A Multiarmed Bandit Incentive Mechanism for Task Allocation in Crowdsensing

Michael Shell, *Member, IEEE,* John Doe, *Fellow, OSA,* and Jane Doe, *Life Fellow, IEEE*

Abstract—The multi-armed bandit problem has attracted remarkable attention in the machine learning community and many efficient algorithms have been proposed to handle the so-called exploitation-exploration dilemma in various bandit setups. At the same time, significantly less effort has been devoted to adapting bandit algorithms to particular architectures, such as sensor networks, multi-core machines, or peer-to-peer (P2P) environments, which could potentially speed up their convergence.

Index Terms—Task Allocation, multiarmed bandits, cellular network, reward maximization, Thompson sampling

1 Introduction

Mobile crowdsensing is a new type of collaboration which involves a crowd of mobile users who uses their carried smart phone to execute some complex computation or sensing services in some environment. Since it can utilize the mobility of users to solve large-scale mobile sensing tasks, it has stimulated a number of attractive applications, such as urban WiFi characterization, traffic information mapping, and so on. Moreover, there have been some platforms, frameworks, and incentive mechanism, designed for crowdsensing in all kinds of scenario. [21] [16]

Consider that a user carried mobile phone in an mobile social network has some independent sensing tasks, such as sample WiFi fingerprint, traffic information mapping, and so on. However, these tasks cost too much time and exceed its processing ability. Then, it requests other mobile users for help by starting a crowdsensing. In this scenario, the requester will send tasks to and receive results from other mobile users, of which can by calling other or by send message to communicate. And it often involves large size data transmissions, however, we can incent many users joining it, such that can decrease every user's work time. [21] adapts short-distance wireless communication technologies (e.g., WiFi or Bluetooth), in contrast, we consider a different communication technology, which through cellular network get in touch with each other. In this situation, the requester doesn't have to stay not far from the worker, and what's more, the requester can do another work after publishes tasks. 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 send message to the work ask they bid for the task. After the work return their bid and workload, then the requester make a decision to choose a best worker to handle these tasks. The best worker means he will finish the worker very well and at the same time the bid is reasonable. Then, an important problem is how the requester allocates these task to workers, so as to for every task can choose a best worker to finish it.

• *M. Shell is with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332. E-mail: see http://www.michaelshell.org/contact.html*
• *J. Doe and J. Doe are with Anonymous University.*

Manuscript received April 19, 2005; revised September 17, 2014.

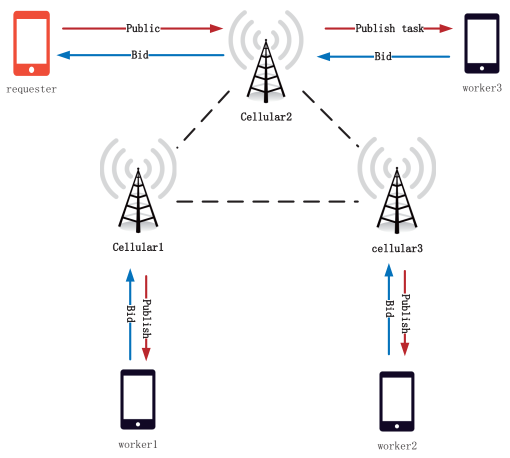

Fig. 1: mobile social network

In this paper, we focus on the above task allocation problem. In this problem, choosing a best worker from the requester's contacts. Selects best worker just like choose excellent contractor, which will let the utility maximize of the requester.

To solve the above task allocation problem in mobile crowdsensing, we proposed a based multi-armed bandits algorithm with incentive mechanism, which is called Thompson sampling task allocation(TS-TA). We choose Thompson sampling to handle this problem.

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, e.g., in [?][?]. 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, play an arm according to its posterior probability of being the best arm. The general structure of TS for the contextual bandits 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 b, reward r) for the past time steps
- a likelihood function $P(r|b, \tilde{\mu})$, which gives the probability of reward given a context b 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.

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. More specifically, the following are our major contributions in this paper:

1) We formulate the problem into a linear payoffs MAB problem that shall find optimal worker when there are tasks need to be allocated. Then we proposed a novel algorithm, Thompson Sampling Task Allocation(TS-TA) which make sure the worker is truthful and rational.
2) We proved that the regret has a lower bound, where any MAB algorithm has to suffer.
3) We conduct our simulations on real traces dataset to evaluate the TS-TA algorithm. The results show that the proposed algorithm can achieve O(logT) lower bound.

