A Multiarmed Bandit Incentive Mechanism for Task Allocation in Mobile Crowdsensing

***Abstract*—Recent years, we have witnessed the emergence of mobile crowd sensing, which leverages the crowd equipped with various mobile devices for large scale sensing tasks. We propose a novel multiarmed bandit mechanism for sensing tasks which makes monetary offers to strategic users who have unknown service qualities, to incentivize user participation. In this paper, we approach this worker selection problem in a distributed online learning manner, where the qualities of the users' service are estimated by the current contexts. We design an effective online incentive algorithm TS-TA using Thompson Sampling according to the current contexts so that the learning algorithm updates over time and learns to choose the most effective users to finish the specific tasks. We prove that our algorithm has a regret of $O(d \ln T \sqrt{T \ln T \ln(\frac{1}{\delta})})$ uniformly in time under mild assumption on the users' qualities, as well as dominant strategy incentive compatible and individually rational.**

I. Introduction

Mobile crowdsensing has been an emerging paradigm which involves the collaboration of a crowd of mobile users who use their smart phones to execute complex computation or sensing tasks. Significant interests in mobile phone sensing can be attributed to several factors, the mobility of users to solve large-scale mobile sensing tasks; rapid evolution toward smart phones with several built-in sensors; and the possibility of leveraging the cloud via several available connectivity options for computing power and storage. So it has stimulated a number of attractive applications, such as urban traffic information mapping [1], WiFi characterization [2], labeling massive datasets [3], health monitoring [4]. Moreover, there have been some platforms, frameworks, and incentive mechanisms, designed for crowd sensing [5], [6].

Consider that a user carrying mobile phone in an mobile social network has some independent sensing tasks. Usually, these tasks cost too much time and exceed its processing ability. Then, it asks other mobile users for help by starting a crowd sensing. The user who has some tasks need to be allocated is called requester, and the user who finishes these tasks is called worker. In this scenario, the requester sends tasks to workers by calling or sending messages. And it often involves large size data transmissions, however, we can incent other users joining it, to decrease every user's work time. We consider a communication technology which is different with [5], which adapts short-distance wireless communication technologies (e.g., WiFi or Bluetooth). In contrast, we consider the requester gets in touch with workers through cellular networks regardless the distance between the requester and workers. In this situation, the requester doesn't have to stay close to the workers. Fig.1 depicts a simple example of a task allocation procedure in crowdsensing. At some time, the requester wants to collect some data, and then it sends message to the workers. After the workers receive the request, they will return their bids and workloads to the requester, then the requester makes a decision to choose a best worker set to handle these tasks. The best worker means he will finish the work very well (also called service quality) and at the same time the bid is reasonable. Then, an important problem is how the requester allocates these tasks to workers, where the workers' qualities are unknown.

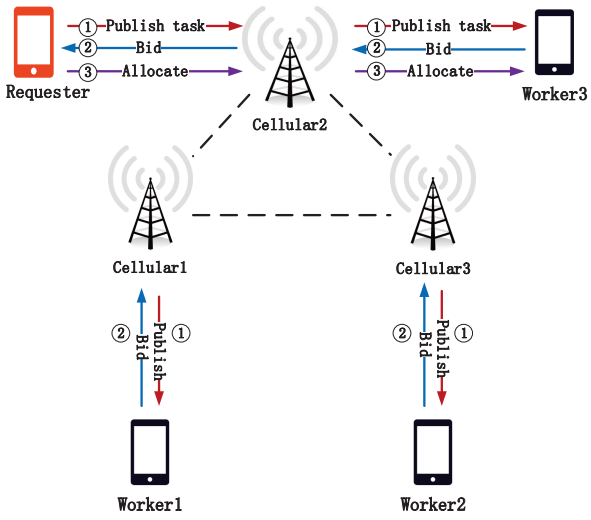

Fig. 1. An Example of Task Allocation in CrowdSensing

Another key issue in enabling large-scale mobile crowdsensing applications is to guarantee that mobile device users are willing to contribute their resources in a participatory or opportunistic way. To stimulate each mobile user to participate in mobile crowdsensing, the research community has recently developed some incentive mechanisms for mobile crowdsensing [7], [8], [9], [10]. The main idea is that for each user who participates in crowdsensing, some reward (in a monetary form) is given to the user depending on the amount of sensing time that it contributes. The incentive mechanisms are carefully designed, so that it is always to the best interest of the users to participate cooperatively and the system achieves equilibrium.

In this paper, we focus on the above task allocation problem and the incentive mechanism. We aim to choose a best worker

from the requester's contacts to finish a task. We propose to selects best worker which is like choosing excellent contractor, to maximize the utility of the requester. And the incentive mechanism guarantees that the worker can obtain benefit from it.

To solve the above task allocation problem in mobile crowdsensing, we propose a multi-armed bandit based algorithm with incentive mechanism, which is called Thompson Sampling Task Allocation (TS-TA). The major difference of our algorithm compared with the existing work is to formulate worker's unknown service quality through the workers' current contexts, which are related to their own status.

Thompson Sampling (TS) is one of the earliest heuristics for multi-armed bandit problems. The first Bayesian heuristics is around 80 years old, dating to Thompson (1933). Since then, it has been rediscovered numerous times independently in the context of reinforcement learning. It is a member of the family of *randomized probability matching* algorithms. The basic idea is to assume a simple prior distribution on the underlying parameters of the reward distribution of every arm, and at every time step, an arm is played according to its posterior probability of being the best arm.

In each round, TS plays an arm according to its posterior probability of having the best parameter. A simple way to achieve this is to produce a sample of parameter for each arm using the posterior distribution, and play the arm that produces the best sample. Our major contributions in this paper are as follows:

1) We formulate the problem into a linear payoffs MAB problem that shall find optimal workers when there are tasks need to be allocated. Then we propose a novel algorithm, Thompson Sampling Task Allocation (TS-TA) which guarantees the workers are truthful and rational.
2) We define a new standard to choose workers not only according to their bids, which is a traditional way in crowdsourcing, but also, we define the workers' service qualities through workers' contexts, to achieve a high utility of requester. We prove that the regret has a bound, where any MAB algorithm has to suffer.
3) We conduct our simulations on real-world datasets to evaluate the TS-TA algorithm. The results show that the proposed algorithm have a bound of $O(d \ln T \sqrt{T \ln T \ln(\frac{1}{\delta})})$.

