Channel Capacity Allocation & Convex Optimization

Kasra Khoshjoo

Sharif University of Technology

Lightning Day
September 5, 2023

Presentation Overview

1. Priliminaries
2. Convexity of our setting
3. Solving the bandit problem

Problem Setting

- The Topology of the network is fixed
- Predetermined Budget : $\sum c(e_i) \leq B$
- for each pair (s, t) the amount of each transaction is an independent variable with distribution $P_{s,t}(\tau) = \lambda_{s,t} e^{-\lambda_{s,t}\tau}$

A potential Solution

Stochastic convex optimization with bandit feedback (Agarwal et al., 2011)

Convex Optimization

consider the following bandit problem:

- arms are vectors from the convex set $\mathcal{X}$
- the feedback for each arm x has the same distribution as $f(x) + \epsilon$
- f is a 1-Libschitz convex function
- ϵ is sub-gaussian. i.e. $\forall x > 0 : \mathbb{P}\left[|\epsilon| > x\right] \leq 2e^{-\frac{x^2}{2\sigma^2}}$

the algorithm is guaranteed to have a regret in $\tilde{O}\left(\text{poly}(d)\sqrt{T}\right)$

Convexity of our setting

Does our problem satisfy these criteria?

Convexity

- convexity of the feasible set is straight forward
- consider the expected reward of each transaction:

$$
\begin{aligned}
F \triangleq\; & \\
& \sum_{s,t} \mathbb{P}\left[S = s, T = t\right] \mathbf{1}_{\{v \text{ is in the shortest path of } (s,t)\}} \\
& \quad \int_0^\infty \mathbb{P}_{s,t}\left[\text{amount} = \tau\right] \mathbf{1}_{\{\tau < C(e_s), \tau < C(e_t)\}} \text{reward}(\tau) d\tau
\end{aligned}
$$

Convexity

$$F = \sum_{s,t} \mathbb{P}\left[S = s, T = t\right] \mathbb{1}_{\{v \text{ is in the shortest path of } (s,t)\}} \left[\int_0^{\min\{C(e_s),C(e_t)\}} \lambda_{s,t} e^{-\lambda_{s,t}\tau} d\tau\right]$$

we have :

$$\frac{d^2}{dx^2} \int_0^x \lambda e^{-\lambda\tau} \tau d\tau = \lambda e^{-\lambda x}\left[-\lambda x + 1\right]$$

given that $\lambda_{s,t} \min\{C(e_s), C(e_t)\} < 1$, F as a function of capacities is concave

Solving a convex optimization bandit problem

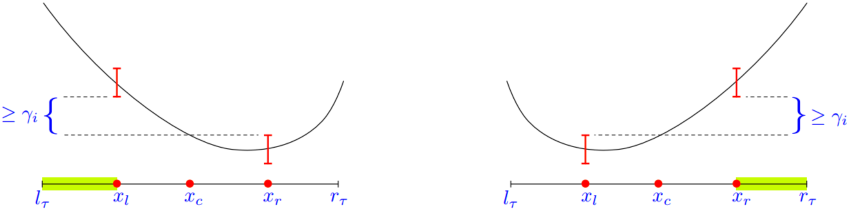


Figure 1: Two possible configurations when the algorithm enters case 1.

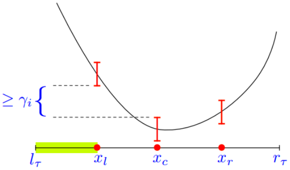


Figure 2: One of the possible configurations when the algorithm enters Case 2.

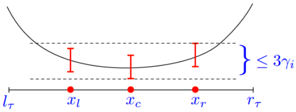


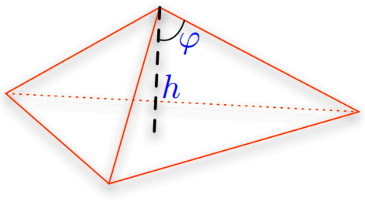
Figure 3: Configuration of the confidence intervals in Case 3 of Algorithm 1.

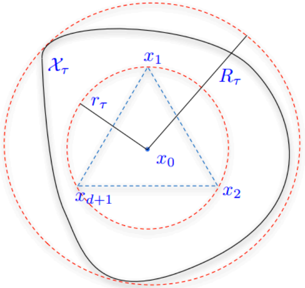
Algorithm 1 One-dimensional stochastic convex bandit algorithm

input noisy black-box access to $f\colon [0,1] \to \mathbb{R}$, total number of queries allowed T.