The remainder of the paper is organized as follows.We present a summary of the relevant work in Section 2. In Section 3, we introduce how multi-armed bandits can be model for task allocation problem. In Section 4, we formulate the task allocation problem. And next in Section 5, we proposed our algorithm thompson sampling for task allocation, which according to the thompson sampling to choose a worker to finish current work and then decide to give how much payment to that worker. In Section 6, we do some theory work to prove our algorithm's regret is bounded. In Section 8, we provide a simulation on a real dataset to verify our algorithm. Conclusions is provided in Section 9.

2 RELATED WORK

2.1 Task allocation

In this paper, we focus on the task assignment problem in mobile crowdsensing, which involves a lot of mobile users in solving a large job through their carried mobile devices [14]. By far, there have been several frameworks, platforms, and incentive mechanisms designed for crowdsensing system, [9], [12], [13], [15], [14]. Among them, the most related work is the LRBA algorithm designed for a task allocation problem in mobile crowdsensing [15]. Unlike our work, the problem in this work pay attention to how to choose the worker who will provide better service and the tasks are location dependent.

On the other hand, our task assignment problem is also different from traditional parallel machine scheduling problems. In fact, there have been thousands of papers on parallel machine scheduling problems by far. Literatures [5], [11] have made a detail review on these works. Even in recent years, there is still much research on the complex parallel machine scheduling problems, such as [4], [10]. The most related works among the existing researches is [18], who consider a crowdsourcing problem where in the process of labeling massive datasets, multiple labelers with unknown annotation quality must be selected to perform the labeling task for each incoming data task. And they formulate it to MAB problems, then proposed an algorithm to solve it. In contrast, the tasks allocates has to consider the current context and then select a worker, what's more, the payment of workers' efforts also be taken into consideration. More specifically, each task being sent to a mobile worker is according to thompson sampling algorithm and the payment is decided just like an auction mechanism just like what have existed in crowdsourcing. Such a unique task assignment model makes our problem different from existing parallel machine scheduling problems.

2.2 MAB Algorithms

The fundamental challenge in bandit problems is the the need for balancing exploration and exploitation. Changing the aim of maximize reward to minimize the regret multiple experiments, an algorithm A will exploits 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 A's knowledge. In other to avoid this undesired situation, A has to explore by actually choosing seemingly suboptimal arms to gather more information about them. Exploration can increasing short-term regret since some suboptimal arms may be chosen. However, obtaining information about the arms' average payoffs can refine A'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. Our problem belongs to the stochastic MAB setting, where the distribution function of reward of each arm is known but the parameter is unknown. A recent survey by Bubeck and Cesa-Bianchi [8] compiles several variations on stochastic and non-stochastic MAB problems. Our setting is very close to [3] where a general bandit problem with linear payoff function which is related to the context. Our setting is a further concretization, as the context in our algorithm is concrete and we do some experiments to decide the weights of different feature which are used for the context.

The context-free K-armed bandit problem has been studied by statisticians for a long time like [7], [19] and [20]. One of the simplest and most straightforward algorithms is $\epsilon - greeedy$. In each round t, this algorithm first estimate the average payoff $\hat{\mu}_{t,a}$ of each arm a according to the history data. Then, it selects the arm who has the highest payoff estimate with probability $1 - \epsilon$; with probability ϵ , it chooses a random arm. In the limit, each arm will be tried infinitely usually, and so the payoff estimate $\hat{\mu}_{t,a}$ converges to the true value μ_a with probability 1 (Central Limit Theorem). Furthermore, by decaying ϵ appropriately like [19], the per-step regret, $R_A(T)/T$, converges to 0 with probability 1.

In contrast to the unguided exploration strategy adopted by $\epsilon - greedy$, another class of algorithms generally known as upper confidence bound algorithm [1], [6], [17] use a smarter way to balance exploration and exploitation. Specifically, in trial t, these algorithms estimate both mean payoff $\hat{\mu}_{t,a}$ of each arm a as well as a corresponding confidence interval $c_{t,a}$, so that $|\hat{\mu}_{t,a} - \mu_a| < c_{t,a}$ holds with high probability. They then select the arm that achieves a highest upper confidence bound (UCB): $a_t = \arg\max_a(\hat{\mu}_{t,a} + c_{t,a})$. With appropriately defined confidence intervals, it can be shown that such algorithms have a small total T-trial regret that is

Multi-armed Bandit(MAB)	Task Allocation(TA)
Player	Requester
Armed bandit	Workers
Pull level	Select worker

TABLE 1: Parallel between Multi-armed bandit and Task Allocation

Notation	Description
K	Set of all workers in the requester's contacts
m	Number of tasks
$b_i(t)$	Context of arm i at time t
μ	The prior parameter of the arm
$r_i(t)$	The reward of arm i at time t
π	The allocation mechanism
RE^{π}	The total regret
${p_i}^t$	Payment of arm i

TABLE 2: Notation used

only logarithmic in the total number of trials T, which turn out to be optimal [17].