The remainder of the paper is organized as follows. We present a summary of the related work in Section II. In Section III, we formulate the task allocation problem. In Section IV, we propose our algorithm based Thompson Sampling for task allocation to choose best workers to finish current works and then decide how much payment to the workers. In Section V, we theoretically prove our algorithm's regret is bounded and prove our mechanism is individually rational, truthful, and computational efficient. In Section VI, we carry out extensive simulations based on real-world datasets to verify our algorithm. Finally, we conclude the work in Section VII.

II. RELATED WORK

A. Task allocation

In this paper, we focus on the task assignment problem in mobile crowdsensing, which involves a lot of mobile users to perform a large number of tasks through their carried mobile devices [11]. By far, there have been several frameworks, platforms, and incentive mechanisms designed for crowdsensing systems [12], [2], [13], [14], [11]. LRBA ([14]) designs a task allocation problem in mobile crowdsensing to choose the worker who will provide better service and the tasks are location dependent. The most related work among existing work is [3], which considers a crowd-sourcing problem where in the process of labeling massive datasets, multiple labelers with unknown annotation qualities must be selected to perform the labeling task. And they formulate it to a MAB problem, then propose an algorithm to solve it. In our work, we consider the context of each worker, and take the payment of workers into account. We aim to pick up a proper worker to finish a specific task, and make sure the worker accomplishes the task with good quality, at the same time the worker's bid should be acceptable. Meanwhile, the utility of requester is also acceptable.

B. MAB Algorithms

The fundamental challenge in multi-armed bandit problems is the need for balancing exploration and exploitation. Changing the aim of maximizing the reward to minimizing the regret of multiple experiments, an algorithm will exploit its past experience to select the arm that appears best. On the other hand, this seemingly optimal arm may in fact be suboptimal, due to imprecision in the algorithm's knowledge. In order to avoid this undesired situation, the algorithm has to explore by actually choosing seemingly suboptimal arms to gather more information about them. Exploration can increase short-term regret since some suboptimal arms may be chosen. However, obtaining information about the arms' average payoffs can refine the algorithm's estimate of the arms' payoffs and in turn reduce long-term regret. Clearly, neither a purely exploring nor a purely exploiting algorithm works best in general, and a good tradeoff is needed.

There are a lot of literatures available on the MAB problem. The context-free K-armed bandit problem has been studied by statisticians for a long time [15], [16], [17]. One of the simplest and most straightforward algorithms is ϵ-greeedy. In each round t, this algorithm first estimates the average payoff of each arm a according to the history data. Then, it selects the arm who has the highest payoff estimate with probability 1-ϵ, and selects another arm with ϵ. In contrast to the unguided exploration strategy adopted by ϵ-greedy, another class of algorithms are generally known as upper confidence bound algorithms [18], [19], [20] which estimate both mean payoff of each arm a and a corresponding confidence interval, so that the arm which achieves a highest upper confidence bound is selected.

While context-free K-armed bandits are extensively studied, the more general contextual bandit problem has remained challenging. The EXP4 algorithm uses the exponential weighting technique to achieve an $O(\sqrt{T})$ regret, but the computational complexity may be exponential in the number of features. Another general contextual bandit algorithm is the epoch-greeedy algorithm, which is similar to ϵ-greedy with shrinking ϵ. This algorithm is computationally efficient given an oracle optimizer but has the weaker regret guarantee of $O(T^{2/3})$.

TABLE I
PARALLEL BETWEEN MULTI-ARMED BANDIT AND TASK ALLOCATION

Multi-armed Bandit (MAB)	Task Allocation (TA)
Player	Requester
Armed bandits	Workers
Pull level	Select worker
Reward	Finish task successful or failure

III. PROBLEM FROMULATION

The goal in our paper is to find all eligible workers that provide the highest service qualities when allocating tasks to them. One way to model is to represent workers using a set of features and build a classification model. However, due to the sparsity and inaccuracy of data, the classification model may be not effective enough. Hence, we use the context to make decision. We use feature space to illuminate the context. The feature space is constructed along two dimensions: task and worker. Let $W = \{w_1, w_2, ..., w_k\}$ represent the worker set, let $TA = \{task_1, task_2, ..., task_m\}$ represent the task set. Let $c^t_{i,j}$ denote the context at time t, where i, j represent worker i and task j separately, $i = 1, ..., k; j = 1, ..., m$. At each round, we get an context vector of $(task \times worker)$.

Due to the dynamic nature of the problem and time-varying performance of the workers, we formulate online worker selection problem as a MAB problem, where the set of available workers changes over time. The TABLE I shows the parallel between our task allocation problem and MAB problem. The reason behind this is two-fold: First, through an online evaluation we have observed that based on specific contexts, the high quality service provider can be selected with high probability; Secondly, the performance varies significantly over time.

Choosing the best worker for a task can be naturally viewed from the perspective of exploration-exploitation trade-off. The concept of exploration-exploitation is central to the problem in decision making under uncertainty, and is best illustrated by MAB problem. A bandit problem consists of k probability distributions $\{D_1, D_2, ..., D_k\}$, which represent the reward distribution of every bandit, and the expected value can be illustrated by $\{\mu_1, \mu_2, ..., \mu_k\}$. At the beginning, the distributions are unknown to the player. The algorithm is regarded as a gambler whose goal is to get as much money as possible by pulling these arms over rounds. At each round, $t = 1, 2, ...$, the player selects an arm, with index a_t, and receives a reward $y_t \sim D_i(t)$. The goal of the algorithm is to find out which distribution has the highest expected value, and also to gain as much reward as possible during the game.

