Institute (Quzhou), School of Computer Science and Engineering, Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, China. Email: {jiaxinliu1999, liyunchuan}@std.uestc.edu.cn, zhengkai@uestc.edu.cn. *Corresponding authors: K. Zheng and D. Zhang.*

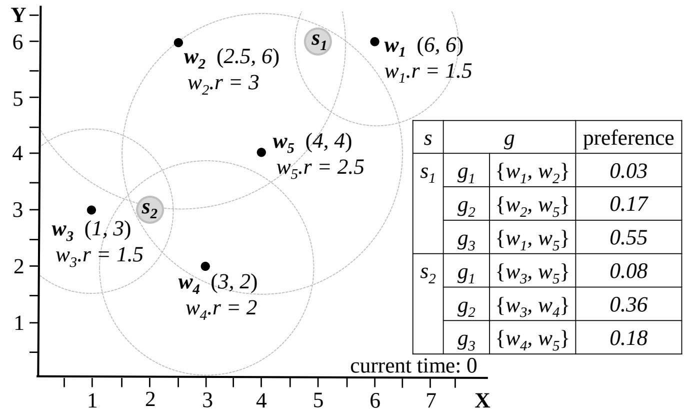


Fig. 1. Running Example

able to arrive at the task location before the deadline for completing the task, 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 $\{\{w_3, w_5\}, \{w_3, w_4\}, \{w_4, w_5\}\}$. Figure 1 also shows the preference values of the worker groups for the available tasks. Without considering group preferences, a general group task assignment problem is to maximize the total number of assigned tasks. In the example, the task assignment $\{(s_1, \{w_1, w_2\}), (s_2, \{w_3, w_5\})\}$ is optimal. However, the two assigned worker groups have extremely low preferences for tasks s_1 and s_2, and the preference value sum for the assignment is only 0.11. Such an assignment may result in workers refusing to accept the task or completing it with low quality.

To address this issue, we propose a data-driven framework that considers the preferences of worker groups when making assignments. 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, it first maximizes the mutual information between worker and group representations to train a discriminator by means of contrastive learning, which aims to adjust preference representations of workers and worker groups. Then, it uses an attention mechanism to form worker groups of highly relevant workers by setting different weights for different workers adaptively, to improve the preference representation of each worker group. Finally, group preferences for different task categories are obtained through a prediction layer. The assignment phase first identifies available worker groups for each task without violating the spatio-temporal constraints. Then it uses a tree-decomposition algorithm to assign a suitable worker group to each task while prioritizing worker groups according to their interests 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 total preference value of 0.91 is obtained.

The initial conference version [19] of this study only considers the similar non-members that do not belong to a worker group as negative samples for the group when computing group preferences, thus ignoring dissimilar non-members. To further improve the effectiveness of group preference learning, the MIMP phase introduces a Curriculum Negative Sampling (CNS) method that follows the principles of curriculum learning. Specifically, given an input worker group, CNS first selects the dissimilar non-members as negative samples, which are easy to distinguish from the input group. Then, it gradually selects non-members that are increasingly similar to the input group and thus are more difficult to distinguish from the input group. The resulting curriculum negative sampling enables effective learning of distinguishable group representations.

For improving the task assignment efficiency, we propose a preference-constrained pruning strategy to disregard worker groups with relatively low preferences for a specific task category, as such groups are unlikely to accept to perform the tasks in that category. Accordingly, the pruning strategy improves the efficiency of task assignment since fewer worker groups need to be considered during task assignment.

In summary, the new technical contributions are five-fold.

1) We identify and study in depth the limitations in our previous group task assignment framework that disregards dissimilar non-member workers when learning group preference and fails to consider the groups with low preferences in task assignment.

2) We adopt a curriculum negative sampling method to select both similar and dissimilar non-member workers as negative samples following the principles of curriculum learning.

3) We propose a preference-constrained pruning strategy for improving efficiency of task assignment.

4) We report on extensive experiments that offer insight into the effects of key parameters and the effectiveness of the new proposal. In particular, the algorithms considering curriculum negative sampling, i.e., CNS-MIPM+OGTA and CNS-MIPM+OGTA+P, can improve the assignment success rate by up to 7.93% and 9.31% over MIPM+OGTA (that does not consider curriculum negative sampling), respectively. With preference-constrained pruning, the CPU time of CNS-MIPM+OGTA+P is only 39.35%–85.33% of that of CNS-MIPM+OGTA (without pruning), while being able to achieve similar assignment success rate and similar number of task assignments of CNS-MIPM+OGTA.

The remainder of this paper is organized as follows. Section 2 covers the related work, and Section 3 introduces notation and defines the problem. Section 4 then proposes a mutual information-based approach to worker group preference modeling and a tree-decomposition-based algorithm for preference-based group task assignment, followed by a coverage of experimental findings in Section 5. Finally, Section 6 offers conclusions.

2 Related Work

Spatial crowdsourcing (SC) is a new framework that has emerged in recent years, requiring workers with GPS devices to travel to a specific location physically under certain restrictions to perform spatial task [9], [10], [16], [24], [28]–[30], [32]–[35], [39], [40]. Most of the existing studies focus on task assignment. Based on the task publishing models, Kazemi et al. [14] divides SC into worker selection task (W-ST) 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 [14], [15], [22], [23], [25], maximizing the coverage of worker skills required by tasks [6], or maximizing the total profit of the platform [33], [43]. For example, Tong et al. propose a Global Online Micro-task Allocation framework in SC, which allocates micro-tasks to suitable workers in online scenarios [26]. Some studies aim to design route planning for shared mobility to achieve effective task assignment in the SC applications such as ride-sharing, food delivery, and crowdsourced parcel delivery [27], [36]. However, these 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 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 single task assignment, group task assignment is more complex, and few studies exist in this field. The recent study [13] proposes a Top-k team recommendation problem in SC, where a method that recommends suitable crowdsourcing teams for each task is proposed. Cheng et al. [5] consider that the completion of complex tasks requires the cooperation of workers and propose a greedy and a game-theoretic method to assign multiple workers with high cooperative qualities to complete a task together. Another related work studies a reliable diversity-based spatial crowdsourcing problem, where each task (e.g., taking videos/photos of a landmark and checking whether or not parking spaces are available) is assigned to multiple workers that are moving towards some directions, and the workers must finish the assigned task before its expiration time [8]. However, these studies on group task assignment do not consider whether workers are interested in the assigned task, which may lead to workers' rejection of performing the task or low-quality completion of the task.

Recently, Li et al. [18] use a bipartite graph embedding model and an attention mechanism to learn the social impact-based preferences of worker groups for different categories of tasks and assign tasks to the groups according to their preferences. In the study [18], 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, e.g., 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 additional information. Instead, it uses historical task-performing data to maximize the mutual information among workers in order to learn the informative representations of groups and group preferences.

3 Problem Definition

We proceed to introduce a set of preliminary concepts and then give our problem statement. Table 1 summarizes the main notation used in the paper.

