A Multi-armed Bandit Approach to Online Spatial Task Assignment

Umair ul Hassan
Insight Centre of Data Analytics
National University of Ireland
Galway, Ireland
umair.ulhassan@insight-centre.org

Edward Curry
Insight Centre of Data Analytics
National University of Ireland
Galway, Ireland
ed.curry@insight-centre.org

***Abstract*—Spatial crowdsourcing uses workers for performing tasks that require travel to different locations in the physical world. This paper considers the online spatial task assignment problem. In this problem, spatial tasks arrive in an online manner and an appropriate worker must be assigned to each task. However, outcome of an assignment is stochastic since the worker can choose to accept or reject the task. Primary goal of the assignment algorithm is to maximize the number of successful assignments over all tasks. This presents an exploration-exploitation challenge; the algorithm must learn the task acceptance behavior of workers while selecting the best worker based on the previous learning. We address this challenge by defining a framework for online spatial task assignment based on the multi-armed bandit formalization of the problem. Furthermore, we adapt a contextual bandit algorithm to assign a worker based on the spatial features of tasks and workers. The algorithm simultaneously adapts the worker assignment strategy based on the observed task acceptance behavior of workers. Finally, we present an evaluation methodology based on a real world dataset, and evaluate the performance of the proposed algorithm against the baseline algorithms. The results demonstrate that the proposed algorithm performs better in terms of the number of successful assignments.**

***Keywords*—spatial crowdsourcing, task assignment, multi-armed bandit**

I. Introduction

The popularity of crowdsourcing has encouraged the use of potentially large numbers of people for problem solving, in areas such as human computation [9], citizen actuation [4], and participatory sensing [13]. People contribute by performing tasks in either the virtual or the physical environments. *Spatial crowdsourcing* is a form of crowdsourcing that primarily deals with the tasks in physical environment. A spatial task requires the worker to travel to a specific location in order to perform it. There are three types of interacting agents in spatial crowdsourcing: *requesters*, *workers*, and the *platform*. A requester submits tasks with their associated locations to the platform. A worker receives and performs the spatial tasks, after registering with the platform. The platform serves as a mediator that provides services such as task-worker matching, quality control, and reputation management.

Consider a scenario in which a requester is interested in collecting high quality and representative photos of real life events. The events are being reported at various locations in a disaster hit region, as shown in Figure 1. The events are reported after irregular intervals, and the requester is interested in the coverage of all events. After an event is reported, the requester submits a corresponding spatial task. The task requires a worker to visit a specific location, take photos, and upload them. The platform assigns an appropriate worker to the task and notifies her. The worker either accepts or rejects the task depending on her situation. If accepted, the worker uploads the photos after some time. This scenario poses some specific challenges to the platform in terms of the worker assignment. The dynamic arrival of tasks means that the platform has no prior knowledge of the quantity and timing of tasks. The passive nature of worker interaction dictates that the workers do not actively visit the platform to seek tasks. The outcome of an assignment is stochastic; therefore, the platform must make assignment decisions under uncertainty.

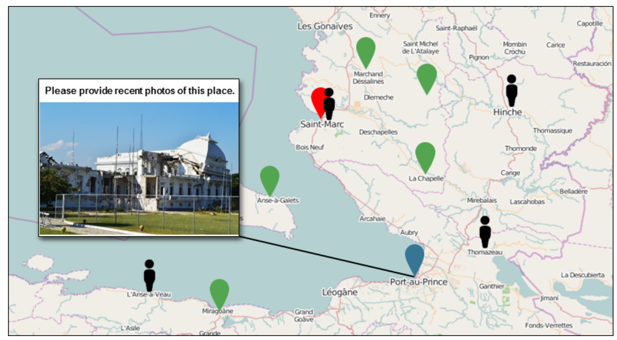


Fig. 1: Example of spatial crowdsourcing on the map of Haiti after the 2010 earth quake. A new spatial task (in blue) requests recent photos of a building at the indicated location.

Existing approaches have primarily focused on exploiting the spatial features of tasks and workers in the assignment process; either to maximize the quantity of assignments [12] or to maximize the success rate of assignments [19]. These approaches do not adapt the assignment process based on the outcome of previous assignments. Adaptive assignment approaches in other domains, such as crowdsourced classification tasks, do not consider the spatial context [10]. We consider the spatial task assignment as an online learning and optimization problem, by calling it the *online spatial task assignment* problem. The problem requires designing an algorithm for sequential assignment decisions. The objective of such an algorithm is to maximize the number of successful assignments over time, while learning from the outcomes of previous assignments. This posses an exploration-exploitation trade-off: the assignment algorithm must learn the task acceptance behavior by sampling different workers, while selecting the workers who have already shown willingness to accept.

