Multi Armed Bandits

Alireza Shafaei

Machine Learning Reading Group
The University of British Columbia
Summer 2017

Outline

Online Convex Optimization

Definition

- At each iteration t, the player chooses $x_t \in \mathcal{K}$.
- A convex loss function $f_t \in \mathcal{F} : \mathcal{K} \to \mathbb{R}$ is revealed.
- A cost $f_t(x_t)$ is incurred.
  - $\to$ $\mathcal{F}$ is a set of bounded functions.
  - $\to$ f_t is revealed after choosing x_t.
  - $\to$ f_t can be adversarially chosen.

Online Convex Optimization

Regret

- Given an algorithm $\mathcal{A}$.
- *Regret* of $\mathcal{A}$ after T iterations is defined as:

$$\mathrm{regret}_T(\mathcal{A}) := \sup_{\{f_i\}_{i=1}^{T} \in \mathcal{F}} \{\sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x)\}$$

- Online Gradient Descent $\mathrm{O}(GD\sqrt{T})$.
- If f_i is α-strongly convex then $\mathrm{O}(\frac{G^2}{2\alpha}(1 + \log(T))$.

Bandit Convex Optimization

Definition[1]

- In OCO we had access to $\nabla f_t(x_t)$.
- in BCO we only observe $f_t(x_t)$.
- Multi Armed Bandit further constrains the BCO setting.

[1] Material from [1].

Bandit Convex Optimization

Motivation

- In data networks, the decision maker can measure the RTD of a packet, but rarely has access to the congestion pattern of the entire network.
- In ad-placement, the search engine can inspect which ads were clicked through, but cannot know whether different ads, had they been chosen, would have been click through or not.
- Given a fixed budget, how to allocate resources among the research projects whose outcome is only partially known at the time of allocation and may change through time.
- *Wikipedia*. Originally considered by Allied scientists in World War II, it proved so intractable that, according to Peter Whittle, the problem was proposed to be dropped over Germany so that German scientists could also waste their time on it.

Multi Armed Bandit (MAB)

Definition

- On each iteration t, the player choses an action i_t from a predefined set of discrete actions $\{1, \ldots, n\}$.
- An adversary, independently, chooses a loss $\in [0, 1]$ for each action.
- The loss associated with i_t is then revealed to the player.
- There are a variety of MAB specifications with various assumptions and constraints.
- This definition is similar to the multi-expert problem, except we do not observe the loss associated with the other experts.

MAB as a BCO

- The algorithms usually choose an action w.r.t a distribution over the actions.
- If we define $\mathcal{K} = \Delta_n$, an n-dimensional simplex then

$$f_t(x) = \ell_t^\top x = \sum_{i=1}^{n} \ell_t(i)x(i) \qquad \forall x \in \mathcal{K}$$

- We have an *exploration-exploitation* trade-off.
- A simple approach would be to
  - → Exploration With some probability, explore by choosing actions uniformly at random. Construct an estimate of the actions' losses with the feedback.
  - → Exploitation Otherwise, use the estimates to make a decision.

A Simple MAB Algorithm

Definition

- An algorithm can be constructed:

Algorithm 17 Simple MAB algorithm

```
Input: OCO algorithm 𝒜, parameter δ.
for t = 1 to T do
    Let b_t be a Bernoulli random variable that equals 1 with probability δ.
    if b_t = 1 then
        Choose i_t ∈ {1, 2, ..., n} uniformly at random and play i_t.
        Let
```

$$\hat{\ell}_t(i) = \begin{cases} \frac{n}{\delta} \cdot \ell_t(i_t), & i = i_t \\ 0, & \text{otherwise} \end{cases}.$$

```
        Let f̂_t(x) = ℓ̂_t^⊤ x and update x_{t+1} = 𝒜(f̂_1, ..., f̂_t).
    else
        Choose i_t ~ x_t and play i_t.
        Update f̂_t = 0, ℓ̂_t = 0, x_{t+1} = 𝒜(f̂_1, ..., f̂_t).
    end if
end for
```

A Simple MAB Algorithm

Analysis

- This algorithm guarantees:

$$\mathbb{E}[\sum_{t=1}^{T} \ell_t(i_t) - \min_i \sum_{t=1}^{T} \ell_t(i)] \leq \mathrm{O}(T^{\frac{3}{4}}\sqrt{n})$$

$$\mathbb{E}[\hat{\ell}_t(i)] = \mathbb{P}[b_t = 1] \cdot \mathbb{P}[i_t = i | b_t = 1] \cdot \frac{n}{\delta}\ell_t(i) = \ell_t(i)$$

$$\|\hat{\ell}_t\|_2 \leq \frac{n}{\delta} \cdot |\ell_t(i_t)| \leq \frac{n}{\delta}$$

- $\mathbb{E}[\hat{f}_t] = f_t$

A Simple MAB Algorithm

Analysis

$$
\begin{aligned}
& \mathbf{E}[\text{regret}_T] \\
&= \mathbf{E}[\textstyle\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\Delta_n} \sum_{t=1}^{T} f_t(\mathbf{x})] \\
&= \mathbf{E}[\textstyle\sum_{t=1}^{T} \ell_t(i_t) - \min_i \sum_{t=1}^{T} \ell_t(i)] \\
&= \mathbf{E}[\textstyle\sum_{t=1}^{T} \ell_t(i_t) - \sum_{t=1}^{T} \ell_t(i^\star)] \\
&\leq \mathbf{E}[\textstyle\sum_{t\notin S_T} \hat{\ell}_t(i_t) - \sum_{t\notin S_T} \hat{\ell}_t(i^\star) + \sum_{t\in S_t} 1] && i^\star \text{ is indep. of } \hat{\ell}_t \\
&\leq \mathbf{E}[\textstyle\sum_{t\notin S_T} \hat{\ell}_t(i_t) - \min_i \sum_{t\notin S_T} \hat{\ell}_t(i) + \sum_{t\in S_t} 1] \\
&\leq \tfrac{3}{2} GD\sqrt{T} + \delta\cdot T && \text{Theorem 3.1} \\
&\leq 3G\sqrt{T} + \delta\cdot T && \text{For } \Delta_n,\ D \leq 2 \\
&\leq 3\tfrac{n}{\delta}\sqrt{T} + \delta\cdot T && \|\ell_t\| \leq \tfrac{n}{\delta} \\
&= O(T^{\frac{3}{4}}\sqrt{n}). && \delta = \sqrt{n}T^{-\frac{1}{4}}
\end{aligned}
$$