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

3 A Parallel with Crowdsourcing

"Crowdsourcing is the act of taking job traditionally performed by a designated agent(usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call", this is what Jeff Howe ever offered the definition of crowdsourcing. The designated agent is often called the requester, while people working on the task are called crowd workers. The requester needs to assign the task to these workers with unknown capability and willingness to complete the task based on the goals to achieve. In mobile crowdsensing network, there is a task publisher at some time, and he/she can chooses a worker from his/her contacts. The task publisher just like the requester, while his/her contacts just like the group of works. Table 1 precisely demonstrates this paralle between crowdsoucing and mobile crowdsensing.

4 A Parallel with MAB

5 Model&Problem

5.1 Model for Task Allocation

Consider a network denoted by a graph $G^t = (V^t, E^t)$ at time t with a set $V^t = \{v_i | i = 1, ..., K\}$ of K nodes represented the node set who meet the task requester node represented by v_0 at time t, a set E of edges denoting v_0 can communicate with other nodes. And the graph is star network structure. The system is been divided into time slot, and at each round t, node v_0 has K choices of relay node, where relay nodes are i.i.d. Suppose the ability of every node accomplish a task as stochastic process $\chi_i(t)$ over time with a mean $\mu_i \in [0, 1]$, as done in many previous works [1] [2] [3].

At each round t, an K-dimensional *strategy* vector $\Phi_0(t) = \{\Phi_{0,i} | i = 1, ..., K\}$ is selected under some *policy* from the *feasible strategy set* F. Here $\Phi_{0,i}$ is the index of relay node selected by node carried message in strategy Φ_0. We use i=1,...,K to index strategies of feasible set F in XXXXX. By feasible we mean at that time slot, carried message node meet another k nodes(who can communicate with the message carrier directly). When a strategy Φ_0 is determined, then one node v_i will be selected, and it will give back a reward $r_i(t)$. And then the total reward of T round is defined as $R = \sum_{t=1}^{T} r(t)$. The goal of this allocation problem is to maximize the accumulated reward R over the T rounds. Obviously, if the reward of every node is constant, then we can simply choose the node who will have maximize reward every time slot. While in our problem, the reward is variable over time. And it's probability distribution follows $f(x, \theta)$ where θ is an unknown parameter. The gap between the actual total reward we get and the optimal total reward theoretically is call regret of the algorithm, and it can be formally written as $RE = \sum_{t=1}^{T} r^*(t) - \sum_{t=1}^{T} r(t)$, where r^* is the max reward we can get every time slot. The goal of the algorithm is to keep the regret RE as low as possible.

An allocation strategy(also called a policy) π specifies the action chosen by the task requester(carried message node) at each time in the system. Formally, π is a sequence of random variables $\pi = \{\pi^1, \pi^2, ..., \pi^T\}$ where $\pi^t = \{a_1^t, a_2^t, ..., a_k^t\}$ is the vector of zeros and ones(deterministic mechanism). If $a_i^t = 1$ then node i will be chosen as a relay node at slot t, whereas, $a_i^t = 0$ means node i doesn't be chosen at all, i.e., $\forall t, \sum_i a^t \leq 1$.

There are K arms at time t for a requester, a context vector $b_i(t) \in R^d$, is revealed for every arm i. These context vectors are chosen by a requester in an adaptive manner after observing the arms played and their rewards up to time t-1, i.e. history H_{t-1},

$H_{t-1} = \{a(\tau), r_{a(\tau)}(\tau), b_i(\tau), i = 1, ..., K, \tau = 1, ..., t - 1\}$,

where $a(\tau)$ denotes the worker played at time τ, $r_{a(\tau)}(\tau)$ denotes when chose arm a the reward you have gotten. Given $b_i(t)$, the reward for arm i at time t is generated from an (unknown) distribution with mean $b_i(t)\mu$, where $\mu \in R^d$ is a fixed but unknown parameter [3].