The *online spatial task assignment* problem is closely related to the *multi-armed bandit* problem. In the multi-armed bandit model an agent simultaneously acquires knowledge about the available choices while making optimum selection decision among those choices. The contextual multi-armed bandit uses information from external sources for improving selection decisions; it has been successfully applied in domains such as online news recommendation, online ad placement, and adaptive packet routing [16]. Our proposed solution is inspired by this literature and applies it to the domain of spatial crowdsourcing. The specific contributions of this paper are summarized as follows:

- We present IMIRT (Individualized Models for Intelligent Routing of Tasks): a framework for online spatial task assignment. The framework motivates the need for online learning and optimization in spatial crowdsourcing. We provide a multi-armed bandit formalization of the assignment problem. We detail the exploitation-exploration trade-off for the assignment decisions, and describe the performance metrics for comparing different assignment algorithms.
- We propose an individualized linear model for learning task acceptance behavior of a worker, based on the observed outcome of previous assignments. We extend the multi-armed bandit algorithm with the proposed worker model and spatial information. The algorithm learns and exploits the worker preferences for routing tasks to workers with a higher likelihood of acceptance.
- We present an evaluation methodology to validate the proposed algorithms based on a real-world dataset. The goal of the experiments is to demonstrate the effectiveness of the proposed assignment algorithm, in terms of maximized success rate of assignments. The results suggests that the proposed algorithm does perform better than the baseline algorithms.

The rest of this paper is organized as follows. Section II provides an overview of the task routing challenge in spatial crowdsourcing. Section III describes the proposed IMIRT framework. Section IV presents the multi-armed bandit formalization of the online spatial task assignment problem. Section V describes the evaluation methodology and Section VI discusses the results of the experiment. Section VII summarizes the literature related to this work. Section VIII concludes the paper and discusses plans for future work.

II. Background

Intelligently matching tasks with the best workers is a fundamental challenge of crowdsourcing, also known as *task routing* [9], [8], [19]. The specific approach to task routing depends on factors such as worker expectation, worker interaction, number of tasks, and requester objectives. The majority of existing crowdsourcing platforms employ a *pull approach* to task routing, where workers self-assign the tasks through a search and browse interface [5]. The pull approach is prone to search friction and starvation issues [15], [8], possibly due to the inherent design of the interaction mechanism. Search friction arises when workers have difficultly finding the right tasks for themselves or vice versa. Starvation occurs when no worker chooses a task and eventually the task expires without completion. These issues can be addressed through the alternative *push approach* that algorithmically controls the task routing process [12], [19]. The push approach relies upon the knowledge about task attributes and worker characteristics to find suitable matches, while optimizing an objective function. We focus our attention on the push approach due to its compatibility with the requirements of our motivating scenario.

TABLE I: Comparison of existing frameworks for online assignment in crowdsourcing based on the push approach.

Framework	Optimization	Learning	Spatial	Decision
Ho & Vaughan [11]	Quality	Reliability		Choose Task
Kazemi & Shahabi [12]	Coverage		Yes	Find Match
Mehta & Panigrahi [17]	Coverage			Find Match
IMIRT	Coverage	Acceptance	Yes	Choose Worker

Table I compares the existing frameworks for the push approach of task routing. All of the frameworks assume an online setting for assignment decisions, where either tasks or workers arrive dynamically. The online setting of assignment is characterized by the sequential decision making under uncertainty. The knowledge about tasks and workers is revealed iteratively in the online setting, as opposed to offline setting in which case all information is available a priori. The specific characteristics of each framework are described as follows:

- The *online task assignment* framework assumes that workers actively visit the platform to request tasks [11], [10]. The assignment decision is made on the arrival of each worker. The objective of the assignment algorithm is to choose a task such that the possibility of a correct outcome is maximized. An adaptive algorithm based on primal-dual technique has been proposed for classification tasks [10]. Classification tasks require a worker to identify the correct categories for a set of items, such as images.
- The *online stochastic matching* framework assumes that the tasks and workers are nodes in a graph [17]. When a node arrives it must be matched with another node, where the stochastic outcome of matching is considered as the weight of the edge between the two matched nodes. The objective of the matching algorithm is to maximize the number of successful assignments.
- The *maximum task assignment* framework assumes that worker announce their availability and locations to the platform through mobile devices [12]. The assignment process runs after a fixed interval, during which the newly arrived tasks and worker are matched. The objective of the assignment algorithm is to maximize the number of matchings, while considering spatial features as well as the capacity constraints

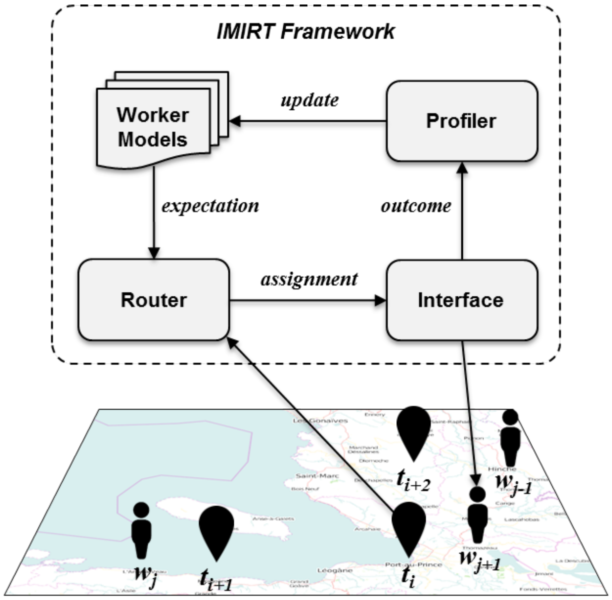


Fig. 2: The architecture of IMIRT framework; designed for online spatial task assignment in spatial crowdsourcing.

imposed by workers. A non-adaptive algorithm have been proposed by formalizing the assignment problem as the maximum flow problem [12].