TABLE 1
Summary of Notation

Notation	Definition
s	Spatial task
$s.l$	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
S	A set of tasks
w	Worker
$w.l$	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
$w.sp$	Speed of worker w
W	A set of workers
$AWS(s)$	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 $w = (l, r, on, off, sp)$, includes a location $w.l$, a reachable distance $w.r$, an online time $w.on$, an offline time $w.off$, and a speed sp. 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 an offline mode. When a worker is ready for performing a task (from the online time $w.on$ to the next offline time $w.off$), the worker is online.

In group task assignment, a task s requires multiple workers to complete it cooperatively. Workers can only handle one task at a specific time, which is reasonable in practice.

Definition 3 (Available Worker Set). The available worker set for 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 online, i.e., $w.on \leq t_{now} \leq w.off$,
2) task s is located in the reachable range of worker w, i.e., $d(w.l, s.l) \leq w.r$,
3) worker w can travel from the current location $w.l$ to the task location $s.l$ directly before it expires, i.e., $t_{now} + t(w.l, s.l) \leq s.e$, and
4) worker w can travel from the current location $w.l$ to the task location $s.l$ 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 locations $w.l$ and $s.l$, and $t(w.l, s.l) = w.sp * d(w.l, s.l)$ is the travel time between locations $w.l$ and $s.l$.

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 their offline times and the expiration time of s. 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., $AWG(s) \subset AWS(s)$,
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., $\forall w_i, w_j \in AWG(s)$ $(t_{now} + t(w_i.l, s.l) \leq w_j.\mathit{off})$.

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., $\forall A_i \in \mathbb{A}$ $(|A_i.S| \leq |A_o.S|)$) 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 in A_i, and $\mathbb{A}$ denotes all the possible ways of assignments.

4 Framework

In this section, we give the details of our proposed group task assignment framework. In practice, 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, where a mutual information-based contrastive representation learning model with curriculum negative sampling is proposed. 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 worker groups for tasks and adopt a tree-decomposition algorithm to obtain the optimal task assignment. In the following, we first give an overview of the framework and then provide specifics on each part in the framework.

4.1 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), and 2) Preference-based Group Task Assignment (PGTA).

In the MIPM part, we notice that group activities often reflect the following two phenomena: 1) the difference between different groups, and 2) the connection among members in a group. To capture these features, we propose a Curriculum Negative Sampling (CNS) method to sample negative samples (i.e., the non-members that do not belong to a group) for an input worker group. Then, we use a Mutual Information-based Contrastive Representation Learning (MI-CRL) model to contrast the representations of the members in a group against those of non-members and maximize the worker-group mutual information to train a discriminator to regularize the representations of workers and groups. The goal of contrastive representation learning is to learn such an embedding space in which similar sample pairs (i.e., the members in a group) stay close to each other while dissimilar ones (i.e., the non-members that do not belong to a group) are far apart. In order to overcome the sparsity of the group-task interaction data, we propose an adaptive group preference weighting technique to set different weights for each worker in a group, which uses the workers' personal preferences to improve group preference representations. 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 a set of available worker groups (AWGs) for each task by considering the spatio-temporal constraints (i.e., the reachable radius and the valid time of workers, as well as the expiration time of tasks). Then we employ the optimal task assignment algorithm based on tree decomposition to assign tasks to appropriate worker groups to maximize the total number of assigned tasks while giving higher priorities to worker groups with higher preferences on tasks.

4.2 Mutual Information-based Preference Modeling

We present how to use historical interaction data to model worker groups' preferences. The main problem of group preference modeling is the sparsity of the group-task interaction data. To solve this problem, we use curriculum negative sampling and mutual information maximization for contrastive representation learning and utilize an adaptive group preference weighting technique to relief the data sparsity. By aggregating the representations of each group member, we can obtain the representations of worker groups. Then the representations of 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.

4.2.1 Mutual Information-based Contrastive Representation Learning with Curriculum Negative Sampling

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

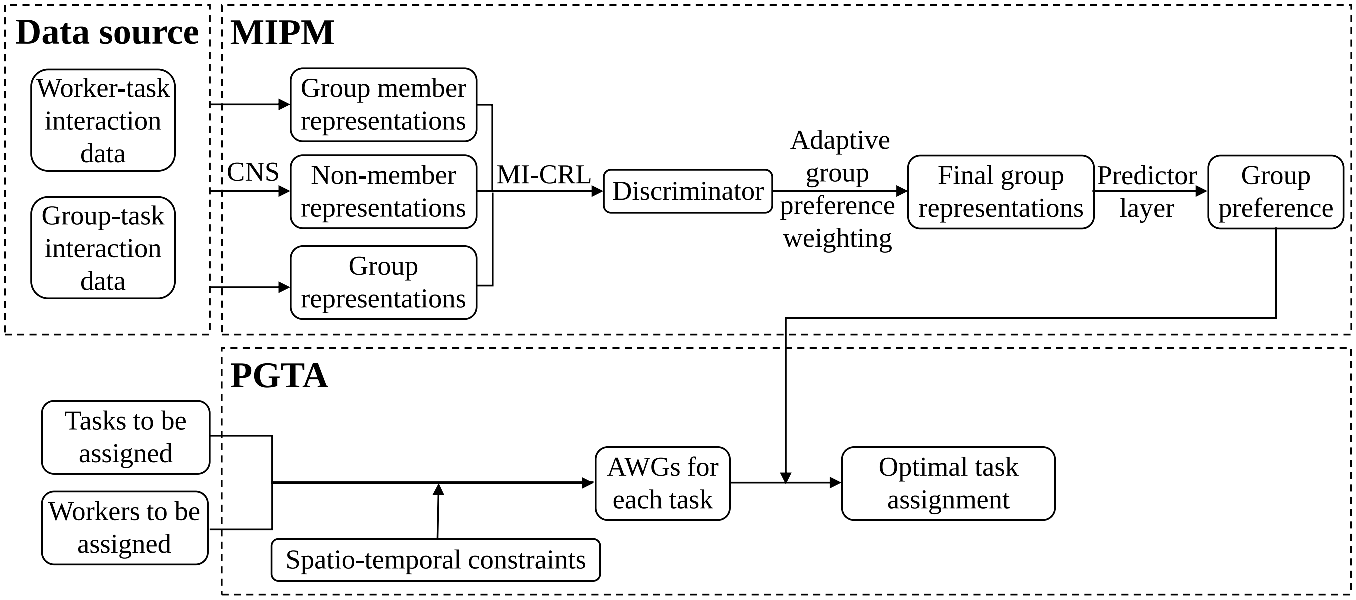


Fig. 2. Framework Overview

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

$$\mathbf{w} = f_e(w, \mathbf{X}_{WC}) = \sigma(K_2^T(\sigma(K_1^T x_w + b_1)) + b_2), \quad (1)$$

where $\mathbf{w}$ denotes the preference representation of worker w, $f_e(\cdot)$ is a preference encoding function, $\mathbf{X}_{WC}$ denotes the worker-task interaction matrix, and x_w is the row of $\mathbf{X}_{WC}$ that corresponds 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 Encoder. Since the preference of a worker group depends on the preference of each group member, we use an attention mechanism [1] to better obtain the representation of the worker group. The attention mechanism can learn the contributions of different group members to the group decision, as a result of which each group member's preference representation can be weighted. The computation is shown in the following.