TABLE II
NOTATION USED

Notation	Description
k	Number of requester's contacts
m	Number of tasks
$c_i(t)$	Context of arm i at time t
μ	The prior parameter of arm i
$r_i(t)$	The reward of arm i at time t
$R(T)$	Total regret
$p_i{}^t$	Payment of arm i
$b_i, \hat{b}_i$	Actual and reported bid of worker w_i
W	The requester's worker set
W_s	The selected worker set
w_i, w_{-i}	Worker i and the worker set without i
$TA(t)$	The task set at time t
H_{t-1}	The history data

The reason of using bandit algorithm in crowdsensing to select proper workers is that, we assume that the bandit algorithm has access to a set of workers, whose performance can change over time, and for each round t the bandit has to pay a payment to the worker who wins the auction. Hence, each trial of selecting the winning worker consists of the following steps:

1) The requester constructs his worker set W, and sends a task request to all his workers.
2) Each available worker provides his current context $c_i(t)$ to the requester.
3) The algorithm chooses one worker from W according to the context matrix, and updates the distribution of every worker.
4) The requester observes the reward of the selected worker, i.e. $r_t \in \{0, 1\}$.
5) If $r_t = 0$, the algorithm runs again, until $r_t = 1$.
6) The requester decides the payment of the winning worker and gives the payment.

Note that, in step 4, we point out the reward belongs to $\{0,1\}$. If the worker successfully accomplishes the task, then the reward is 1, otherwise the reward is 0. We use a variable to represent the different ability of a worker to accomplish a task. And this is the user's private information. Our algorithm aims to pick up the workers with high service quality.

Various strategies have been proposed to choose the right winner to accomplish a task in crowdsensing which just care about its bid. But very few consider the situation where every worker is different, which also means that the service quality of every worker is different.

IV. MULTI-ARMED BANDIT INCENTIVE MECHANISM

A. Worker Selected based on Thompson Sampling

There are K workers, all of whom are regarded as arms. At time $t = 1, 2, ...$, a context $c_i(t) \in \mathbb{R}^d$, is revealed for every worker i. These context vectors are chosen by the requester in an adaptive manner after observing that which workers have

finished some tasks and their rewards up to time $t-1$. We denote history H_{t-1} as follows,

$$H_{t-1} = \{w(\tau), r_{w(\tau)}(\tau), c_i(\tau), i = 1, ..., K, \tau = 1, ..., t-1\}, \tag{1}$$

where $w(\tau)$ means that the worker is chosen at time τ, $r_{w(\tau)}(\tau)$ denotes the reward when choosing worker w. We denote $c_i(t)^T\mu$ the mean reward for arm i at time t, given $c_i(t)$, where $\mu \in \mathbb{R}^d$ is a fixed but unknown parameter [21], thus we have

$$E[r_i(t)|\{c_i(t)\}_{i=1}^K, H_{t-1}] = E[r_i(t)|\{c_i(t)\}] = c_i(t)^T\mu. \tag{2}$$

Our Algorithm for contextual bandit problem needs to select a worker $w_i(t)$ to play, using H_{t-1} and current context matrix. Let $w_*(t)$ denote the optimal worker at time t, i.e., $w_*(t) = \arg\max \frac{c_{w_i}(t)^T\mu}{b_{w_i}}$. Then the regret, i.e., the difference of the optimal worker and the selected worker at time t is defined as

$$regret(t) = \frac{c_{w_*(t)}(t)^T\mu}{b_{w_*(t)}} - \frac{c_{w_i}(t)^T\mu}{b_{w_i}}. \tag{3}$$

The objective is to minimize the total regret R over T.

In our work, we play each arm which is played randomly in proportion to its probability of being optimal. This approach is known as Thompson Sampling and has been shown to be especially successful in system whose limitations allow only periodic updates. It is broadly applicable, easy to implement and can be extended to the work with a broad class of reward distributions.

Before introducing Thompson Sampling, let's build up some intuition. If we have an infinite amount of experiments, then we know exactly what the expected parameter of every worker, and there is no reason to ever explore. When we have a finite number of experiments, then we have to explore and the reason is that we are uncertain of the parameter of every worker's probability distribution. The right machinery to quantify uncertainty is the probability distribution. In Bayesian thinking, we intend to use the entire probability distribution. Let $p_w(r)$ define the probability distribution from which reward is drawn. We don't need to estimate the entire distribution for infinite amount of pulls, because what we care about is its mean μ_w. What we do is to encode our uncertainty of μ_w in the probability distribution $p(\mu_w|data_w)$, and then use that probability distribution to decide when to explore and when to exploit.

The general structure of Thompson Sampling for the contextual bandit problem involves the following elements:

- a set θ of parameters $\tilde{\mu}$;
- a prior distribution $P(\tilde{\mu})$ on these parameters;
- past observations D consisting of context c and reward r for the past time steps
- a likelihood function $P(r|c, \tilde{\mu})$, which gives the probability of reward given a context c and a parameter $\tilde{\mu}$;
- a posterior distribution $P(\tilde{\mu}|D) \propto P(D|\tilde{\mu})P(\tilde{\mu})$, where $P(D|\tilde{\mu})$ is the likelihood function.

More specifically, $p_w(r)$ is the probability distribution that bandit w uses to generate reward when it is pulled. $p(\mu_w|data_w)$ is the probability distribution of the mean of $p_w(r)$ after observing some data.