Table I highlights that fact that none of the frameworks satisfies all the requirements of our motivation scenario. To address this gap we propose a new push-based framework for task routing, as detailed in the next section.

III. IMIRT Framework

We begin by describing the IMIRT framework that assumes the online spatial task assignment problem. Figure 2 gives an overview of the IMIRT framework and its main components. The framework assumes submission of tasks from requesters, through appropriate interfaces and protocols. The tasks arrive in an online manner, therefore the framework has no prior knowledge about the number of tasks arriving at any instance in time. Each new task t is transformed to a local representation and stored in the local database as a quadruple, $t = <\mathsf{desciption}, \mathsf{location}, \mathsf{expiry}, \mathsf{type}>$. The description is a textual attribute that lists the instructions to be followed for correctly performing the task. The location attribute defines the coordinates of the location associate with the task. The expiry attribute is a time-stamp defining the deadline for task completion, after which the task becomes invalid. The type attribute indicates the type of task.

The *profiler* component is responsible for managing the requisite information about all registered workers. The profiler maintains a separate profile for each worker w as the following triple, $w = <\mathsf{history}, \mathsf{model}, \mathsf{location}>$. The history attribute is a vector that stores the number of tasks assigned to the worker and the number of tasks accepted by the worker. The model is a set of the vectors that stores the variables specific to the task acceptance behavior of the worker. The location attribute stores the last reported location of the worker.

The primary function of the *router* component is to assign workers to the arriving tasks. For each task t, the router uses the worker profile to calculate the expectation of assignment success for each worker; then it chooses a worker[1] w such that the assignment success rate is maximized in the long term. Next, the router forwards the task t to the worker w and stores the assignment tuple $< t, w >$ in the local database.

The *interface* component resides on the worker's personal device, such as a mobile or a tablet. It receives the task from the router and generates an alert. The worker can then choose to accept or reject the assigned task. If the worker ignores a task without providing any explicit response then the task is considered to be rejected by default, after the expiry time lapses. An accepted task is rendered on the worker's device using dynamic forms. The worker performs the task by following the instructions and submits the response, that is then forwarded back to the requester. It should be noted that if a task is rejected before the deadline the router might re-assign the task as a new arrival.

A. Router Design

The fundamental challenge of the router is to balance the exploration-exploitation trade-off. The router could repeatedly select a worker which seems to have the best acceptance rate considering previous assignments, also known as *exploitation*. However, due to the uncertainty of knowledge about workers this seemingly best worker might be the suboptimal choice. Alternatively, the router might choose another worker for the purpose of *exploration*, hence deliberately making a potentially suboptimal choice. This exploration might adversely affect the task acceptance in the short term but the additional knowledge about workers can help improve assignment choices in the long-term. Understandably, neither a pure exploration nor a pure exploitation strategy can produce the best results. Therefore, a good assignment algorithm strikes the right balance between exploration and exploitation.

B. Performance Metrics

The passive nature of worker interaction in the IMIRT framework makes the success of assignment susceptible to external factors. For instance, it is possible that no worker is nearby the task location. A worker might be occupied with some other work. A worker might even ignore the task due to the communication problems e.g. delayed email, disabled notifications, etc. We define *worker success rate* (WSR) as the ratio of the number of tasks accepted and the number of tasks assigned to a worker. The primary metrics of the algorithm performance are

- ***Assignment Success Rate*** **(ASR)** is defined as the ratio of the number successfully accepted tasks against the total number of assigned tasks . We do not consider the quality of tasks performed, as a factor of assignment success, since it may not be directly observable due to the open and spatial nature of tasks.
- ***Average Travel Distance*** **(ATD)** is defined as the ratio of the sum of the distance traveled for all tasks against

[1] Assigning multiple workers to a task might also maximize the success rate; however, we focus on single task-worker assignment in this work

the total number of assigned tasks . This distance can also be considered as the cost of performing a task in case of paid crowdsourcing. However, we focus on the volunteered crowdsourcing hence distance is a secondary performance metric.

IV. Online Spatial Task Assignment

In this section, we describes the *online spatial task assignment* problem in further detail. Each task t is spatial by nature, that may require the worker to travel to a physical location. There is a fixed set of workers $\mathcal{W} = \{w_1, ..., w_m\}$, along with their last known locations. If the worker w_j is assigned to the task t_i, then the outcome of the assignment is a binary variable $y_{i,j}$ that indicates the worker's choice to accept or reject the task.

$$y_{i,j} = \begin{cases} 1 & \text{with probability } p_{i,j} \\ 0 & \text{otherwise} \end{cases}$$

The probability $p_{i,j}$ defines the likelihood of assignment success. A worker can be assigned to multiple tasks, meaning that the assignment algorithm can identify the best workers and choose them repeatedly. This raises the issue of overloading a worker with many tasks. It should be noted that the worker can choose to reject the tasks if already busy. We do not discuss the details of implementation for the sake of brevity.