$$\mathbf{g}_i = \sum_{w_j \in g_i} \alpha(j,i) K_{agg} \mathbf{w}_j, \quad (2)$$

$$\alpha(j,i) = \frac{exp(h^T K_{agg} \mathbf{w}_j)}{\sum_{w_k \in g_i} exp(h^T K_{agg} \mathbf{w}_k)}, \quad (3)$$

where $\mathbf{g}_i$ and $\mathbf{w}_i$ denote the representations of 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 worker group g_i, which can be calculated in Equation 3, where h^T represents the hidden layer parameters of the attention network. Further, K_{agg} denotes represents the weight matrix. A larger $\alpha(j,i)$ means that worker w_j contributes more to the group decision.

Curriculum Negative Sampling. We propose a Curriculum Negative Sampling (CNS) method, which generates a small amount of negative samples for each worker group, following the principles of curriculum learning [3]. The idea behind it is that it starts to train a model with easier samples first, and then gradually increase the difficulty levels. Instead of randomly selecting non-members that do not belong to the group as negative samples, CNS first finds workers that are largely different from the input group members and thus are easy to be distinguished from the input group. Then, we find workers that are increasingly similar to the input worker group and thus are more difficult to be distinguished from the input group. CNS facilitates effective learning of distinguishable worker and group representations.

Specifically, given an input worker group g, we can get a similar worker set and a dissimilar worker set according to whether the workers have performed the tasks with same categories with the input worker group, i.e., a non-member worker $\tilde{w}$ is similar with g if $x_{\tilde{w}}^T \cdot x_g > 0$, where $x_{\tilde{w}}$ is the row (corresponding to $\tilde{w}$) of the worker-task interaction matrix $\mathbf{X}_{WC}$ and x_g represents the row (corresponding to g) of the group-task interaction matrix $\mathbf{X}_{GC}$; otherwise, $\tilde{w}$ is dissimilar with g. We first generate negative worker samples that are different from the input worker group, e.g., sampling m dissimilar workers. In such a case, it can be easy to train worker and group representation encoders that return distinguishable representations of the input worker group and the negative samples. Then, we gradually generate negative worker samples that are increasingly similar to the input worker group, e.g., sampling m workers from the similar and the dissimilar workers where the number of similar workers increases from 1 to m gradually. In such a situation, it is more difficult for the worker and group representation encoders to generate distinguishable worker and group representations. The negative sampling distribution of similar workers is shown as follows:

$$P(\tilde{w}|g) \propto \eta I(x_{\tilde{w}}^T \cdot x_g > 0) + (1-\eta)\frac{1}{|W|}, \quad (4)$$

where η is the sampling ratio, and $I(\cdot)$ is an indicator function. We select the dissimilar workers randomly when sampling them.

Mutual Information-based Contrastive Representation Learning. Inspired by the success of the MINE framework [2] that verifies the flexibility and effectiveness of Mutual Information (MI) maximization methods in discriminator networks to distinguish positive and negative samples accurately, we design a Mutual Information-based Contrastive Representation Learning method. Through such a contrast method, using a scoring function can increase the scores of positive cases and reduce the scores of negative cases.

More specifically, we maximize the mutual information between workers and groups, i.e., the mutual information between the representation of the group members (obtained by Equation 1) and the representation 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 scoring function of the worker-group pair. Compared with non-members, the members in group g have higher scores.

Furthermore, we use a simple bilinear function to calculate the score of the worker-group representation pair, i.e., $\mathcal{D}(\mathbf{w}, \mathbf{g}) = \sigma(\mathbf{w}^T \mathcal{W} \mathbf{g})$, where $\sigma(\cdot)$ is a non-linear function, and $\mathcal{W}$ is a learnable score matrix. The positive sample $(\mathbf{w}, \mathbf{g})$ in the network $\mathcal{D}$ represents the latent preference representation of the worker-group pair (w, g), where $w \in g$. The negative sample is denoted as $(\tilde{\mathbf{w}}, \mathbf{g})$, where $\tilde{\mathbf{w}}$ is the representation of the non-members that do not belong to group g.

We train the discriminator $\mathcal{D}$ using a noise-contrast target, and calculate a Binary Cross-Entropy loss function between the positive and negative samples. The objective function is shown in the following.

$$
\begin{aligned}
O_{MI} = &-\frac{1}{|G|} \sum_{g \in G} \frac{1}{\mu_g} \big[\sum_{w \in g} log \mathcal{D}(\mathbf{w}, \mathbf{g}) \\
&+ \sum_{i=1}^{m} \mathbb{E}_{\tilde{w} \sim CNS(g)} log(1 - \mathcal{D}(\tilde{\mathbf{w}}, \mathbf{g})) \big],
\end{aligned} \tag{5}
$$

where G denotes all the worker groups, and μ_g represents the sum of the number $|g|$ of workers in group g and the number m of negative worker samples of g, i.e., $\mu_g = |g| + m$. Next, $\mathcal{D}(\mathbf{w}, \mathbf{g})$ denotes the score between worker w and group g, and $\mathbb{E}_{\tilde{w} \sim CNS(g)} log(1 - \mathcal{D}(\tilde{\mathbf{w}}, \mathbf{g}))$ is the mathematical expectation of $log(1 - \mathcal{D}(\tilde{\mathbf{w}}, \mathbf{g}))$, where $\tilde{w}$ is sampled based on the CNS of group g, i.e., $\tilde{w} \sim CNS(g)$. The objective function is based on the Jensen-Shannon divergence to maximize the mutual information between the worker representation $\mathbf{w}$ and the worker group representation $\mathbf{g}$ [31]. When training the discriminator, by contrasting group members against non-members, we can learn the discriminative characteristics shared by group members, thereby improving the representations of workers and groups.

4.2.2 Adaptive Group Preference Weighting

In this section, we present a 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 interaction data $\mathbf{X}_{GC}$, we use a polynomial likelihood equation to optimize the group loss function to obtain the group representation $\mathbf{g}$. The group representation $\mathbf{g}$ 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 set 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:

$$
O_{group} = -\sum_{g \in G} \frac{1}{|x_g|} \sum_{c \in C} x_{gc} log \pi_c(\mathbf{g}), \tag{6}
$$

where x_g denotes the row corresponding to group g of group-task interaction matrix $\mathbf{X}_{GC}$, x_{gc} denotes the interaction between group g and task category c, $\pi_c(\mathbf{g}) = softmax(K_C \mathbf{g})$, and K_C represents the weight matrix of the prediction layer. Similarly, based on the polynomial likelihood equation and using the worker-task interaction data $\mathbf{X}_{WC}$ to adjust the worker representation w, the worker's objective function is calculated as follows:

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

Equation 7 is to predict the preferences of all the groups that worker w belongs to based on w's representation $\mathbf{w}$, which will lead to the overfitting problem since the group-task interaction data is very sparse and lacks flexibility in applying preferences among different worker groups. 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 of the group. In order to measure the contextual relevance, we introduce an adaptive group preference weighting strategy 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). Workers with more contextual information will have higher scores. We use the discriminator score as the adaptive group 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 group representation $\mathbf{g}$. Equation 8 shows the objective function of the adaptive group preference weighting strategy.