Let $l_1 := 0$ and $r_1 := 1$.
for epoch $\tau = 1, 2, \ldots$ **do**
  Let $w_\tau := r_\tau - l_\tau$.
  Let $x_l := l_\tau + w_\tau/4$, $x_c := l_\tau + w_\tau/2$, and $x_r := l_\tau + 3w_\tau/4$.
  for round $i = 1, 2, \ldots$ **do**
    Let $\gamma_i := 2^{-i}$.
    For each $x \in \{x_l, x_c, x_r\}$, query $f(x)$ $\frac{2\sigma}{\gamma_i^2} \log T$ times. sampling
    if $\max\{\mathrm{LB}_{\gamma_i}(x_l), \mathrm{LB}_{\gamma_i}(x_r)\} \geq \min\{\mathrm{UB}_{\gamma_i}(x_l), \mathrm{UB}_{\gamma_i}(x_r)\} + \gamma_i$ **then**
      {**Case 1**: `CI's at` x_l `and` x_r `are` γ_i `separated`}
      if $\mathrm{LB}_{\gamma_i}(x_l) \geq \mathrm{LB}_{\gamma_i}(x_r)$ **then** let $l_{\tau+1} := x_l$ and $r_{\tau+1} := r_\tau$.
      if $\mathrm{LB}_{\gamma_i}(x_l) < \mathrm{LB}_{\gamma_i}(x_r)$ **then** let $l_{\tau+1} := l_\tau$ and $r_{\tau+1} := x_r$.
      Continue to epoch $\tau + 1$.
    else if $\max\{\mathrm{LB}_{\gamma_i}(x_l), \mathrm{LB}_{\gamma_i}(x_r)\} \geq \mathrm{UB}_{\gamma_i}(x_c) + \gamma_i$ **then**
      {**Case 2**: `CI's at` x_c `and` x_l `or` x_r `are` γ_i `separated`}
      if $\mathrm{LB}_{\gamma_i}(x_l) \geq \mathrm{LB}_{\gamma_i}(x_r)$ **then** let $l_{\tau+1} := x_l$ and $r_{\tau+1} := r_\tau$.
      if $\mathrm{LB}_{\gamma_i}(x_l) < \mathrm{LB}_{\gamma_i}(x_r)$ **then** let $l_{\tau+1} := l_\tau$ and $r_{\tau+1} := x_r$.
      Continue to epoch $\tau + 1$.
    end if
  end for
end for

Higher Dimensions





- smallest euclidian ball containing $\mathcal{X}_\tau$ has radius R_τ
- $r_\tau \triangleq \frac{R_\tau}{c_1 \cdot d}$, $\mathcal{B}(r_\tau)$ is an euclidian ball of radius r_τ where $\mathcal{B}(r_\tau) \subseteq \mathcal{X}_\tau$

Cone Cutting

Algorithm 3 CONE-CUTTING

input pyramid Π with apex y, (rounded) feasible region $\mathcal{X}_\tau$ for epoch τ, enclosing ball $\mathcal{B}_\tau$

1: Let $z_1, \ldots, z_d$ be the vertices of the base of Π, and $\bar{\varphi}$ the angle at its apex.

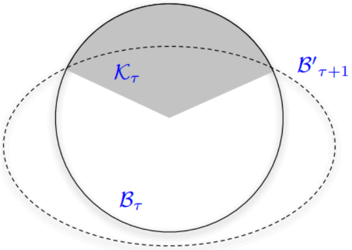
2: Define the cone

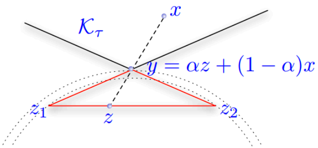
$$\mathcal{K}_\tau = \{x \mid \exists \lambda > 0, \alpha_1, \ldots, \alpha_d > 0, \sum_{i=1}^{d} \alpha_i = 1 \ : \ x = y - \lambda \sum_{i=1}^{d} \alpha_i (z_i - y)\}$$

3: Set $\mathcal{B}'_{\tau+1}$ to be the min. volume ellipsoid containing $\mathcal{B}_\tau \setminus \mathcal{K}_\tau$.

4: Set $\mathcal{X}_{\tau+1} = \mathcal{X}_\tau \cap \mathcal{B}'_{\tau+1}$.

output new feasible region $\mathcal{X}_{\tau+1}$ and enclosing ellipsoid $\mathcal{B}'_{\tau+1}$.





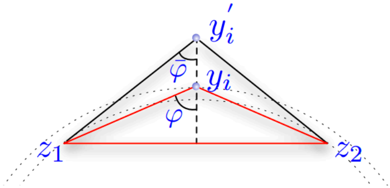
Hat Raising

Algorithm 4 HAT-RAISING

input pyramid Π with apex y, (rounded) feasible region $\mathcal{X}_\tau$ for epoch τ, enclosing ball $\mathcal{B}_\tau$.

1: Let CENTER be the center of Π.
2: Set $y' = y + (y - \text{CENTER})$.
3: Set Π' to be the pyramid with apex y' and same base as Π.
4: Set $(\mathcal{X}_{\tau+1}, \mathcal{B}'_{\tau+1}) = \text{CONE-CUTTING}(\Pi', \mathcal{X}_\tau, \mathcal{B}_\tau)$.

output new feasible region $\mathcal{X}_{\tau+1}$ and enclosing ellipsoid $\mathcal{B}'_{\tau+1}$.



Algorithm for Higher Dimensions