A. Offline Formalization

We first consider a simplified offline setting of the assignment problem, to better our understanding. In this setting, the router has access to a set of tasks $\mathcal{T}$ and a set of workers $\mathcal{W}$, such that the number of task is $|\mathcal{T}| = n$, the number of workers is $|\mathcal{W}| = m$, and $n >> m$. Additionally, the outcome $y_{i,j}$ of each assignment is known a priori. The assignment algorithm must find an assignment of tasks and workers such that the maximum number of tasks are accepted. Let $a_{i,j}$ denote the variable that is 1 if worker w_j is assigned to the task t_i and 0 otherwise. We express the offline assignment problem as a linear program with the following objective function:

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} \sum_{j=1}^{m} y_{i,j} a_{i,j} && (1) \\ \text{s.t.} \quad & \sum_{j=1}^{m} a_{i,j} = 1 && \forall i \\ & a_{i,j} \in \{0, 1\} && \forall (i, j) \end{aligned}$$

This mathematical programming based formalization of the offline assignment problem can be solved optimally due to the totally unimodular constraint matrix; although, finding an optimal solution to integer programs is considered an NP-Hard problem in general. For instance, the *Hungarian method* can be used to find the solution in polynomial time [14].

B. Multi-armed Bandit Formalization

Now we formalize the online assignment problem, as inspired by the *multi-armed bandit* (MAB) framework [18]. The MAB assumes a player who wishes to play a slot machine that has multiple arms. The player chooses to play an arm and receives the reward in return. If arms are stochastic then the value of the reward depends on the distribution of reward for each arms. The primary goal of the player is to maximize her reward over a fixed number of rounds. Naturally, this poses an exploration-exploitation trade-off while choosing an arm to play.

procedure SpatialUCB($\alpha, \mathcal{W}$)
  // Initialization
  $m \leftarrow |\mathcal{W}|$
  for all $w_j \in \mathcal{W}$ **do**
    $\mathbf{A}_j \leftarrow \mathbf{I}_{2,2}$
    $\mathbf{b}_j \leftarrow \mathbf{0}_{2,1}$
  end for
  // Main loop
  for $i \leftarrow 1$ to n **do**
    Observe the distance vector $\mathbf{D}_i = (d_{i,1}, ..., d_{i,m})$
    Observe the type variable e_i
    for all $w_j \in W$ **do**
      $\mathbf{X}_{i,j} \leftarrow (d_{i,j}, e_i)^\top$
      $\hat{\boldsymbol{\beta}}_j = \mathbf{A}_j^{-1} \mathbf{b}_j$
    end for
    Choose worker $w \in \mathcal{W}$ that optimizes following

$$w_a = \arg\max_{w_j \in \mathcal{W}} \hat{\boldsymbol{\beta}}_j^\top \mathbf{X}_{i,j} + \alpha \sqrt{\mathbf{X}_{i,j}^\top \mathbf{A}_j^{-1} \mathbf{X}_{i,j}}$$

    Observe the outcome $y_{i,a}$ for worker w_a
    $\mathbf{A}_a \leftarrow \mathbf{A}_a + \mathbf{X}_{i,j} \mathbf{X}_{i,a}^\top$
    $\mathbf{b}_a \leftarrow \mathbf{b}_a + y_{i,a} \mathbf{X}_{i,a}$
  end for
end procedure

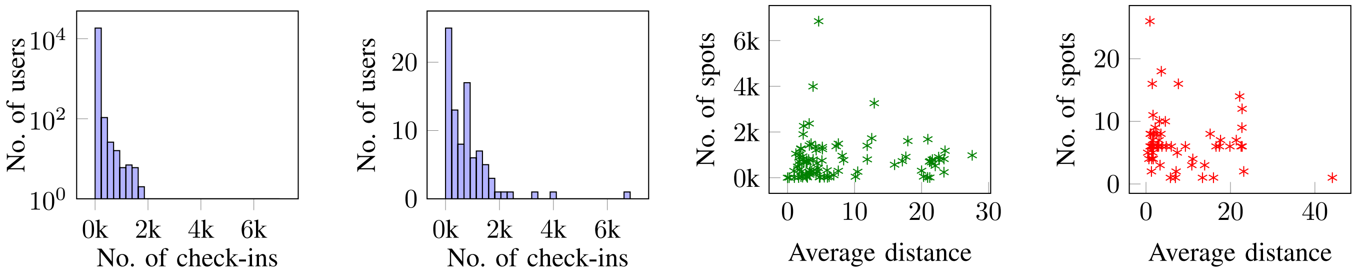
Fig. 3: The SpatialUCB algorithm.

In our formalization, the pool of workers W is considered the multi-armed bandit such that each arm corresponds to a worker. An assignment $a_{i,j}$ is equivalent to playing an arm, where as the outcome of assignment $y_{i,j}$ is the resulting reward. For a dynamically arriving task, choosing a worker with the highest expectation of $y_{i,j}$ leads to the maximization of the assignment success rate. As opposed to the offline setting, knowledge about tasks and workers is revealed on task arrival in the online setting. The spatial features of task and workers are revealed before assignment and the outcome of the assignment is revealed afterwards.

The standard form of multi-armed bandit problem assumes a fixed number of rounds, however we consider the case where the number of incoming tasks is unknown. The standard form does not consider any additional information, during each round. Whereas our formulation is inspired by the *contextual bandit* model [16], that includes external information into the assignment process. For instance, the location attributes of task and worker can be considered the contextual information that might be useful for making assignment decisions.