In our work, we use Gaussian likelihood function and Gaussian prior to design our Thompson Sampling algorithm. More precisely, suppose that the likelihood of reward is $r_i(t)$ at time t, context $c_i(t)$ and parameter μ are given by the pdf of Gaussian distribution $N(c_i(t)^T\mu, v^2)$. Here, $v = R\sqrt{ln(T)}$ parameterizes our algorithm. We have

$$B(t) = I_d + \sum_{\tau=1}^{t-1} c_{w(t)}(\tau)c_{w(t)}(\tau)^T \tag{4}$$

$$\hat{\mu}(t) = B(t)^{-1}(\sum_{\tau=1}^{t-1} c_{w(t)}(\tau)r_{w(t)}(\tau)) \tag{5}$$

Then, if the prior probability for μ at time t is given by $N(\hat{\mu}(t), v^2B(t)^{-1})$, it is easy to compute the posterior distribution at time $t+1$, which is

$$\Pr(\hat{\mu}|r_i(t)) \propto \Pr(r_i(t)|\hat{\mu}) \Pr(\hat{\mu}). \tag{6}$$

In our Thompson Sampling algorithm, at every time step t, we generate a sample $\tilde{\mu}(t)$ from $N(\hat{\mu}(t), v^2B(t)^{-1})$, and play the worker w_i who maximizes $\frac{c_{w_i}(t)^T\mu}{b_{w_i}}$.

Then our algorithm should satisfy:

Definition 1 *Allocation Efficiency (AE)*: An allocation policy is said to be allocation efficiency if it minimizes the regret. Formally,

$$minR(T) = \sum_{t=1}^{T} regret(t) \tag{7}$$

It should be noted that we use a simple filter before we perform worker selection step. The filter means we exclude those workers whose bids are larger than the task's value by assuming the requester is rational.

B. Payment Determination Scheme

In the above discussion, the requester knows that the task should be allocated to which worker. Then we should think about how much we should pay for its effort.

We compute the payment of each worker using the following strategy. For every winning worker for a special task, we run our auction incentive mechanism. First, we temporarily remove the winning worker from the worker set and let the rest workers bid for the task. For each task, we sort all workers' bids, compare all of them to the winning worker's bid until it achieves that the first bid is bigger than the winning one, then its payment is the first bid which is larger than its bid. If there is no one whose bid is larger than the one for the winning one, its payment is its bid. Through this way, we can guarantee that worker's utility is positive.

The bid of every worker at time t can be represented by $b_1(t), b_2(t), ..., b_k(t)$, and their payments can also be represented in $p_1(t), p_2(t), ..., p_k(t)$. If the worker finishes a task successfully, he can get the payment.

A keen concern about the bidding system is that some worker may gain a higher utility by dishonest behavior. This makes the mechanism vulnerable to malicious price

manipulation. Therefore, there are several desirable economic properties that the incentive mechanism should possess. We now define some desirable properties that an allocation policy should satisfy:

Definition 2 *Computational efficiency:* An incentive mechanism is computationally efficient if it can be executed within polynomial time.

Definition 3 *Dominant Strategy Incentive Compatibility (DSIC):* Mechanism M is said to be $DSIC$ if truth telling is a dominant strategy for all the workers. Formally, $\forall i \in K$,b_i and $\hat{b}_i$

$$\sum_{t=1}^{T}\{p_i(t)(w_i, w_{-i}) - b_i(t)\} \geq \sum_{t=1}^{T}\{p_i(t)(w_i, w_{-i} - \hat{b}_i(t))\} \tag{8}$$

Definition 4 *Individual Rationality (IR):* Mechanism M is said to be IR for a worker if participating in mechanism always gives him positive utility. Formally,

$$U_i(t) = p_i(t)(w_i, w_{-i}) - b_i(t) \geq 0, \forall i \in K \tag{9}$$

In general, at every step t of Thompson Sampling, task allocation algorithm consists of generating a d-dimensional sample $\tilde{\mu}$ from a multi-variate Gaussian distribution, and solving the problem $\arg\max \frac{c_i(t)^T \tilde{\mu}(t)}{b_i}$. Therefore, even if the number of arms is large (or infinite), the algorithm is efficient as long as the problem $\arg\max \frac{c_i(t)^T \tilde{\mu}(t)}{b_i}$ is efficiently solvable. After choosing an arm, we decide to give how much payment to that worker. For the payment, we first record the worker's bid who has the maximum $\frac{c_i(t)^T \tilde{\mu}(t)}{b_i}$, and then we temporarily put the worker aside. The requester then sorts the rest workers' bids in an increasing order and choose the first user's bid which is larger than the winning worker's bid as the payment of the winning worker.

C. Utility model

After the requester publishes the task request, it expects to benefit from the task. In our paper, we consider the utility of the requester and worker. Due to the IR property, all workers' utilities are positive. For every task, its value is v_j, after the worker accomplishes a task successfully, it obtains a payment p_i, then the utility of the requester U_r is defined as

$$U_r = \sum_{j=1}^{m}(v_j - p_i) \tag{10}$$

The utility of worker U_w is defined as

$$U_w = \sum_{i=1}^{k}(p_i - cost_i) \tag{11}$$

It is obvious that the worker's cost is usually smaller than its bid. Then the social welfare (sum of the utility) can be defined as

$$SW = U_r + U_w \tag{12}$$

Algorithm 1: Thompson Sampling Task Alloction (TS-TA)

Initilization
Set $B = I_d, \hat{\mu} = 0_d, f = 0_d$;
for $t = 1, 2, ...$ **do**
  if $TA(t) \neq \phi$ **then**
    (WinWorkerSelect) $W_s = \phi$,
    for all task in $TA(t)$ **do**
      for all $w_i \in W$ **do**
        if $b_{w_i} > task.value$ **then**
          $W \leftarrow W \backslash w_i$
        end if
      end for
      repeat
        sample $\tilde{\mu}(t) \leftarrow N(\hat{\mu}(t), v^2 B^{-1})$;
        select worker $w_i(t) := argmax_i \frac{c_i(t)^T \tilde{\mu}(t)}{b_i}$, and observe reward r_t;
        $B = B + c_{a(t)}(t) c_{a(t)}(t)^T$;
        $f = f + c_{a(t)}(t) r_t$, $\hat{\mu}(t) = B^{-1} f$;
      until $r_t = 1$
      $W_s \leftarrow W_s \cup \{(w_i(t), task_j)\}$;
    end for
    (Payment-Decide)
    for all $(w_i, task_j) \in W_s$ **do**
      $Bid' = \phi$, $W' \leftarrow W \backslash w_i$;
      for all w in W' **do**
        $Bid' \leftarrow Bid' \cup b_w^j$;
      end for
      $Bid' \leftarrow$ sort(Bid')
      for b in Bid' **do**
        if $b \geq b_i$ **then**
          $p_i = b$;
          break;
        end if
      end for
      if $p_i = 0$ **then**
        $p_i = b_i$
      end if
    end for
  end if
end for

V. Analysis of TS-TA

In this section, we provide theoretical guarantees for TS-TA. The contextual version of multi-armed bandit problem presents new challenges for the analysis of TS-TA algorithm, and the techniques used so far for analyzing the multi-armed bandit problem [**?**] cannot be directly applicable.