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

At every time slot t, for the contextual bandit problem algorithm needs to select an arm $a(t)$ to play, using history $H_t - 1$ and current contexts $b_i(t), i = 1, ..., K$. Let $a^*(t)$ denote the optimal arm at time t, i.e. $a^*(t) = \arg\max_i b_i(t)^T \mu$. And then the regret of strategy π at time t is defined as follows:

$$RE^{\pi} = \sum_{t=1}^{T} \{r^*(t) - \sum_{i=1}^{K} a_i r_i(t)\} \tag{1}$$

The objective is to minimize the total regret RE in overall T. The time horizon T is finite but possibly unknown.

Suppose that $\eta_{i,t} = r_i(t) - b_i(t)^T\mu$ is conditionally R-sub-Gaussian for a constant $c \geq 0$ just like in agrawal2012thompson, i.e.,

$\forall\lambda \in R, E[e^{\lambda\eta_{i,t}}|\{b_i(t)\}_{i=1}^N, H_{t-1}] \leq \exp(\frac{\lambda^2c^2}{2})$

This assumption is satisfied whenever $r_i(t) \in [b_i(t)^T\mu - c, b_i(t)^T\mu + c]$ (see Remark 1 in Appendix Filippi et al.(2010)). We will also assume that $\|b_i(t)\| \leq 1, \|\mu\| \leq 1$, and $regret_i(t) \leq 1$ for all i,t (the norms, unless otherwise indicated, are L_2-norms). These assumptions are required to make the regret bounds scale-free, and are standard in the literature on this problem. If $\|b_i(t)\| \leq \alpha, \|\mu\| \leq \alpha, regret_i(t) \leq \alpha$ instead, then our regret bounds would increase by a factor of α.

we now define some of the desirable properties that an allocation policy show satisfy:

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

$$\pi = \arg\min_{\pi} RE^{\pi}$$

Definition 2 *Dominant Strategy Incentive Compatibility(DSIC):* Mechanism M is said to be DSIC if truth telling is a dominant strategy for all the workers. Formally,

Definition 3 *Individual Rationality(IR):* Mechanism M is said to be IR for a worker if participating in mechanism always gives him positive utility. Formally,

5.2 Problem

Consider a mobile crowdsensing in the mobile social network. Without loss of generality, we let the requester be user v_0, and suppose that the requester has m homogeneous mobile sensing tasks in one cycle, denoted by $j(n)$. The length of one cycle is T, and these tasks might be different types of task in different cycle. For simplicity, we assume that every requester will choose one worker from who ever get contacted with the requester for every task allocation and the requester's contact is built at the beginning within time t. It means if someone get contact with the requester during the period of the beginning t, the one will be add to the requester's contacts, regardless of who is the caller.

In this paper, we focus on how to choose a sequence of workers so as to maximize the utility of requester in every cycle.

6 Task allocation algorithm

6.1 Thompson Sampling for Task Allocation

In this section, we will explain our Thompson Sampling algorithm for task allocation with incentive mechanism. The most important challenge is to balance exploitation with exploration. That is, we have two somewhat conflicting goals: (a) quickly find the best arm to pull and (b) pull the best arm as often as possible. Now we'll introduce the asymptotic optimal algorithm: Thompson Sampling.

Before I introduce the Thompson sampling, let's build up some intuition. If we have an infinite amount of pulls, then we know exactly what the expected rewards are, and there is no reason to ever explore. When we have a finite number of pulls, then we have to explore and the reason is that we are uncertain of what is the best arm. The right machinery to quantify uncertainty is the probability distribution. In Bayesian thinking, we want to use the entire probability distribution. Let's define $p_a(r)$ is the probability distribution from which rewards are drawn. That's what controls the bandit. We don't need to estimate the entire distribution for infinite amount of pulls, because all we care about is its mean μ_a. What we will do is encode our uncertainty of μ_a in the probability distribution $p(\mu_a|data_a)$, and then use that probability distribution to decide when to explore and when to exploit.

let's say more clearly about the two probability:
$p_a(r)$ - the probability distribution that bandit a uses to generate rewards when it is pulled.
$p(\mu_a|data_a)$ - the probability distribution of where we think the mean of $p_a(r)$ after observing some data.

With Thompson sampling you keep around a probability distribution $p(\mu_a|data_a)$ that encodes your belief about where the expected reward μ_a is for arm a. For the simple coin-flip case, we can use the convenient Beta-Bernoulli model, and the distribution at round t after seeing S_a, t successes and F_a, t failures for arm a is simply:
$p(\mu_a|data_a) = Beta(S_a, t+1, F_a, t+1)$,
where the added 1's are convenient priors.