The assignment process proceeds in discrete iterations $i \in \{1, 2, 3, ..., n\}$, where each iteration i is performed at the time of task arrival. The following steps are performed during each iteration:

1) The algorithm considers the current task t_i and the current pool of workers $\mathcal{W}$ together with a set of



(a) All users (b) Top 90 users (c) Check-ins (d) Highlights

Fig. 4: Distributions of users against check-ins in the Gowalla dataset and comparison of distance with check-ins and highlights.

vectors $\mathbf{X}_{i,j}$'s. Each vector $\mathbf{X}_{i,j}$ contains the features defined according to the spatial attributes of the task and workers.

2) The algorithm chooses a worker $w \in \mathcal{W}$ for the current task t_i, based on the spatial contextual and the observed outcome of the previous iterations. For each assignment the algorithm observes the outcome $y_{i,j} \in \{0, 1\}$, which depends on the current task and the chosen worker.
3) The algorithm improves its assignment strategy based on the outcome of the current assignment. The algorithm does not observe any information from the workers that are not chosen for assignment i.e. $w \neq w_j$, which is a fundamental assumption of the multi-armed bandit.

In the above process, the assignment success rate of an assignment strategy after I iterations is defined as

$$ASR(I) = \frac{1}{I} \sum_{i=1}^{I} y_{i,w} \tag{2}$$

where $y_{i,w}$ is the outcome observed only for the assigned worker. Similarly, the expected optimal success rate $ASR^*(I)$ is defined according to the outcome y_{i,w^*} for the worker w^* chosen by the optimal offline solution. The *regret* after I iterations is defined as

$$R(I) = ASR^*(I) - ASR(I) \tag{3}$$

The assignment algorithm should employ an assignment strategy which minimizes the *regret*. Alternatively, our goal is to design an algorithm such that the $ASR(I)$ is maximized.

C. Assignment Algorithm

The multi-armed bandit algorithms have been widely studied under the assumptions of stochastic arms [1]. It is assumed that the reward for each arm is independent and identically distributed, according to some known statistical distribution[2]. Optimal arm selection strategies have been proposed under Markov setting [6], which are considered to be computationally expensive. Alternatively, most of the applied research work has focused on approximate solutions [20]. We highlight some of the most widely used algorithms employing approximation strategies.

Algorithms based on semi-uniform strategies alternate between exploration and exploitation phases. The most simplest algorithm within the semi-uniform strategies is the $\varepsilon-$greedy algorithm, which selects the best arm during $1-\varepsilon$ proportion of the iterations and a random arm is selected during ε proportion [20]. Probability-matching strategies select arms based on the probability distribution that measures the likelihood of each arm being close to the optimal arm, for instance the SoftMax algorithm [20]. Another group of algorithms calculate the confidence interval on the expectation of reward; then select the arm with the highest upper confidence bound, hence known as UCB algorithms [1], [20]. All of the discussed strategies rely entirely on the observed rewards of previous played arms for the next iteration, therefore they are categorized as the *context-free* algorithms.

We emphasize that the main source of uncertainty in online spatial task assignment is the outcome of assignment $y_{i,j}$. An assignment algorithm could follow a simplistic approach by assuming that each worker has fixed behavior of task acceptance, i.e., $p_{i,j} = p_{i',j}$ for any i and i'. In such a case, the success rate of each worker can be modeled as a Binomial process with parameter p_j. Given the spatial contextual $\mathbf{X}_{i,j}$ for the task t_i and worker w_j, we exploit this information to improve the assignment algorithm. We assume that the task acceptance behavior, of each worker, is linear over spatial contextual vector. Therefore, the expectation of assignment success is defined as below

$$\mathbf{E}[y_{i,j}|\mathbf{X}_{i,j}] = \boldsymbol{\beta}_j^\top \mathbf{X}_{i,j} \tag{4}$$

The vector $\boldsymbol{\beta}_j$ defines the unknown worker specific coefficients that need to be learned from observed outcomes. On one hand, the objective of the assignment algorithm is to choose a worker with highest expectation of $y_{i,j}$. On the other hand, the algorithm must learn the coefficient in $\boldsymbol{\beta}_j$ for each worker by assigning tasks to unexplored workers. One such algorithm is the LinUCB algorithm, that uses ridge regression over observed data for learning the coefficients on the linear model [16]. We adapt the LinUCB algorithm with two variables from the spatial context. The first is a numerical variable $d_{i,j} \in \mathbb{R}^+$ that quantifies the distance between the current task t_i and a worker w_j. The second a categorical variable e_i that indicates the type of task. We define the spatial context vector

[2]Adversarial arms do not make any assumptions of the reward distribution; however, adversarial workers are not in scope of this work

as $\mathbf{X}_{i,j} = (d_{i,j}, e_i)^\top$ and propose the SpatialUCB algorithm, as shown in Figure 3. The parameter α controls the intensity of the confidence bound. A reasonable value of α can be found with the help of domain specific experimentation.

V. Evaluation Methodology

In this section, we present the details of an evaluation methodology based on a real-world dataset. There is a general lack of openly available datasets that describe the behavior of workers in spatial crowdsourcing scenarios. More importantly, the large scale deployment of prototypes for the purpose of experimentation is prohibitively expensive and time consuming. Existing research has generally circumvented this issue by adopting datasets from location-based social networks, for the purpose of evaluating spatial task assignment [19], [5], [12]. We follow a similar approach to evaluate the performance of various assignment algorithms. First we describe our dataset and the preprocessing steps of preparing it for the evaluation. Then, we list the algorithms compared during the experiments.