Lemma 1. *TS-TA is computationally efficient.*

Proof. In our incentive mechanism, for every worker, when we need to give a payment, we only need to focus on all other workers' bids. At every round, the task set has m tasks, and the number of worker set is k, in the worst case, it needs compute

$m * k$ times, while it needs m times in the best case, so the average computation times is $\frac{m(k+1)}{2}$. Hence the running time of our incentive mechanism is bounded by $O(m * k)$. □

Lemma 2. *The payment is greater than worker's bid.*

Proof. In our algorithm, we first choose a worker to finish a task, in the payment determination step, we sort the bids for the task in an increasing order, and choose the worker's bid which is the first one which is greater than the winning worker's bid, then the winning worker's payment is decied by that worker's bid. So $p_i \geq b_i$. □

Theorem 1. *TS-TA is IR for all the workers.*

Proof. By Lemma 1. For worker i, we get $p_i \geq b_i$, then the utility of worker i satisfies $utility_i = p_i - b_i \geq 0$. □

Theorem 2. *TS-TA is DSIC.*

Proof. We show that for any worker w_i, it cannot have a higher utility by unilaterally changing its bid from b_i to $\hat{b}_i$. We use case analysis.

Case 1: w_i is not selected for the task by the context c_i and bid b_i, and the winning worker is w_j now. Suppose $\hat{b}_i > b_i$, then the reward $\frac{c_{w_i}(t)^T \mu}{Bid_{w_i}}$ will become more smaller, which makes worker i lose again in the worker selection phase. Otherwise, if $\hat{b}_i < b_i$, the reward will become bigger, in this case it may win the task (depending on context), then we have $p_i \leq b_i$. Thus its utility is $u_i = p_i - b_i \leq 0$.

Case 2: Worker w_i wins in the worker selection phase, earning a payment of p_i to work on a task. Hence by Lemma 1, we get $u_i = p_i - b_i \geq 0$. If it changes its bid, there exists two situations:

Case 2.1: By changing its bid, worker w_i becomes a loser, then the utility of worker w_i becomes 0. It will not get any benefit from this.

Case 2.2: By changing its bid, worker w_i wins in the selection phase, earning a payment $\hat{p}_i$. Since in the payment determination stage, the payment for worker w_i does not depend on the the bid of worker w_i, we have $\hat{p}_i$. Therefore, $\hat{p}_i - b_i \leq p_i - b_i$, the utility for worker w_i cannot increase by changing its bid from b_i to $\hat{b}_i$ unilaterally. This proves Theroem 2. □

Theorem 3. *The regret of our algorithm is bounded by* $O(d \ln T \sqrt{T \ln T \ln(\frac{1}{\delta})})$.

Proof. The worker's context changes at every round and remains the same in each time slot. The regret of the algorithm in time T is the sum of regret of all rounds: $R(T) = \sum_{t=1}^{T} regret(t)$. We follow the proof of [**?**], by defining $v = R\sqrt{\ln T}$, $g(T) = v\sqrt{4d \ln T}$, $\ell(T) = \frac{v}{2\sqrt{2}} + 1$, we get $R(T) \leq \frac{6d \ln T \sqrt{T}}{p} + \frac{8dR \ln T \sqrt{2T \ln T \ln(\frac{1}{\delta})}}{p} + \frac{2R\sqrt{d} \ln T}{pT}$. With probability $1 - \frac{1}{\delta}$,

$$
\begin{aligned}
R(T) &= \sum_{t=1}^{T} regret(t) \\
&\leq \sum_{t=1}^{T} (\frac{g(T)}{p} I(a(t) \notin C(t)) s_{t,a^*(t)}) + \frac{2g(T)}{pT} \\
&\quad + \frac{2g(T)^2}{\ell(T)} \sum_{t=1}^{T} s_{t,a(t)} + \frac{8}{p} \frac{g(T)^2}{\ell(T)} \sqrt{2T \ln(\frac{2}{\delta})} \\
&\leq \sum_{t=1}^{T} (\frac{g(T)}{\ell(T)} \frac{1}{p} I(a(t) \notin C(t)) s_{t,a(t)}) + \frac{2g(T)}{pT} \quad (13) \\
&\quad + \frac{2g(T)^2}{\ell(T)} \sum_{t=1}^{T} s_{t,a(t)} + \frac{8}{p} \frac{g(T)^2}{\ell(T)} \sqrt{2T \ln(\frac{2}{\delta})} \\
&\leq \frac{g(T)^2}{\ell(T)} \frac{3}{p} \sum_{t=1}^{T} s_{t,a(t)} + \frac{2g(T)}{pT} + \frac{8}{p} \frac{g(T)^2}{\ell(T)} \sqrt{2T \ln(\frac{2}{\delta})}
\end{aligned}
$$

Now we define a function of x as $f(x) = \frac{3d \ln x \sqrt{x}}{p} + \frac{8dR \ln x \sqrt{x \ln x \ln(\frac{1}{\delta})}}{p} + \frac{2R\sqrt{d} \ln x}{px}, x \in [0, \infty)$, then we can get $R(T) \leq \sum_{t=1}^{T} f(t)$. According to the principle that if function $g(x)$ and function $h(x)$ are both convex functions, then $g(x) + h(x)$ is also a convex function. So we can easily prove $f(x)$ is a convex function for any $x \geq 3$ and $0 < \delta < 1$, and then we can use Jensen's inequality and obtain that: with probability $1 - \delta$, the total regret bound in time horizon T is bounded by:

$$
O(d \ln T \sqrt{T \ln T \ln(\frac{1}{\delta})}), 0 < \delta < 1. \quad (14)
$$

So our algorithm's regret is bounded. □

VI. Numerical Experiments

This section gives a detailed description of our experimental setup, including data collection, feature construction and performance evaluation.