EXP3

Algorithm 18 EXP3 - simple version

1: Input: parameter $\varepsilon > 0$. Set $\mathbf{x}_1 = (1/n)\mathbf{1}$.
2: **for** $t \in \{1, 2, ..., T\}$ **do**
3: Choose $i_t \sim \mathbf{x}_t$ and play i_t.
4: Let

$$\hat{\ell}_t(i) = \begin{cases} \frac{1}{\mathbf{x}_t(i_t)} \cdot \ell_t(i_t), & i = i_t \\ 0, & \text{otherwise} \end{cases}$$

5: Update $\mathbf{y}_{t+1}(i) = \mathbf{x}_t(i) e^{-\varepsilon \hat{\ell}_t(i)}$, $\mathbf{x}_{t+1} = \frac{\mathbf{y}_{t+1}}{\|\mathbf{y}_{t+1}\|_1}$
6: **end for**

- Has a worst-case near-optimal regret bound of $O(\sqrt{Tn \log n})$.
- See P104 of the OCO book for proof.

Stochastic Multi Armed Bandit

Definition

- On each iteration t, the player choses an action i_t from a predefined set of discrete actions $\{1, \ldots, n\}$.
- Each action i has an underlying (fixed) probability distribution $\mathbb{P}_i$ with mean μ_i.
- The loss associated with i_t is then revealed to the player. (A sample is taken from $\mathbb{P}_{i_t}$).
- $\mathbb{P}_i$'s could be a simple Bernoulli variable.
- A more complex version could assume a Markov process for each action, within which the state of one or all processes change after each iteration.
- We still have the *exploration-exploitation* trade-off.

Bernoulli Multi Armed Bandit

Definition[2]

Algorithm 1 Bernoulli Multi Arm Bandit

for a in $1..K$ **do**
  $Q[a] = 0,\ N[a] = 0,\ S[a] = 0,\ F[a] = 0$
end for
for t in $1..T$ **do**
  $a = PickArm(Q, N, S, F)$
  $r = BernoulliReward(a)$
  $N[a] = N[a] + 1$
  $Q[a] = Q[a] + \frac{1}{N[a]}(r - Q[a])$
  $S[a] = S[a] + r$
  $F[a] = F[a] + (1 - r)$
end for

- $N[a]$ The number of times arm a is pulled.
- $Q[a]$ The running average of rewards for arm a.
- $S[a]$ The number of successes for arm a.
- $F[a]$ The number of failures for arm a.
- The same notion of regret, except the optimal strategy is to pull the arm with the largest mean, μ^*.

[2] Material from [2].

Algorithms

- Random Selection $\Rightarrow O(T)$.
- Greedy Selection $\Rightarrow O(T)$.
- ϵ-Greedy Selection $\Rightarrow O(T)$.
- Boltzmann Exploration $\Rightarrow O(T)$.

$$P(a) = \frac{\exp(Q[a]/\tau)}{\sum_a \exp(Q[a]/\tau)}$$

- Upper-Confidence Bound $\Rightarrow O(\ln(T))$.

$$A = argmax_a \left[Q[a] + \sqrt{\frac{2\log t}{N[a]}} \right]$$

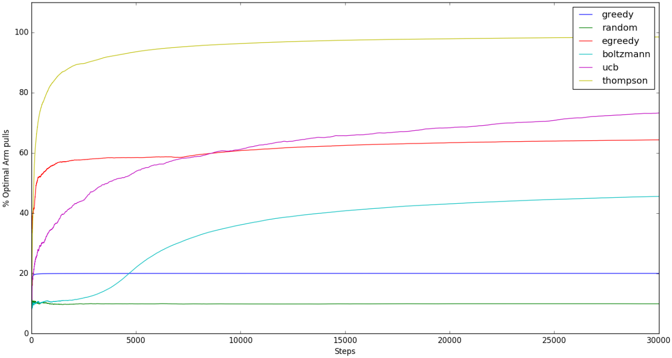
- Thompson Sampling $\Rightarrow O(\ln(T))$.

$$\text{For } a \text{ in } 1..K: \quad \theta[a] \sim Beta(S[a]+1, F[a]+1)$$

$$A = argmax_a(\theta[a])$$

Algorithms

Empirical Evaluation



Contextual Bandits

Motivation

- There are scenarios within which we can have access to more information.
- The extra information can be encoded as a context vector.
- In online advertising, the behaviour of each user, or the search context for instance, can provide valuable information.
- One simple way is to treat each context having its own bandit problem.
- Variations of the previous algorithms relate the context vector with the expected reward through linear models, neural networks, kernels, or random forests.

Thanks

Thanks!
Questions?

References I

E. Hazan.
Introduction to Online Convex Optimization.
http://ocobook.cs.princeton.edu/OCObook.pdf

S. Raja.
Multi Armed Bandits and Exploration Strategies.
https://sudeepraja.github.io/Bandits/