A. Dataset

We used a real-world dataset based of a location-based social network; Gowalla. This dataset contains data about people who have voluntarily reported their visits to various locations. The data was collected for locations in the New York city during October 2011. The dataset contains 19,183 users, 30,367 spots, 2,767 highlights, and 357,753 check-ins. Each *spot* is a geo-referenced location in New York city. A *highlight* represents a particular tag associated to a spot by a user. A *check-in* represents the visitor relationship between a user and a spot.

Similar to the existing literature [12], [19], we use the dataset to simulate a spatial crowdsourcing scenario. We assume that the users in the dataset are the volunteering crowd workers. We also consider all unique spots in check-ins data as spatial tasks, requiring physical travel. A check-in indicates that worker has traveled to the location and performed the task. Furthermore, we consider all unique spots in highlights as a spatial tasks which do not necessitate actual travel. A highlight indicates that a worker has performed the task without travel. Finally, we consider the last spot visited by each worker as her current location. As discussed earlier, we consider two spatial features i.e. distance and task type. The type feature is represented as a binary indicator variable, 1 for check-in and 0 for highlight. The distance feature is based on the Euclidean distance between coordinates of the task and the worker.

We analyzed the check-in and highlight behavior of users. Figure 4a shows the distribution of the number of check-ins by each user on a logarithmic scale. The distribution shows the Zipf's law behavior for the number of unique check-ins by a user; the majority of users have very low activity. This behavior is commonly observed in various physical and social phenomena. We excluded the long tail of low activity users by selecting top ranked users based on their check-ins and highlights. The resulting dataset had 90 users with relatively high levels of activity. The distribution of check-ins, for selected users, is shown in Figure 4b. This group of users forms the pool of worker in our spatial crowdsourcing scenario. A worker's stochastic behavior of task acceptance is modeled by the corresponding check-ins and highlights [19].

We further analyzed the relationship between the distance of a spot from users with their check-ins and highlights. Figure 4c shows the number of check-ins against the average distance from user's current spot. Clearly, the check-in behavior varies across users. Some users have higher number of check-ins with in 5-10 kilometers, while other users have visited spots as far as 25 kilometers away. Conversely, there are users who visit very small number of spot irrespective of the distance. The average distance shows a negative correlation with the number of check-ins. Overall this behavior is representative of worker dynamics in spatial crowdsourcing; more tasks tend to be completed in the near vicinity of workers [12]. The behavior for highlights is similar for all users, as shown in Figure 4d. Again the majority of users are clustered around the bottom left corner to indicate the majority of highlights are in the near vicinity of workers.

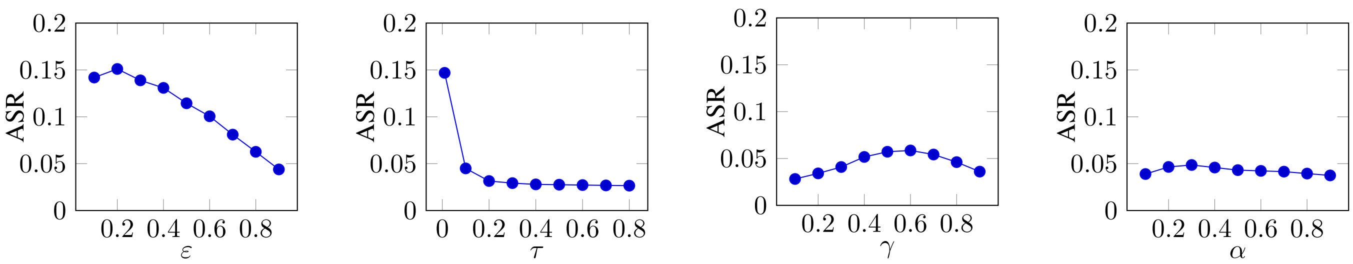
B. Compared Algorithms

We compared the following algorithms during the experiments to evaluate their relative performance:

- $\varepsilon-$greedy: This algorithm chooses a random worker with probability ϵ, otherwise the worker with the highest WSR is chosen. Each worker's WSR is calculated from history of previous assignments [1]. The parameter ε controls the rate of exploration. High values of ε result in high exploration.
- Softmax: This algorithm chooses a worker with a probability that is proportional to its previous WSR [2]. It takes τ parameter, called the *temperature*, that moderates the degree to which the worker with the highest expectation is chosen. Higher values of τ means more exploration of workers.
- Exp3: This algorithms is a variant of Softmax algorithm. The parameter γ controls the degree of probability matching [3]. The value of parameter $\tau = 1$ means all workers are equally likely to be chosen. Generally it starts with a pure exploration phase where a random worker is chosen for first γ tasks, and starts to match the distribution of worker WSR over time.
- UCB2: This algorithm chooses a worker who has the highest upper confidence bound the on expected success rate of worker. [1]. The confidence bound is a function of the worker success rate. The parameter α controls the intensity of the confidence bound.
- SpatialUCB: This algorithm chooses a worker based on a linear function defined over spatial contextual features. The parameter α for SpatialUCB behaves in a similar way as the UCB2 algorithm [1].