A. Data Collection

Our datasets comes from MIT Reality Mining. It collects data from 100 mobile phone users over the course of 9 months (September 2004 to June 2005) at MIT. The datasets includes the participants' messages, phone call histories, Bluetooth discovery records, answers of a survey, the cell tower IDs, etc. We extract communication information and location based on phone call histories. We use the first six month data as training set and the other part is used for testing. In the train phase, we record every user who gets in touch with the requester, including call and message history. By this, we can confirm everyone's contacts, and then in our algorithm we take the k members of his contacts as k arm bandits. Another important pre-processing is that we execute a simple user analysis, we can get each user's home and work place. Because the datasets contains cell tower IDs, we can get each worker's home or work place by the history special time the worker usually showed up.

B. Feature Construction

We now describe the task/worker features (also called context) constructed for our experiments. The features are used for our reward model. Context means what situation the requester meets in order to allocate the tasks, for example, how much is the worker's bid for the task, how many buffer the worker holds, how far is the worker away from the task location, and so on. For different contexts, the requester's choice is different, thus we start with raw contexts that are collected after the requester publishes a task request. Then, each worker is originally represented by a raw feature vector, which includes:

- **worker's bid for current task**: it's crucial for the requester because it wants to maximize its utility which is task's value minus the payment, and worker's payment is relevant to its bid in our incentive mechanism ($p_i \geq b_i$). In other words, the worker's bid will affect its payment, so we should take this factor into consideration.
- **worker's buffer size**: every worker has limited space or ability to finish limited tasks. We assume every worker's buffer size is 100 initially.
- **the probability of accomplish a task**: this relates to worker's location and its ability or willingness to finish the task.
- **centrality**: this will be viewed as a factor which may affect the requester's decision. Intuitively, the bigger the centrality of a worker is, the higher possibility the worker to accomplish a task.
- **task value**: we construct a vector (task $\times$ worker), by considering the value of a task. We may consider more dimensions about the task, for example the category of the task, the duration of the task, and so on.

Note that for the task value, we carry out normalization to make the feature's value belongs to [0,1]. Of course, we may consider more information, however we should first check if the new feature is efficient to the algorithm. In this work, we carry out some experiments shown in Fig.2 and Fig.3 to verify the efficiency of the features we provide.

C. Compared Algorithms

In this paper, we use several exploration/exploitation strategies. They only differ in ways the worker is selected and the rest of the model remains the same, these strategies are as follows:

Thompson Sampling task allocation: the current worker context vector is drawn from a posterior distribution $\mu \sim N(\hat{\mu}(t), v^2 B(t)^{-1})$, and it selects the best worker based on the sampling results, $w_i(t) := \arg\max_i \frac{c_i(t)^T \tilde{\mu}(t)}{b_i}$.

ϵ-greedy: With $(1-\epsilon)$, it selects the best worker who has the highest reward, which is $w_i(t) := argmax_i \frac{c_i(t)^T \tilde{\mu}(t)}{b_i}$, otherwise it selects another worker with ϵ.

Epoch-greedy: The algorithm is similar as ϵ-greedy, while in this algorithm, ϵ is small enough, so that it can explore for a while, then after that, it sets ϵ to $1/\sqrt{t}$, where t represents the times of the algorithm has been executed. After the algorithm has been executed enough times, the reward distribution of every worker can be estimated, so there is no need to explore any more.

Upper Confidence Bound(UCB): This algorithm selects the best worker based on Upper Confidence Bound of the reward, which means it considers the mean and the variance of reward. The best worker will satisfy $w_i = \arg\max(\frac{c_i(t)^T \tilde{\mu}(t)}{b_i} + \alpha\sqrt{c(t)_w{}^T B^{-1} c_w(t)})$.

Random selection: It selects the best worker randomly.

These algorithms are used in the multi-armed bandit problem to deal with the exploration/exploitation trade-off. We compare these algorithms with our proposed algorithm.

D. Simulation study

We first carry out a simulation to check the influence of different contexts. In the MIT datasets, there are 100 nodes, and through analyzing the data trace, we can get their contacts, the number of which is the number of arms. At some time stamps, we generate several tasks ($\#task \in [1, 20]$), and the requester allocates these tasks in order. The worker's bid is according to the task value, it can be greater or smaller than the task value. The buffer size decreases 1 if the worker accepts a task, when it finishes the task, the buffer size increases by 1. The probability of a worker to accomplish a task is related to the worker's location. We first analyze the data trace, it's easy to get every worker's work place and home location. Through behavior analysis, we can predict when the worker will appear at a specific place. So we can calculate the probability of a worker to finish the task. The centrality is calculated by the worker's contacts. The task's value is time-dependent, which means the more time the task needs, the more value it is.

First, we check the influence of different contexts to our algorithm. In the following table, we show the different contexts used in the algorithm.