In our problem, we use Gaussian likelihood function and Gaussian prior to design our version of Thompson Sampling algorithm. More precisely, suppose that the likelihood of reward $r_i(t)$ at time t, given context $b_i(t)$ and parameter μ, were given by the pdf of Gaussian distribution $N(b_i(t)^T\mu, v^2)$. Here, $v = R\sqrt{\frac{24}{\varepsilon}d\ln(\frac{1}{\delta})}$, with $\varepsilon \in (0,1)$ which parameterizes our algorithm,just like [3]. Let

$$B(t) = I_d + \textstyle\sum_{\tau=1}^{t-1} b_{a(t)}(\tau)b_{a(t)}(\tau)^T$$

$$\hat{\mu}(t) = B(t)^{-1}(\textstyle\sum_{\tau=1}^{t-1} b_{a(t)}(\tau)r_{a(t)}(\tau))$$

Then, if the prior 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,

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

as $N(\hat{\mu}(t), v^2B(t)^{-1})$ (detail of this computation are in Appendix A.1). In our Thompson Sampling algorithm, at every time step t, we will simply generate a sample $\hat{\mu}(t)$ from $N(\hat{\mu}(t), v^2B(t)^{-1})$, and play the arm i that maximizes $b_i(t)^T\hat{\mu}(t)$.

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 $argmax_i b_i(t)^T\hat{\mu}(t)$. Therefore, even if the number of arms N is large (or infinite), the above algorithm is efficient as long as the problem $argmax_i b_i(t)^T\hat{\mu}(t)$ is efficiently solvable. After choose an arm, then, we should decide to give how much payment to that work. For this part, we have to make sure the worker is rational and truthful. For the payment, we first record the worker's bid who have the max $b_i(t)^T\hat{\mu}(t)$, and then we temporarily put the worker aside who has been selected just now, and selected the other worker who has the max $b_i(t)^T\hat{\mu}(t)$ over the rest of workers. The requester then collects the rest worker's bid and choose a property one's bid, which will be as the payment of the first worker. Just as in our

Algorithm 1 TS-TA

```
    Set $B = I_d, \hat{\mu} = 0_d, f = 0_d$
    for $t = 1, 2, ...,$ do
3:      Sample $\hat{\mu}(t)$ from distribution $N(\hat{\mu}(t), v^2 B(t)^{-1})$.
        Play arm $a(t) := argmax_i b_i(t)^T \hat{\mu}(t)$, and observe reward $r_t$.
        Update $B = B + b_{a(t)}(t) b_{a(t)}(t)^T, f = f + b_{a(t)}(t) r_t, \hat{\mu}(t) = B^{-1} f$.
6:      if $r_t = 1$ then
            current arm is i(t), then we need to give him payback.
            choose arm $j(t) := \arg\max_{j \in K \setminus i} \theta_j(t)$ and observe reward ${r_t}'$
9:          Perform a Bernoulli trial with success probability $\tilde{r}_t'$ and observe output ${r_t}'$
            if ${r_t}' = 1$ then
                ${p_i}^t = min({p_j}^t, p_m ax)$
12:         else
                ${p_i}^t = min({p_i}^t, p_m ax)$
        else
15:         do not allocation this task at this round.
```

algorithms.

7 Analysis of AT-algorithm

In this section, we provide theoretical guarantees for TS-TA. The contextual version of the multi-armed bandit problem presents new challenges for the analysis of TS algorithm, and the techniques used so far for analyzing the basic multi-armed bandit problem by [2] do not seem directly applicable.

Theorem 1. *TS-TA is IR for all worker.*

Proof of TS-TA is IR. Here is the proof of IR. □

Theorem 2. *TS-TA is DSIC.*

Proof. here is proof of DSIC. □

Lemma 1. *our regret is bounded.*

8 Simulation Experiments

This subsection gives a detail description of our experimental setup, including data collection, feature construction, performance evaluation.

8.1 Data Collection

Our dataset comes from MIT Reality Mining. It collected from 100 mobile phone users over the course of 9 month (September 2004 to June 2005) at MIT. The dataset includes the participants' message and phone call history, Bluetooth discovery records, answers of a survey etc. We again extract communication information and location based on phone call history. We use the before six month data as training set and the later part is used for testing algorithm. In train, we statistic every user who ever have gotten in touch with him, including call and message history. Through this, we can confirm everyone's contacts, and then in our algorithm we will regard the k members of his contacts as k arm bandits.