VI. Experiments & Results

In this section, we discuss a set of experiments that evaluate the effectiveness of the assignment algorithms. The experiments utilize various properties of the Gowalla dataset to generate representative spatial crowdsourcing scenarios. We perform two experiments to analyze the effects of various parameters on the performance of the algorithms.



(a) $\varepsilon-$Greedy Algorithm (b) Softmax Algorithm (c) EXP3 Algorithm (d) UCB2 Algorithm

Fig. 5: Results of context-free assignment algorithms for 5000 tasks, with workers simulated as Binomial stochastic processes.

TABLE II: Comparison of algorithms for 5000 tuning tasks.

Assignment Algorithm	ASR	ATD
Softmax, $\tau = 0.01$	**0.09204**	18.52864
Softmax, $\tau = 0.1$	0.020880	15.950382
$\varepsilon-$Greedy, $\varepsilon = 0.2$	**0.06326**	17.058638
$\varepsilon-$Greedy, $\varepsilon = 0.3$	0.06232	18.604301
SpatialUCB, $\alpha = 0.2$	0.1465	16.28432
SpatialUCB, $\alpha = 0.5$	**0.1465**	15.85458
SpatialUCB, $\alpha = 1$	0.09814	13.858756
SpatialUCB, $\alpha = 2$	0.06608	14.271388

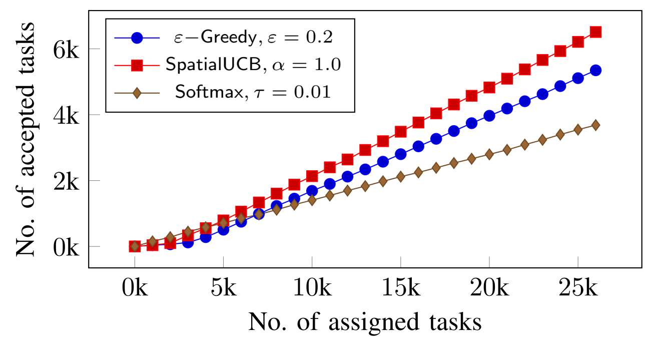


Fig. 6: Comparison of algorithms on 26662 test tasks.

A. Assignment without Spatial Context

The first experiment compares the performance of assignment algorithms without spatial context. We simulated the task acceptance behavior of workers as Binomial process with parameter p_j. The parameter p_j was set according to the WSR of all 90 workers in the dataset. We ran each of the algorithm for 5,000 simulated tasks without any spatial contextual. We report the average ASR for 10 runs of each algorithm.

Figure 5 shows the comparison of algorithm parameter values against the ASR. The results indicate that the $\varepsilon-$greedy and Softmax algorithms achieve the highest ASR level. The $\varepsilon-$greedy algorithm achieves the highest ASR at small values of ε. Similarly, the highest ASR for Softmax is at $\tau = 0.01$. These results suggest that high exploration has negative effects on the success rate. The better performing algorithm were quickly able to identify the small set of workers with higher acceptance level. By comparison, the optimistic exploration strategy of the UCB2 algorithm resulted in necessary selection of workers with lower acceptance level. We limited the context-free algorithms to only $\varepsilon-$greedy and Softmax, for the later experiments, with parameter values of $\varepsilon \in \{0.2, 0.3\}$ and $\tau \in \{0.01, 0.1\}$.

B. Assignment with Spatial Context

In the second experiment, we evaluated the performance of assignment algorithms with spatial context. Typically in real life deployments the assignment algorithms are tested offline with data sampled from running system, instead of a live deployment. Therefore, we first fine tuned the parameter values of the context-based assignment algorithm on a sample of tasks. We sampled a random set of 5,000 tasks from the dataset. We ran each algorithm 10 times and report the ASR values for each algorithm. We compared the SpatialUCB algorithm with the algorithms selected in previous the experiment. The parameter value of SpatialUCB were set according to the commonly found values in the literature i.e. $\alpha \in \{0.2, 0.5, 1, 2\}$. Table II lists the resulting ASR on the tuning tasks. The results indicate that the SpatialUCB performs better as compared to the other algorithms; both it terms of the assignment success rate and the average travel distance. This highlights the fact that the SpatialUCB was effective in learning the linear relationship between contextual information and the task acceptance behavior of a worker.

We compared the performance of tunned algorithms on the rest of the tasks in dataset. Figure 6 shows the performance of $\varepsilon-$Greedy, Softmax, and SpatialUCB algorithm with tuned parameters. We ran each algorithm 10 times on 26,662 tasks and report the number of completed tasks after each round of assignments. The results show that the number of completed tasks increases linearly with the number of assigned tasks. The tuned SpatialUCB algorithm performs consistently better that other algorithms. The difference in the performance of algorithm is not in multiple orders of magnitude.

VII. Related Work

Kazemi and Shahbi [12] proposed the maximum task assignment framework for spatial crowdsourcing. The objective of their framework is to maximize the coverage of tasks i.e. the number of tasks matched with workers. To et al. [19] defined at privacy enabled framework for task assignment in spatial crowdsourcing. Their framework is to hide the actual locations of workers while maximizing the success rate of assignments. Deng et al. [5] propose approximation algorithms for scheduling task for pull based approach to spatial crowdsourcing. In comparison, our framework aims to maximize the success rate of assignments, by adapting the assignment strategy according to the outcome of previous assignments. Our framework is differentiated from these works due to two reasons: our framework uses individualized task acceptance models for workers and the assignment strategy is adaptive with respect to observed outcomes of assignment.