TABLE III
TS-TA OF DIFFERENT CONTEXT

algorithm name	context used
$TS-TA-1$	bid
$TS-TA-2$	bid+probability
$TS-TA-3$	bid+probability+buffersize
$TS-TA-4$	probability+buffersize+centrality+value
$TS-TA-5$	bid+probability+buffersize+centrality+value
$TS-TA-6$	bid+probability+buffersize+centrality

In the part of analysis of TS-TA, we show that the regret is bounded by $O(d \ln T \sqrt{T \ln T \ln(\frac{1}{\delta})})$, it means the regret is related to the dimension of context. When providing more information of the worker, we can describe the worker's quality more accurately, but it needs more time to explore the distribution of a worker. In Fig. 2, firstly we can see that the curve is a logarithmic form, which is identical to our analysis. And the curve represents that as time goes by, we can know more about the worker's service quality, then we don't need to explore any more, we can choose the best worker to finish

a task. Secondly, it's easy to find that the regret of TS-TA-5 is the highest, this exactly proves that the dimension of context influences the regret. We see that TS-TA-6 overlaps with TS-TA-5, and its dimension is smaller than TS-TA-5. In this situation, it means that different contexts may have different influences to the algorithm.

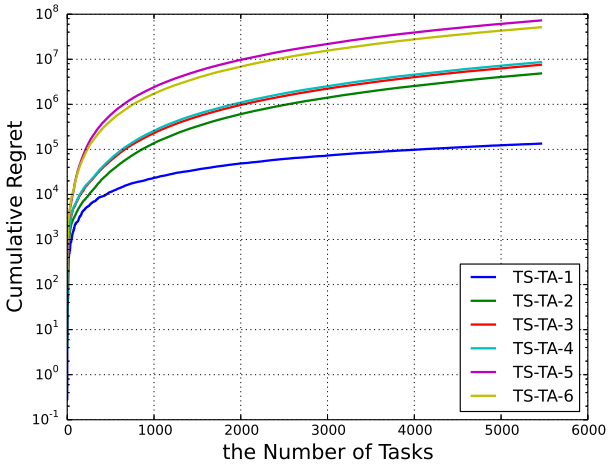

Fig. 2. Cumulative Regrets of Different Contexts in TS-TA

Furthermore, we carry out simulation to demonstrate the performance of the requester's utility on different contexts. In Fig.3, we can find that TS-TA-4 and TS-TA-5 have the highest utility. In order to find the best worker as soon as possible and at the same time maximize the requester's utility with acceptable service quality, we choose TS-TA-4, since the contexts have serval dimensions which can represent the service quality of worker. In Fig.2, we can see its regret is at the middle, and at same time the requester's utility is quite high. So we choose the contexts used in TS-TA-4 as our algorithm context vector.

Fig.3 shows the results in terms of the requester's utility under different MAB algorithms. Our goal is to achieve a high probability to choose a worker with good service quality and maximize the utility of the requester. In Fig.4, we can find that our TS-TA achieves the highest utility of requester. And with the increase of time, the difference of utility between TS-TA and other algorithms becomes larger, which demonstrates that our algorithm can achieve long term benefit. For the random selection algorithm, it's obvious that the utility cannot guarantee to be always acceptable. UCB and epsilon-greedy cannot maximize the utility. As seen in the curve, it's obvious that if more tasks are finished by a worker, the cumulative utility of requester becomes larger.

Fig.5 depicts the sum of different workers' utilities in our TS-TA algorithm. It is clear that some workers' utilities are very high, while others are almost approaching to zero, which illuminates that some workers are selected to accomplish the task more frequently, while others are not.

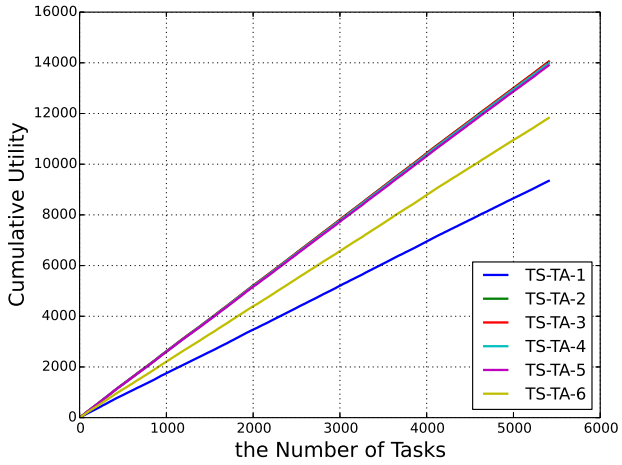

Fig. 3. Requester Cumulative Utility of Different Contexts

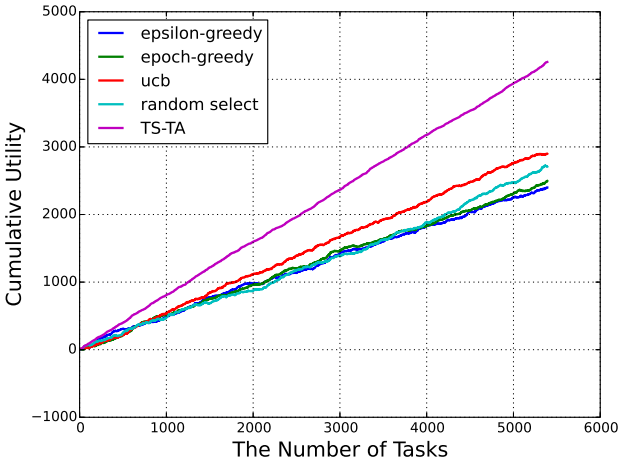

Fig. 4. Requester Cumulative Utility of Different Algorithms

Fig.6 shows the cumulative social welfare of our algorithm. Usually, social welfare can be regarded as the income of all participants in the mechanism. In Fig.6, the trend of social welfare is the same as the utility of requester, while its value is greater than the requester's utility, which means workers also gain positive utilities.

In summary, the simulation results show that (1) Thompson Sampling algorithm for task allocation can learn the workers' service qualities using the workers' contexts and select a proper worker to finish a task; (2) when context changes, the algorithm can dynamically change to pick current suitable worker; (3) Thompson Sampling algorithm for task allocation can achieve highest requester's utility compared to epsilon-greedy, UCB and random selection, to deal with the exploration/exploitation trade-off in our context-bandit problem.