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

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 [4] strategy to minimize the objective function.

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

where λ is a parameter controlling the contribution of O_{WG}.

By optimizing the above objective function, we can obtain the group representation $\mathbf{g}$ for each worker group g and then obtain the group preferences for different task categories through a prediction layer. 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. We define the worker objective function O_{worker} (shown in Equation 7) to regularize the worker representations with worker-task interactions $\mathbf{X}_{WC}$, thus facilitating the joint training with shared encoder $f_e(\cdot)$ and prediction layer. The calculated preference, $\pi(\mathbf{g})$, of each worker group g will be used in the next phase.

4.3 Preference-based Group Task Assignment

In this section, we first obtain the available worker group sets for each task and then adopt a tree-decomposition algorithm [38], [41] to assign tasks to appropriate worker groups based on their preferences.

4.3.1 Available Worker Group Set (AWGS) Generation

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 all available worker groups of s, and $|\mathbb{AWG}(s)|$ denotes the number of groups in $\mathbb{AWG}(s)$. The time complexity of computing AWGs is $O\left(|S| \cdot |AWS_{\max}|^{numW}\right)$, where $|S|$ is the number of tasks to be assigned, $|AWS_{\max}|$ denotes the maximal number of available workers among all the tasks (i.e., $|AWS_{\max}| = \max_{s\in S}|AWS(s)|$), and $numW$ is the number of group members.

Preference-constrained Pruning. We observe that some available worker groups with relatively low preference exist, which means that workers in these groups are unwilling to perform tasks together. Based on this observation, we propose a preference-constrained pruning strategy that prunes the available worker groups with low preference to further enhance the task assignment efficiency. Specifically, for each task s, we sort its available worker groups in $\mathbb{AWG}(s)$ according to their preference for s descendingly. Then we remove the top-k available worker groups with low preference for task s, where k is a preference-constrained factor. Due to the fact that some tasks may have less-than-k available worker groups, they have no available worker groups after pruning, which impacts the final task assignment negatively, e.g., reducing the number of task assignments. To solve this issue, we introduce a lower bound, LB, to enable enough available worker groups for each task s after pruning, where $LB = \min\{\mathcal{L}, |\mathbb{AWG}(s)|\}$. Here, $\mathcal{L}$ is a user- or platform-specified threshold, which determines the minimal number of available worker groups after pruning. In practice, $\mathcal{L}$ can be set based on the complexity and difficulty of tasks. For example, a complex and difficulty task needs more workers (with a larger $\mathcal{L}$ value) to be finished. If a task s has enough available worker groups, i.e., $|\mathbb{AWG}(s)| \geq \mathcal{L}$, the lower bound, LB, is set to $\mathcal{L}$; otherwise, LB is set to $|\mathbb{AWG}(s)|$. We study the effect of k in our experimental part in Section 5.2. The studies show that the pruning strategy with a suitable preference-constrained factor k can result in the same task assignment result as does AWGS generation without pruning, and it can improve the efficiency of the task assignment.

4.3.2 Task Assignment based on Group Preference

In order to assign suitable worker groups to tasks, we use an optimal assignment algorithm based on tree decomposition [38], [41]. 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 the task dependency graph into clusters, each of which is a maximal clique. A clique is maximal if and only if it is not a subset of the other cliques. The time complexity of the task dependency graph construction is $O(|S|^2 \cdot |AWS_{max}|)$, where $|AWS_{max}|$ is the maximal number of available workers for all the tasks.

Next, we use the Recursive Tree Construction (RTC) algorithm [38], [41] to organize these clusters into a balanced tree structure, where the sibling nodes in the balanced tree do not share the common available workers. The time complexity of RTC is $O(\sum_i^m(|\mathbb{X}^i| + |G_{sub}^i| \cdot (|V^i| + |E'^i|)))$, where m is the number of recursions of the RTC algorithm. Next, $\mathbb{X}^i$, G_{sub}^i, V^i, and E'^i denote the task cluster set, the subgraph set, the vertex set, and the edge set to be checked in the ith recursion, respectively.

After getting the tree, we use the depth-first search method to independently solve the optimal assignment subproblem 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. The time complexity of the search procedure is $O(\sum_i^r(|S_N^i| \cdot |Q_s^i| + |N_{child}^i|))$, where r denotes the number of recursions when searching, $|S_N^i|$ is the number of tasks in the tree node N in the ith recursion, $|Q_s^i|$ is the number of AWGs for task s in the ith recursion, and $|N_{child}^i|$ is the number of child nodes of N in the ith recursion.

4.4 Limitations of PGTA

Our PGTA problem aims to assign a group of workers to each task, but it ignores the waiting time of group members. Waiting time is important in group task assignment especially in the scenarios where tasks need multiple workers to cooperate to finish them. However, our proposed algorithms can be extended to handle such scenarios by introducing some waiting time constraints when computing the available worker groups to ensure that the workers in a group can arrive at the assigned task at almost the same time.

Although our preference-based group task assignment algorithm can achieve effective task assignment, the calculation is relatively inefficient. From the time complexity analysis in Section 4.3, we can see that the cost is dominated by the AWG generation phase with an exponential time complexity, which is computationally expensive when

$numW$ (the number of workers required to be assigned to perform a task) is large. Therefore, our task assignment algorithm is not suitable for the tasks that require a large number of workers to perform them. However, in practice, the algorithm is still efficient because the number of required workers for each task is relatively small. Besides, since the AWG generation of each task is independent with each other, we can calculate the AWGs for each task in parallel to improve efficiency.

5 EXPERIMENT

We evaluate the performance of the group preference modeling and the group task assignment on real data, respectively, where the experimental setup is presented in Section 5.1, followed by the major experimental results in Section 5.2.

5.1 Experimental Setup

5.1.1 Dataset

The experiments are carried out on a 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 62,462 POI locations and 61,412 user locations. The dataset is used widely in the experimental evaluation of SC platforms [7], [11], [12]. Since the dataset lacks the corresponding category information of the venue, we use FourSquare's API[1] to generate its category information. Considering that the dataset lacks information about the geographic location of workers and tasks, for each worker/task, we take the average value of the corresponding check-in location as the location. For each check-in, we simulate that a user is a worker, and the venue accessed by the user is the task performed by the worker. The speed of workers are set to 5km/h. The publication 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 (i.e., 10 km), and workers visiting the same category of tasks within a certain period of time (i.e., 1 hour) 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.