8.2 Feature Construction

We now describe the feature(also called context) constructed for our experiments. The features are used for our reward model, for we have define the reward as a linear payoffs. Context means what situation the requester meets in order to allocate the works, for example, how much is the worker's bid for the task, how many size there spares in worker's buffer and so on. Because when meets different context, the requester's choice is different. We start with raw context that were selected by "support". The support of a feature is the fraction of context. To reduce noise in the data, we only selected features with high support and do some preprocess. Then, each worker was originally represented by a raw feature vector, which include:

- **worker's bid for concurrent task**: the requester also should think of its utility (task's value minus the payment), and want to maximize its own utility.
- **worker's buffer size**: every worker have limits space or ability to finish limited worker, for which will make sure it suitable for reality.
- **the probability of accomplish a task**: this relates to worker's location and it's ability or willing to finish the task;
- **the number of have accomplished a kind of task**: there will exits several kinds of task, we will record the worker have finished how many times for every kind of task.
- **centrality**: this will be viewed as a factor, which may affect the requester's decision.

For this feature, we will do the normalization to make the feature's value belong to [0,1]. Other than these features, no other information was used to identify context.

8.3 Performance

This section we will show the result of experiments.

9 Conclusion

In this paper, we study the task allocation problem for crowdsensing. In order to select the best worker from the requester's contacts, we proposed thompson sampling allocation algorithm with incentive mechanism. This algorithm can make sure long run and keep the regret stay low. We showed that intuition from crowdsourcing mechansim is useful, and our algorithm follow the DISC and IR constrains. Analysis of TS-TA showed that it achieves regret of .

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

Appendix B

Appendix two text goes here.

Acknowledgments

The authors would like to thank...

REFERENCES

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

[2] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. *arXiv preprint arXiv:1111.1797*, 2011.

[3] Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. *arXiv preprint arXiv:1209.3352*, 2012.

[4] Bahram Alidaee and Haitao Li. Parallel machine selection and job scheduling to minimize sum of machine holding cost, total machine time costs, and total tardiness costs. *Automation Science and Engineering, IEEE Transactions on*, 11(1):294–301, 2014.

[5] Ali Allahverdi, CT Ng, TC Edwin Cheng, and Mikhail Y Kovalyov. A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187(3):985–1032, 2008.

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

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

[8] Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721*, 2012.

[9] Giuseppe Cardone, Luca Foschini, Paolo Bellavista, Antonio Corradi, Cristian Borcea, Manoop Talasila, and Reza Curtmola. Fostering participaction in smart cities: a geo-social crowdsensing platform. *Communications Magazine, IEEE*, 51(6):112–119, 2013.

[10] Chandra Chekuri, Rajeev Motwani, Balas Natarajan, and Clifford Stein. Approximation techniques for average completion time scheduling. *SIAM Journal on Computing*, 31(1):146–166, 2001.

[11] TCE Cheng and CCS Sin. A state-of-the-art review of parallel-machine scheduling research. *European Journal of Operational Research*, 47(3):271–292, 1990.

[12] Azade Farshad, Mahesh K Marina, and Francisco Garcia. Urban wifi characterization via mobile crowdsensing. In *Network Operations and Management Symposium (NOMS), 2014 IEEE*, pages 1–9. IEEE, 2014.

[13] Zhenni Feng, Yanmin Zhu, Qian Zhang, Lionel M Ni, and Athanasios V Vasilakos. Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing. In *INFOCOM, 2014 Proceedings IEEE*, pages 1231–1239. IEEE, 2014.

[14] Raghu K Ganti, Fan Ye, and Hui Lei. Mobile crowdsensing: current state and future challenges. *IEEE Communications Magazine*, 49(11):32–39, 2011.

[15] Shibo He, Dong-Hoon Shin, Junshan Zhang, and Jiming Chen. Toward optimal allocation of location dependent tasks in crowdsensing. In *INFOCOM, 2014 Proceedings IEEE*, pages 745–753. IEEE, 2014.

[16] Shweta Jain, Balakrishnan 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.

[17] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[18] Yang Liu and Mingyan 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*, pages 217–230. ACM, 2015.

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

[20] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[21] Mingjun Xiao, Jie Wu, Liusheng Huang, Yunsheng Wang, and Cong Liu. Multi-task assignment for crowdsensing in mobile social networks. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*, pages 2227–2235. IEEE, 2015.