Reward Maximization for Task allocation in Opportunistic crowdsensing networks

**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.

Reward Maximization for Task allocation in Opportunistic crowdsensing networks(随机群治感知网络中任务分配reward最大化)

**Introduction**

In a bandit problem, there are m armed-bandits, every time you pull the arm, you will get a reward, while the reward is random. We can you formulate this problem as following.

**Single-Agent Multi-Armed Bandit**

Single-agent multi-armed bandit (SA-MAB) problem was first introduced in [1]. In the most seminal setting, the problem models an agent facing the challenge of sequentially selecting an arm from a set of arms in order to receive an a priori unknown reward, drawn from some unknown reward generating process. As a result of lack of prior information, at each trial, the player may choose some inferior arm in terms of reward, yielding some regret that is quantified by the difference between the reward that would have been achieved had the player selected the best arm and the actual achieved reward. In such an unknown setting, the player decides which arm to pull in a sequence of trials so that its accumulated regret over the game horizon is minimized. This problem is an instance of exploration-exploitation dilemma, i.e., the tradeoff between taking actions that yield immediate large rewards on the one hand and taking actions that might result in larger reward only in future, for instance activating an inferior arm only to acquire information, on the other hand. Therefore, SAMAB is mainly concerned with a sequential online decision making problem in an unknown environment. A solution of a bandit problem is thus a decision making strategy called policy or allocation rule, which determines which arm should be played at successive rounds. Policies are in general evaluated based on their regret or discounted reward performance.

As mentioned before, MAB benefits from a wide range of variations in the setting. In the event that arms have different states, the reward depends on arms’ states, and the bandit model is referred to as stateful (Markovian). Otherwise, the model is stateless, which itself can be divided to few subsets. As stateful and stateless bandits are inherently different, we discuss them separately in the following.

**Model**

In this section, we formally define the one-player M-armed bandit problem. Let denote the set of armed-bandit, or called the set of actions. The bandits have m probability distributionswith associated expected valuesand variances,whereis known but the parameterare unknown. At the same time the random variablehas an unknown probability distribution. A player at time t chooses an action aM, with the payoff/reward denoted by. We assume are independent for each a and t. Furthermore, the distributionshave a bounded support of [0,1] for each a. An allocation policy (or allocation rule)specifies the action chosen by the player at each time in the system. Formally, is a sequence of random variables, where is the action chosen by player i at time t. Let be the reward obtained by the strategy for player i at time t. Let the historybe denote by.

The player is viewed as a gambler whose goal is to collect as much money as possible by pulling these arms over many turns. At each turn, t=1,2,…, the player selects an arm, with index j(t), and receives reward r(t)~ . The player has a two-fold goal: on one hand, finding out which distribution has the highest expected value; on the other hand, gaining as much reward as possible while playing. Bandit algorithms specify a strategy by which the player should choose an arm j(t) at each turn.

The regret of strategyat time t for a fixed is defined by



Where  is the expected reward from the best arm,  is a random variable denoting the number of plays of arm a during the first t turns.

A classical result of Lai and Robbins(1985) states that for any suboptimal arm a,



Where  is the Kullback-Leibler divergence between the reward density  of the suboptimal arm and the reward density  of the optimal arm, defined formally as



Regret thus grows at least logarithmically, or more formally, . An algorithm is said to solve the multi-armed bandit problem if it can match this lower bound, that is if .

**Related work**

**Problem Formulation**

**Simulation**

**Conclusion**

**Reference**

In the following section, we turn our attention to an important application of bandit algorithms: task allocation. Indeed, a task allocation perfectly captures the problem of balancing exploration and exploitation: we are look for a way to simultaneously identify the best relay node(the best “arm”) and ensure that as much allocator as possible can benefit from it.