5.1.2 Evaluation Methods

We verify the efficiency and effectiveness of our methods by comparing the following methods:

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

2) **SIP+OGTA**: the OGTA method with worker groups' Social Impact-based Preference [17].

3) **MIPM+GGTA**: the Greedy Group Task Assignment (GGTA) method with worker groups' preferences calculated by the Mutual Information-based Preference Modeling, where the negative samples are the similar non-members sampled from the distribution $P(\tilde{w}|g)$ in Equation 5. MIPM+GGTA assigns each task to the worker group with the maximal preference from the unassigned workers, until all the tasks are assigned or all the workers are exhausted.

4) **MIPM+OGTA**: our OGTA method with MIPM.

5) **CNS-MIPM+OGTA**: our MIPM+OGTA method with Curriculum Negative Sampling.

6) **CNS-MIPM+OGTA+P**: our CNS-MIPM-OGTA method with preference-constrained Pruning, where $\mathcal{L}$ is set to 3.

5.1.3 Metrics

Three main metrics are compared for the above methods, i.e., CPU time, Assignment Success Rate (ASR), and the number of task assignments, for finding task assignments. ASR is the ratio of successful assignments to the total assignments for all workers in a certain time instance. In the experiments, once a certain period of time (that is set to 1 hour in our settings) when all members of a group actually perform (check in) tasks (locations) with the same category that 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. A larger ASR value implies more accurate assignments.

Table 2 shows our experimental settings, which gives the default values of all parameters. We conduct the experiments on Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz with 32 GB RAM.

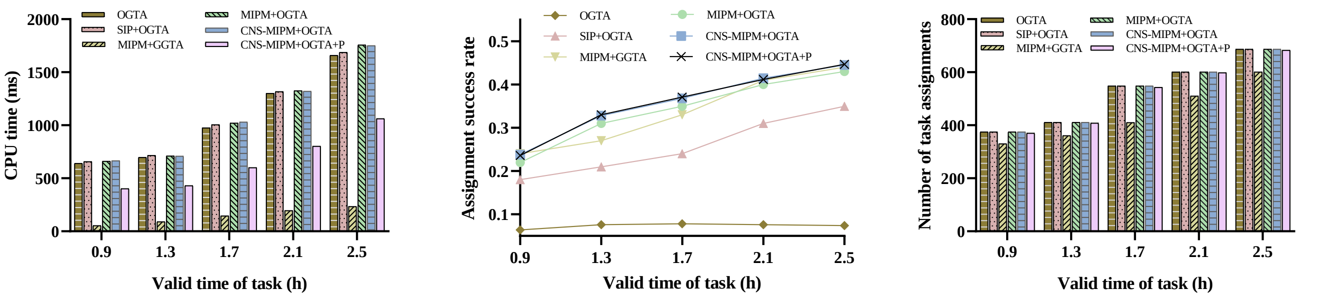
TABLE 2
Experiment Parameters

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
Preference-constrained factor, k	4

5.2 Experiment Results

Effect of $e-p$. We first evaluate the effect of tasks' valid time, $e-p$, on the performance of group task assignment (see Figure 3). It can be seen that the CPU time of all methods shows an increasing trend as the valid time of tasks increases. This is because as the valid time of tasks increases, more groups of available workers exist, which lead to a larger search space. The CPU time of OGTA-related methods except the pruning one (i.e., CNS-MIPM+OGTA+P) show similar trends because these methods all use the optimal task assignment methods based on tree decomposition and have similar time complexity. CNS-MIPM+OGTA+P is most efficient among the OGTA-related methods, demonstrating the superiority of the pruning strategy. As expected, the preference-based task assignment methods (i.e., SIP+OGTA, MIPM+GGTA, MIPM+OGTA, CNS-MIPM+OGTA, and CNS-MIPM+OGTA+P) all increase in terms of ASR (see Figure 3(b)) when the tasks' valid time gets larger. The reason behind it is that as $e-p$ increases, each worker group will have more

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



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

Fig. 3. Effect of $e - p$

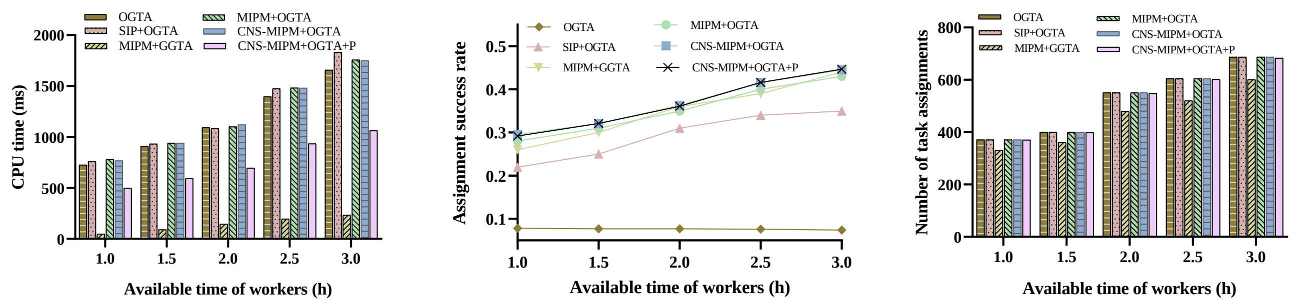
opportunities to be assigned tasks of interest, thereby improving success rate of task assignment. The ASR values of MIPM+OGTA and MIPM+GGTA are higher than than those of OGTA and SIP+OGTA, which shows the advantage of mutual information-based preference modeling. CNS-MIPM+OGTA related methods perform better than MIPM+OGTA related ones, showing the superiority of the curriculum negative sampling strategy. CNS-MIPM+OGTA and CNS-MIPM+OGTA+P are neck to neck in terms of ASR and the number of task assignments (cf. Figure 3(c)), which depicts that the pruning strategy can improve the task assignment efficiency without loss of assignment effectiveness. As shown in Figure 3(c), the MIPM+GGTA method has the least number of task assignments, while the OGTA-related methods can generate more task assignments, which shows the superiority of the optimal task assignment method. Therefore, it can be concluded from Figure 3 that the CNS-MIPM+OGTA methods guarantee both the success rate of task assignment and the number of tasks assigned, which shows the effectiveness of the methods proposed in this paper.

Effect of $off - on$. 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 methods also gradually increase with the similar reason of the effect of tasks' valid time, i.e., the number of available worker groups for each task increases, resulting in a larger search space. In terms of the success rate of task assignment in Figure 4(b), as $off - on$ gets larger, the ASR values of preference-based methods 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 more interested in it. MIPM+OGTA and MIPM+GGTA have similar task assignment success rates, while the number of tasks assigned by MIPM+OGTA is significantly higher than that of MIPM+GGTA (see Figure 4(c)). This is because the optimal task assignment method outperforms the greedy task assignment method in terms of the number of assigned tasks. In addition, the number of tasks assigned to all methods gradually increases as the available time of workers increases since that the number of available worker groups corresponding to each task also increases. The proposed CNS-MIPM+OGTA+P method can achieve the highest assignment success rate and competitive number of task assignments with less CPU time compared with the MIPM+OGTA related methods, i.e., the CPU time is only 60.50%–64.97% of the MIPM+OGTA related methods, which shows the benefits of the curriculum negative sampling and the pruning strategy.