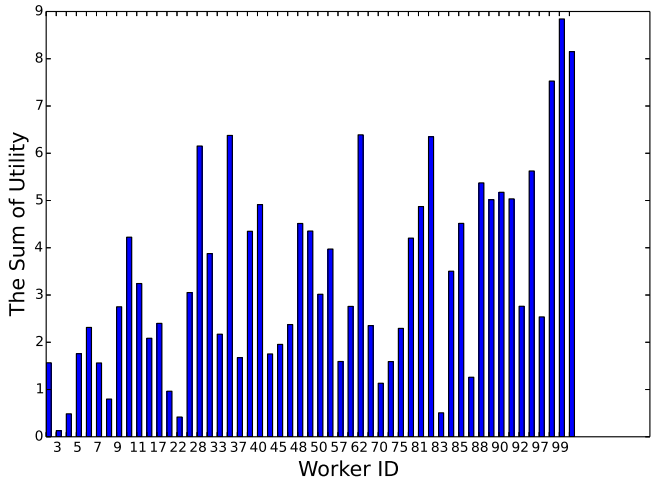

Fig. 5. the Sum of Worker's Utility

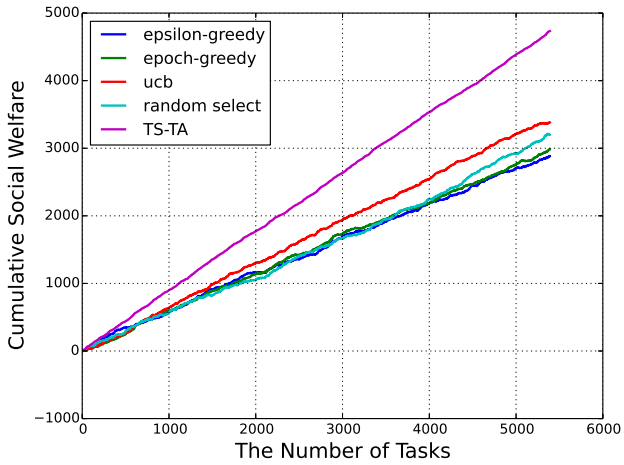

Fig. 6. Cumulative Social Welfare

VII. Conclusion

In this paper, we studied task allocation problem for mobile crowdsensing with unknown uses' service qualities. In order to select the best worker from the requester's contacts, we constructed a context vector which relates to worker and task, and then proposed Thompson Sampling allocation algorithm with incentive mechanism. We showed that our algorithm can explore the workers' service qualities and select the best worker to accomplish a task. This algorithm keeps the regret low, and increases the requester's utility. The experiment results have shown that our algorithm outperforms epsilon-greedy and UCB which are widely used in addressing MAB problems. We have proved our algorithm guarantees DISC, IR and computation efficiency.

References

[1] V. Coric and M. Gruteser, "Crowdsensing maps of on-street parking spaces," in *2013 IEEE International Conference on Distributed Computing in Sensor Systems*. IEEE, 2013, pp. 115–122.

[2] A. Farshad, M. K. Marina, and F. Garcia, "Urban wifi characterization via mobile crowdsensing," in *Network Operations and Management Symposium (NOMS), 2014 IEEE*. IEEE, 2014, pp. 1–9.

[3] Y. Liu and M. Liu, "An online learning approach to improving the quality of crowd-sourcing," in *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*. ACM, 2015, pp. 217–230.

[4] M. Srivastava, T. Abdelzaher, and B. Szymanski, "Human-centric sensing," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 370, no. 1958, pp. 176–197, 2012.

[5] M. Xiao, J. Wu, L. Huang, Y. Wang, and C. Liu, "Multi-task assignment for crowdsensing in mobile social networks," in *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 2015, pp. 2227–2235.

[6] S. Jain, B. Narayanaswamy, and Y. Narahari, "A multiarmed bandit incentive mechanism for crowdsourcing demand response in smart grids," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[7] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 173–184.

[8] Y. Zhang and M. van der Schaar, "Reputation-based incentive protocols in crowdsourcing applications," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 2140–2148.

[9] Z. Feng, Y. Zhu, and L. M. Ni, "imac: Strategy-proof incentive mechanism for mobile crowdsourcing," in *International Conference on Wireless Algorithms, Systems, and Applications*. Springer, 2013, pp. 337–350.

[10] B. Hoh, T. Yan, D. Ganesan, K. Tracton, T. Iwuchukwu, and J.-S. Lee, "Trucentive: A game-theoretic incentive platform for trustworthy mobile crowdsourcing parking services," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2012, pp. 160–166.

[11] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges." *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[12] G. Cardone, L. Foschini, P. Bellavista, A. Corradi, C. Borcea, M. Talasila, and R. Curtmola, "Fostering participaction in smart cities: a geo-social crowdsensing platform," *Communications Magazine, IEEE*, vol. 51, no. 6, pp. 112–119, 2013.

[13] Z. Feng, Y. Zhu, Q. Zhang, L. M. Ni, and A. V. Vasilakos, "Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing," in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 1231–1239.

[14] S. He, D.-H. Shin, J. Zhang, and J. Chen, "Toward optimal allocation of location dependent tasks in crowdsensing," in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 745–753.

[15] D. A. Berry and B. Fristedt, *Bandit problems: sequential allocation of experiments (Monographs on statistics and applied probability)*. Springer, 1985.

[16] H. Robbins, "Some aspects of the sequential design of experiments," in *Herbert Robbins Selected Papers*. Springer, 1985, pp. 169–177.

[17] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3/4, pp. 285–294, 1933.

[18] R. Agrawal, "Sample mean based index policies with o (log n) regret for the multi-armed bandit problem," *Advances in Applied Probability*, pp. 1054–1078, 1995.

[19] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[20] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in applied mathematics*, vol. 6, no. 1, pp. 4–22, 1985.

[21] S. Agrawal and N. Goyal, "Analysis of thompson sampling for the multi-armed bandit problem." in *COLT*, 2012, pp. 39–1.