Pierre Coucheney, Johanne Cohen, Kinda Khawam
Université de Versailles Saint-Quentin en Yvelines
Laboratoire DAVID

Choosing k best arms in adversarial bandit setting : application to inter-cell Interference Coordination

Décembre 2015,
GDT COS

Framework

- Cellular network is made of several base stations (BS) ;
- the time and frequency radio resources are grouped into time-frequency Resource Blocks RBs (the smallest time-frequency resource unit allotted to a mobile user)
- when the same RB is used in neighboring cells, interference may occur which can degrade the Signal to Interference plus Noise Ratio (SINR) perceived by mobile users ;
- each BS is a player that has to pick several RBs ;
- each BS seeks to minimize its regret ;
- other BSs are seen as a single adversary that chooses the sequence of rewards.

EXP3 Algorithm

Let $\gamma \in (0,1)$ and $\omega(0) = 0$.
Until $t = T$, do :

- select action A with probability

$$p_i(t) = (1-\gamma)\frac{\exp(\gamma / K\, \omega_i(t))}{\sum_j \exp(\gamma / K\, \omega_j(t))} + \gamma \frac{1}{K}$$

- receive reward $r_A(t+1)$ and compute

$$g_i(t+1) := \begin{cases} r_A(t+1)/p_i(t) & \text{if } A = i \\ 0 & \text{otherwise} \end{cases}$$

- update ω

$$\omega(t+1) = \omega(t) + g(t+1)$$

EXP3 Algorithm

- T is the horizon. By appropriately choosing $\gamma := \gamma(T, k)$, the expected regret is $O(\sqrt{T K \log(K)})$.
- If $K = \binom{n}{k}$ then the expected regret is $O\left(\sqrt{T n^k k \log(n)}\right)$ and space complexity $O(n^k)$.

Adapt EXP3 algorithm for choosing k among n

- suppose reward is additive

$$r_{i_1 \dots i_k} = \sum_{\ell=1}^{\ell=k} r_{i_\ell}$$

- EXP3 can be uncoupled and run with ω_i instead of $\omega_{i_1 \dots i_k}$ (space complexity $O(n)$)
- The expected regret is $O\left(\sqrt{T k n \log \binom{n}{k}}\right)$

Adapted EXP3

Let $\gamma \in (0,1)$, $\omega(0) = 0$, and $K = \binom{n}{k}$.
Until $t = T$, do :

- select k-tuple $A := (A_1 \dots A_k)$ with probability

$$p_{i_1 \dots i_k} = (1-\gamma) \frac{x_{i_1} \dots x_{i_k}}{\sum_{(j_1 \dots j_k)} x_{j_1} \dots x_{j_k}} + \gamma \frac{1}{K}$$

  where $x_i := \exp(\gamma / n \, \omega_i)$

- receive reward $r_i(t+1)$ for each chosen $i \in A$ and compute

$$g_i(t+1) := \begin{cases} \frac{r_i(t+1)}{(1-\gamma) p_i(t) + \gamma \, k/n} & \text{if } i \in A \\ 0 & \text{otherwise} \end{cases}$$

- update ω

$$\omega(t+1) = \omega(t) + g(t+1)$$

Implementation

- Problem : Simulate the k-tuple $A := (A_1 \dots A_k)$ with probability

$$p_{i_1 \dots i_k} = (1-\gamma) \frac{x_{i_1} \dots x_{i_k}}{\sum_{(j_1 \dots j_k)} x_{j_1} \dots x_{j_k}} + \gamma \frac{1}{K}$$

- Compute the marginal law p_i.
- Need to compute $\sum_{(j_1 \dots j_k)} x_{j_1} \dots x_{j_k}$ with $\binom{n}{k}$ terms.
- Example : with $k = 2$, choose i with probability

$$(1-\gamma) \frac{x_i(1-x_i)}{\sum_j x_j(1-x_j)} + \gamma \frac{1}{n}$$

Compute $\sum_{(j_1 \ldots j_k)} x_{j_1} \ldots x_{j_k}$

Dynamic programming :

- Let $S(\ell, m)$, $0 \leq \ell \leq k$ and $0 \leq m \leq n$ the sum

$$\sum_{(j_1 \ldots j_\ell)} x_{j_1} \ldots x_{j_\ell}$$

  with $j_\ell \in \{1 + (n-m), \ldots, n\}$
  and $S(0, m) = 1$.
- Then, if $\ell > 0$ and $m > 0$,

$$S(\ell, m) = x_{1+(n-m)} S(\ell - 1, m - 1) + S(\ell, m - 1)$$

- Hence computing $S(k, n)$ is $O(k\,n)$.
- Simulating our random variable needs $O(k^2\,n)$.

A more general problem : combinatorial bandits

- choose action a in a finite set $S \subseteq \{0,1\}^n$,
- receive reward $r_a = \sum_{i \in a} r_i$,
- each r_i is unknown,
- there is algorithm with expected regret $O(\sqrt{Tn\log(|S|)})$.

Combinatorial bandits, Cesa-Bianchi and Lugosi, 2012

Examples :

- spanning tree
- path
- permutation
- k among n

Choosing 2 among n

Let $\gamma \in (0,1)$, $\omega(0) = 0$, and $K = \binom{n}{2}$.
Until $t = T$, do :

- select 2-tuple $A := (A_1, A_2)$ with probability

$$p_{i_1,i_2} = (1-\gamma)\frac{\exp(\eta\,\omega_{i_1,i_2}(t))}{\sum_{i,j}\exp(\eta\,\omega_{i,j}(t))} + \gamma\frac{1}{K}$$

- receive reward $r_A(t+1)$ and compute

$$g_{i_1,i_2}(t+1) := \begin{cases} \frac{r_A(t+1)}{p_{i_1,i_2}} & \text{if } (i_1,i_2) = A \\ 0 & \text{if } i_1 \in A \text{ and } i_2 \notin A \text{ or } i_2 \in A \text{ and } i_1 \notin A \\ -\frac{r_A(t+1)}{p_{i_1,i_2}} & \text{if } i_1 \notin A \text{ and } i_2 \notin A \end{cases}$$

- update ω

$$\omega(t+1) = \omega(t) + g(t+1)$$

Numerical Problem

In EXP3 algorithm :

$$\omega(t+1) = \omega(t) + g(t+1),$$

and

$$p_i(t) = (1-\gamma)\frac{\exp(\gamma/K\,\omega_i(t))}{\sum_j \exp(\gamma/K\,\omega_j(t))} + \gamma\frac{1}{K}$$

$\gamma\,\omega_i(T)$ can grow as $\sqrt{T}$ and exp may produce too large numbers.

Idea : replace it by

$$\omega(t+1) = (1-\alpha)\omega(t) + g(t+1),$$

and

$$p_i(t) = \frac{\exp(\gamma\,\omega_i(t))}{\sum_j \exp(\gamma\,\omega_j(t))}$$

with $0 < \alpha < 1$ and γ well chosen...