Effect of r. 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 related methods show a similar growth trend. The reasons behind it are in the following: 1) all methods are based on the tree-decomposition-based optimal task assignment; and 2) when the reachable distance of workers increases, the number of the available worker groups for each task also increases, leading to a larger search space. MIPM+GGTA still consumes the least CPU time, but its performance in the number of task assignments is poor compared with others (see Figure 5(c)). Additionally, as shown in Figure 5(b), with r increasing, the ASR values of the methods considering worker group preferences also increase. This is because the group of workers can be assigned their interested tasks with a higher probability. We also observe that CNS-MIPM+OGTA and CNS-MIPM+OGTA+P can improve the assignment success rate by up to 7.93% and 9.31% over MIPM+OGTA, respectively, which shows the superiority of considering curriculum negative sampling in workers' preference learning.

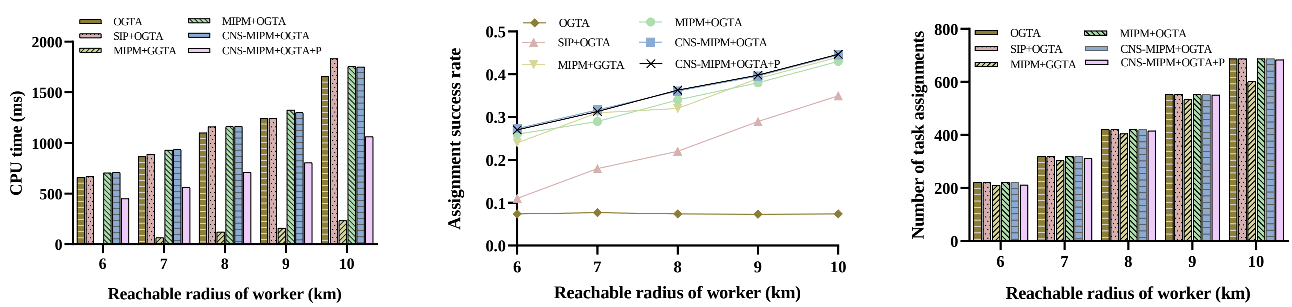
Effect of $numW$. Figure 6 shows the effect of the number of workers in each group on task assignment performance. It can be seen from Figure 6(a) that as $numW$ increases, the number of available workers for each task decreases, reducing the search space in the task assignment process, so the CPU time of all methods shows a downward trend. Since the number of 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 the task. As a result, the ASR values of all the methods except OGTA (that does not consider worker group preferences) show a downward trend (see Figure 6(b)). In addition, as shown in Figure 6(c), the OGTA related methods perform similarly in terms of the number of assigned tasks, and MIPM+GGTA assigns the fewest tasks. Overall, CNS-MIPM+OGTA+P achieves good balance between efficiency and effectiveness.

Effect of $|S|$. We study the scalability of the methods by changing the size of the number of tasks $|S|$. From



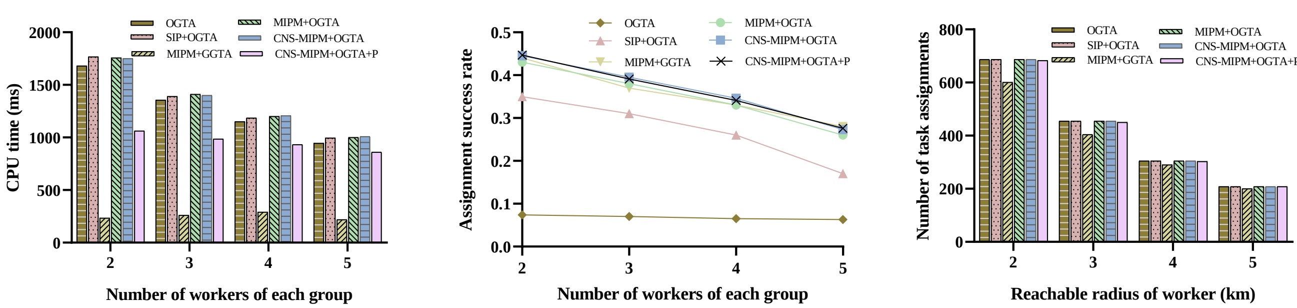
(a) CPU Time (b) Assignment Success Rate (c) Number of Task Assignments

Fig. 4. Effect of $off - on$



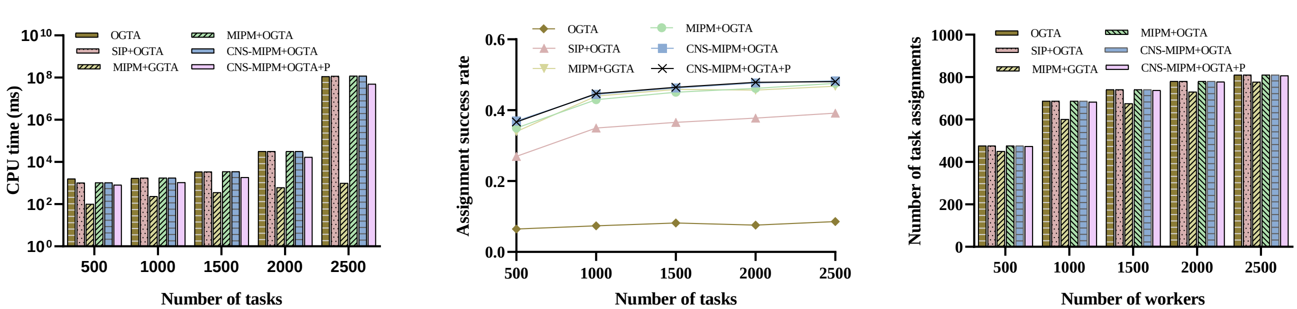
(a) CPU Time (b) Assignment Success Rate (c) Number of Task Assignments

Fig. 5. Effect of r



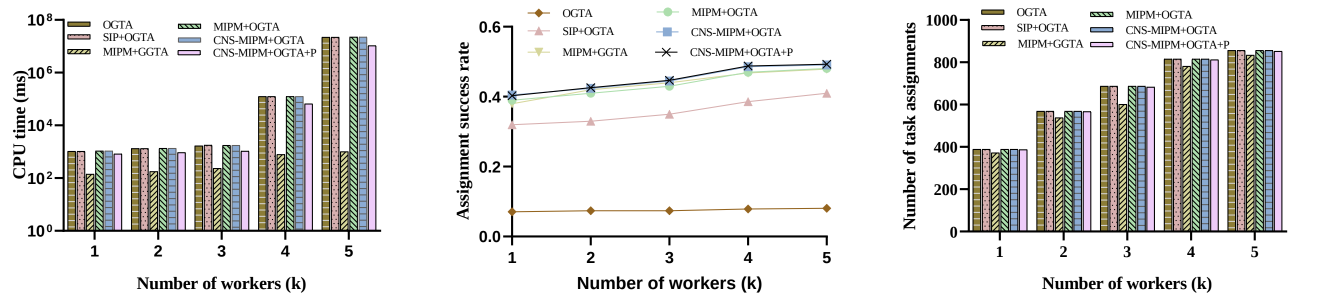
(a) CPU Time (b) Assignment Success Rate (c) Number of Task Assignments