Matching between tasks and workers has been an active area of research among crowdsourcing and human computation. Ho and Vaughan [11] formalized the task assignment problem in online setting for heterogeneous tasks in crowdsourcing markets. They also proposed an adaptive algorithm that is competitive to the offline version of assignment algorithm [10]. Hassan and Curry proposed a performance prediction based approach for push routing of tasks, based on a Bayesian approach for worker capability modeling [8]. The primary focus of these works have been the quality of crowdsourcing tasks and reliability of workers, specifically for classification tasks. By comparison, we focus on the willingness of workers to perform spatial tasks.

VIII. Conclusion & Future Work

In this paper, we introduce the *online spatial task assignment* problem within spatial crowdsourcing. In our setting, the tasks arrive dynamically in an online manner and a worker is assigned to each task. We proposed the IMIRT framework that formulates the online spatial task assignment as the multi-armed bandit problem. The objective of the framework is to maximize the number of tasks accepted by assigned workers. We include the spatial features of tasks and workers to improve the assignment algorithm. We model the task acceptance behavior, of a worker, as a linear function of the spatial features. We presented a contextual bandit algorithm based on the spatial features and evaluate it with a real-world dataset. The results suggest that the algorithm performs best in terms of the cumulative success rate of assignments

This work extends the existing research within spatial crowdsourcing in several key ways. It provides new insights into assignment strategies where the assumption of task acceptance by workers is relaxed. The workers are not considered to be actively searching for the tasks on the crowdsourcing platform; instead, they passively receive tasks chosen by an assignment algorithm. It provides evidence for the effectiveness of assignment strategies for stochastic acceptance behavior. The current formulation of online worker assignment assumes one worker assigned per task. We plan to extend this work with multiple workers assignment per task. We also plan to improve the contextual algorithm by exploiting spatial density of workers around a task. Finally, we plan to explore its effectiveness within a smart environment [7].

Acknowledgments

This work has been supported in part by the Irish Research Council under the New Foundations Scheme, the Enterprise Ireland under the National Development Plan, and the European Union under the grant number IP/2012/0188.

References

[1] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[2] Andrew G. Barto. *Reinforcement learning: An introduction*. MIT press, 1998.

[3] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

[4] David N. Crowley, Edward Curry, and John G. Breslin. Closing the loop: From citizen sensing to citizen actuation. In *Proceedings of the 7th IEEE International Conference on Digital Ecosystems and Technologies*, pages 108–113. IEEE, 2013.

[5] Dingxiong Deng, Cyrus Shahabi, and Ugur Demiryurek. Maximizing the number of worker's self-selected tasks in spatial crowdsourcing. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 314–323. ACM, 2013.

[6] John Gittins, Kevin Glazebrook, and Richard Weber. *Multi-armed Bandit Allocation Indices*. John Wiley & Sons, 2011.

[7] Umair Ul Hassan, Murilo Bassora, Ali H. Vahid, Sean O'Riain, and Edward Curry. A collaborative approach for metadata management for internet of things. In *Proceedings of the 9th International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 593–598. IEEE, 2013.

[8] Umair Ul Hassan and Edward Curry. A capability requirements approach for predicting worker performance in crowdsourcing. In *Proceedings of the 9th International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 429–437. IEEE, 2013.

[9] Umair Ul Hassan, Sean O'Riain, and Edward Curry. Effects of expertise assessment on the quality of task routing in human computation. In *Proceedings of the 2nd International Workshop on Social Media for Crowdsourcing and Human Computation*, pages 1–10. BCS, 2013.

[10] Chien-Ju Ho, Shahin Jabbari, and Jennifer W Vaughan. Adaptive task assignment for crowdsourced classification. In *Proceedings of the 30th International Conference on Machine Learning*, pages 534–542, 2013.

[11] Chien-Ju Ho and Jennifer Wortman Vaughan. Online task assignment in crowdsourcing markets. In *AAAI Conference on Artificial Intelligence*, 2012.

[12] Leyla Kazemi and Cyrus Shahabi. Geocrowd: enabling query answering with spatial crowdsourcing. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 189–198. ACM, 2012.

[13] Aniket Kittur, Jeffrey V. Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. The future of crowd work. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, pages 1301–1318. ACM, 2013.

[14] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[15] Anand Kulkarni, Philipp Gutheim, Prayag Narula, David Rolnitzky, Tapan Parikh, and Björn Hartmann. Mobileworks: Designing for quality in a managed crowdsourcing architecture. *Internet Computing, IEEE*, 16(5):28–35, 2012.

[16] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pages 661–670. ACM, 2010.

[17] Aranyak Mehta and Debmalya Panigrahi. Online matching with stochastic rewards. In *IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 728–737. IEEE, 2012.

[18] Herbert Robbins. Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*, pages 169–177. Springer, 1985.

[19] Hien To, Gabriel Ghinita, and Cyrus Shahabi. A framework for protecting worker location privacy in spatial crowdsourcing. *Proceedings of the VLDB Endowment*, 7(10), 2014.

[20] Joannes Vermorel and Mehryar Mohri. Multi-armed bandit algorithms and empirical evaluation. In *Machine Learning: ECML 2005*, pages 437–448. Springer, 2005.