Fig. 6. Effect of $numW$



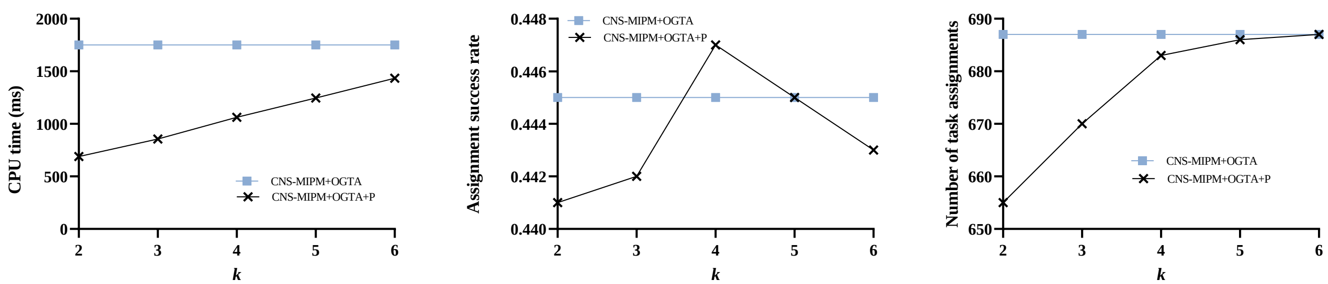
(a) CPU Time (b) Assignment Success Rate (c) Number of Task Assignments

Fig. 7. Effect of $|S|$



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

Fig. 8. Effect of $|W|$



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

Fig. 9. Effect of k

Figure 7(a) we can see that the CPU time of all methods increases. At the same time, we observe that the performance gaps between the OGTA-related methods (i.e., OGTA, SIP+OGTA, MIPM+OGTA, CNS-MIPM+OGTA, and CNS-MIPM+OGTA+P) and MIPM+GGTA become larger when $|S|$ increases. This is because as the number of tasks increases, the OGTA-related methods will cause additional tree construction, and searching these trees costs more CPU time. However, the total number of tasks assigned by MIPM+GGTA is less than those of OGTA-related methods, as shown in Figure 7(c). With the increase of $|S|$, a worker group can access the tasks that they are more interested in, so the ASR values increase (see Figure 7(b)). Moreover, each task corresponds to more available worker groups, so the number of task assignments also increases (see Figure 7(c)). In general, MIPM+OGTA related methods (including MIPM+OGTA, CNS-MIPM+OGTA, and CNS-MIPM+OGTA+P) perform better than others in terms of assignment success rate and task assignment number, demonstrating their superiority.

Effect of $|W|$. We proceed to evaluate the effect of $|W|$. As shown in the Figure 8(a), the larger the $|W|$, the longer the CPU time of all methods is. This is because more and more available workers need to be assigned, which leads to longer time overhead. When it comes to the task assignment success rate in Figure 8(b), all preference-based methods show an increasing trend with the increase of $|W|$, and the number of task assignments also increases (see Figure 8(c)). In summary, MIPM+OGTA related methods perform well in terms of the number of assigned tasks and the assignment success rate, among which CNS-MIPM+OGTA+P achieves a good balance between efficiency and effectiveness.

Effect of k. In the final set of experiments, we study the effect of preference-constrained factor, k, which is used to prune available worker groups with low preference when finding available worker group sets (see Section 4.3.1). We change k from 2 to 6. In this set of experiments, we mainly evaluate the performance of CNS-MIPM+OGTA without pruning and CNS-MIPM+OGTA+P with pruning. As shown in Figure 9(a), CNS-MIPM+OGTA+P can improve the CPU time substantially, i.e., its CPU time is 39.35%–81.90% of that of CNS-MIPM+OGTA. In Figures 9(b) and 9(c), CNS-MIPM+OGTA+P can achieve higher and same assignment success rate when $k = 4$ and $k = 5$, and performs slightly worse than CNS-MIPM+OGTA in terms of the number of task assignments, i.e., the number of task assignments of CNS-MIPM+OGTA+P is 99.42%–99.85% of that achieved by CNS-MIPM+OGTA, when $k = 4$ and $k = 5$. It is noteworthy that CNS-MIPM+OGTA+P can achieve the same number of task assignments as CNS-MIPM+OGTA when $k = 6$. This demonstrates the superiority of the pruning strategy for solving the group task assignment problem when a suitable k is selected.

Summary of the empirical study. The findings of the empirical study can be summarized as follows:

1) OGTA-related algorithms achieve the maximum number of task assignments but at the cost of high CPU time;

2) MIPM+GGTA is the most efficient algorithm but performs the worst in terms of the number of task assignments;

3) CNS-MIPM+OGTA and CNS-MIPM+OGTA+P are

neck to neck in terms of obtaining the highest assignment success rate and the maximal number of task assignments, but CNS-MIPM+OGTA+P is more efficient;

4) CNS-MIPM+OGTA+P achieves a better balance between effectiveness (including the assignment success rate and the number of task assignments) and efficiency (second to the greedy algorithm, MIPM+GGTA).

6 Conclusion

In this paper we propose and offer solutions to an SC problem called Preference-aware Group Task Assignment, which aims to find the optimal task assignment with the 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 groups and using an attention mechanism to model the contributions of different group members. Based on the group preferences, a tree-decomposition approach is adopted to achieve the optimal task assignment. We further improve the group task assignment framework by integrating a curriculum negative sampling method and a preference-constrained pruning strategy to achieve effective group preference learning and efficient task assignment. An extensive empirical study based on a real dataset confirms the superiority of our proposed methods.

Acknowledgment

This work is partially supported by NSFC (No. 61972069, 61836007, and 61832017), Shenzhen Municipal Science and Technology R&D Funding Basic Research Program (JCYJ20210324133607021), and Municipal Government of Quzhou under Grant No. 2022D037.

References

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

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

[3] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *ICML*, pages 41–48, 2009.

[4] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT*, pages 177–186, 2010.

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

[6] P. Cheng, X. Lian, L. Chen, J. Han, and J. Zhao. Task assignment on multi-skill oriented spatial crowdsourcing. *TKDE*, 28(8):2201–2215, 2016.

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

[8] P. Cheng, X. Lian, Z. Chen, L. Chen, J. Han, and J. Zhao. Reliable diversity-based spatial crowdsourcing by moving workers. *PVLDB*, 8(10):1022–1033, 2015.

[9] Y. Cheng, B. Li, X. Zhou, Y. Yuan, G. Wang, and L. Chen. Real-time cross online matching in spatial crowdsourcing. In *ICDE*, pages 1–12, 2020.

[10] Y. Cui, L. Deng, Y. Zhao, B. Yao, V. W. Zheng, and K. Zheng. Hidden poi ranking with spatial crowdsourcing. In *SIGKDD*, pages 814–824, 2019.

[11] D. Deng, C. Shahabi, and U. Demiryurek. Maximizing the number of worker's self-selected tasks in spatial crowdsourcing. *SIGSPATIAL*, pages 314–323, 2013.

[12] D. Deng, C. Shahabi, and U. Demiryurek. Maximum complex task assignment: Towards tasks correlation in spatial crowdsourcing. In *SIGSPATIAL*, pages 314–323, 2013.

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

[14] L. Kazemi and C. Shahabi. Geocrowd: enabling query answering with spatial crowdsourcing. In *SIGSPATIAL*, pages 189–198, 2012.

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

[16] B. Li, Y. Cheng, Y. Yuan, G. Wang, and L. Chen. Three-dimensional stable matching problem for spatial crowdsourcing platforms. In *SIGKDD*, pages 1643–1653, 2019.

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

[18] X. Li, Y. Zhao, X. Zhou, and K. Zheng. Consensus-based group task assignment with social impact in spatial crowdsourcing. *DSE*, 5(4):375–390, 2020.

[19] Y. Li, Y. Zhao, and K. Zheng. Preference-aware group task assignment in spatial crowdsourcing: A mutual information-based approach. In *ICDM*, pages 350–359, 2021.

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

[21] H. To, G. Ghinita, and C. Shahabi. A framework for protecting worker location privacy in spatial crowdsourcing. *PVLDB*, 7(10):919–930, 2014.

[22] H. To, C. Shahabi, and L. Kazemi. A server-assigned spatial crowdsourcing framework. *TSAS*, 1(1):201–228, 2015.

[23] Y. Tong, L. Chen, Z. Zhou, H. V. Jagadish, L. Shou, and W. Lv. Slade: A smart large-scale task decomposer in crowdsourcing. *TKDE*, 30(8):1588–1601, 2018.

[24] 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*, 9(12):1053–1064, 2016.

[25] 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*, 10(12):1334–1345, 2017.

[26] Y. Tong, Y. Zeng, B. Ding, L. Wang, and L. Chen. Two-sided online micro-task assignment in spatial crowdsourcing. *TKDE*, 33(5):2295–2309, 2019.

[27] Y. Tong, Y. Zeng, Z. Zhou, L. Chen, and K. Xu. Unified route planning for shared mobility: An insertion-based framework. *TODS*, 47(1):1–48, 2022.

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

[29] Y. Tong, Z. Zhou, Y. Zeng, L. Chen, and C. Shahabi. Spatial crowdsourcing: a survey. *VLDBJ*, 29(1):217–250, 2020.

[30] J. Tu, P. Cheng, and L. Chen. Quality-assured synchronized task assignment in crowdsourcing. *TKDE*, 33(3):1156–1168, 2021.

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

[32] Z. Wang, Y. Zhao, X. Chen, and K. Zheng. Task assignment with worker churn prediction in spatial crowdsourcing. In *CIKM*, pages 2070–2079, 2021.

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

[34] Y. Yang, Y. Cheng, Y. Yuan, G. Wang, L. Chen, and Y. Sun. Privacy-preserving cooperative online matching over spatial crowdsourcing platforms. *PVLDB*, 16(1):51–63, 2022.

[35] G. Ye, Y. Zhao, X. Chen, and K. Zheng. Task allocation with geographic partition in spatial crowdsourcing. In *CIKM*, pages 2404–2413, 2021.

[36] Y. Zeng, Y. Tong, Y. Song, and L. Chen. The simpler the better: An indexing approach for shared-route planning queries. *PVLDB*, 13(13):3517–3530, 2020.

[37] Y. Zhao, J. Guo, X. Chen, J. Hao, X. Zhou, and K. Zheng. Coalition-based task assignment in spatial crowdsourcing. In *ICDE*, pages 241–252, 2021.

[38] Y. Zhao, Y. Li, Y. Wang, H. Su, and K. Zheng. Destination-aware task assignment in spatial crowdsourcing. In *CIKM*, pages 297–306, 2017.

[39] 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*, pages 13–24, 2020.

[40] 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*, pages 265–276, 2021.

[41] 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*, 32(12):2336–2350, 2020.

[42] L. Zheng and L. Chen. Maximizing acceptance in rejection-aware spatial crowdsourcing. *TKDE*, 29(9):1943–1956, 2017.

[43] L. Zheng, L. Chen, and J. Ye. Order dispatch in price-aware ridesharing. *PVLDB*, 11(8):853–865, 2018.

Yan Zhao is an Assistant Professor with Aalborg University. She received the PhD Degree in Computer Science from Soochow University in 2020. Her research interests include spatial database and trajectory computing.

Jiaxin Liu received the bachelor's degree in Computer Science and Technology from Southwest Jiaotong University, in 2021. She is currently studying for a master's degree in Computer Science and Technology at the University of Electronic Science and Technology of China. Her research interests include spatial database and trajectory data mining.

Yunchuan Li received the bachelor's degree in Software Engineering from Southwest Petroleum University, in 2019. He is currently studying for a master's degree in Computer Science and Technology at the University of Electronic Science and Technology of China. His research interests include spatial database and anomaly detection.

Dalin Zhang is an assistant professor of Computer Science at Aalborg University, Denmark. He received his Ph.D. degree from the School of Computer Science and Engineering at the University of New South Wales (UNSW), Australia in 2020. His current research interests include data mining, machine learning, time-series analysis, and their applications to Internet of Things, human activity recognition, and braincomputer interface.

Christian S. Jensen is a professor of computer science at Aalborg University, Denmark. He was a professor at Aarhus University from 2010 to 2013, and he was previously with Aalborg University for two decades. His research concerns data management and analytics, and its focus is on temporal and spatio-temporal data. He is a member of the Academia Europaea, the Royal Danish Academy of Sciences and Letters, and the Danish Academy of Technical Sciences. He has received several national and international awards for his research, most recently the IEEE TCDE impact award. He is a fellow of the IEEE and ACM.

Kai Zheng is a Professor of Computer Science with University of Electronic Science and Technology of China. He received his PhD degree in Computer Science from The University of Queensland in 2012. He has been working in the area of spatial-temporal databases, uncertain databases, social-media analysis, inmemory computing and blockchain technologies. He has published over 100 papers in prestigious journals and conferences in data management field such as SIGMOD, ICDE, VLDB Journal, ACM Transactions and IEEE Transactions. He is a senior member of IEEE.