Follow the Perturbed leader

Thursday, November 11, 2021 7:54 PM

Yiren

Online Decision Making

time periods $1, 2, 3, \ldots, t, \ldots, T$

decision $d_1, d_2, \ldots, d_t, \quad, d_T \in \mathbb{R}^n$

Observation/state $s_1, s_2, \ldots, s_t, \ldots, s_T \in \mathbb{R}^n$

Cost $d_1 \cdot s_1 \quad \ldots \quad d_t \cdot s_t \quad \cdots \quad d_T \cdot s_T$

Goal Minimize $\sum_{t=1}^{T} d_t \cdot s_t$

But d_t is chosen before knowing s_t!

e.g. experts

shortest paths

online Linear Programming

Search trees

....

⋮

...

Compare with best decision in hindsight.

But then impossible to match?

Adversary can see the future but has to pick a single decision and stick to it!

$$M(s) = \operatorname*{argmin}_{d \in \Omega} \; d \cdot s$$

$$\text{Adversary's cost} = \min_{d} \sum_{t} d \cdot s_t$$

$$= \min_{d} \; d \cdot \sum_{t} s_t$$

$$= M\left(\sum_{t=1}^{T} s_t\right) \cdot \left(\sum_{t=1}^{T} s_t\right).$$

GOAL: Algorithm's cost should be close, i.e. not much higher. Algorithm can make a different decision each day. Regret = Cost(Algo) − Cost(Adversary)

How? Is it possible?!

Let $s_{1:t} = s_1 + s_2 + \dots + s_t$.

"Follow the Leader"

set $d_t = M(s_1 + \dots + s_{t-1}) = M(s_{1:t-1})$

Does this work? No!

"Be the leader"

$$d_t = M(s_{1:t})$$

YES!

But, cannot implement as we don't know s_t.

To be sure,

Claim $$\sum_{t=1}^{T} M(s_{1:t}) \cdot s_{1:t} \leq M(s_{1:T}) \cdot s_{1:T}$$

PF. induction on T.

$T=1$ ✓

$T+1$: $$\sum_{t=1}^{T+1} M(s_{1:t}) \cdot s_{1:t} \leq M(s_{1:T}) \cdot s_{1:T} + M(s_{1:T+1}) \cdot s_{T+1}$$

$$\leq M(s_{1:T+1}) \cdot s_{1:T} + M(s_{1:T+1}) \cdot s_{T+1}$$

$$= M(s_{1:T+1}) \cdot s_{1:T+1}$$

Back to what should we do??!

"Follow the Perturbed Leader"

- Let $p_0 \sim \left[0, \frac{1}{\varepsilon}\right]^n$ uniform random in a cube.
- $d_t = M(s_{1:t-1} + p_0)$

Thm. $\mathbb{E}(FPL(\varepsilon)) \leq \text{min-cost}_T + \varepsilon RAT + \frac{D}{\varepsilon}$

$D \geq \text{diam}(\Omega) = \max_{d, d' \in \Omega} \|d - d'\|_1$

$R \geq \max d \cdot s$ $\qquad$ $\text{min-cost}_T = M(s_{1:T}) \cdot s_{1:T}$

$$A \geq \|s\|_1$$

Setting ε optimally, $\mathbb{E}(\text{FPL}) \leq \text{mincost}_T + 2\sqrt{DRAT}$

To get high probability bound, use

$$d_t = M(s_{1:t-1} + p_t).$$

Lemma. $$\sum_{t=1}^{T} M(s_{1:t} + p_t) \cdot s_t$$

$$\leq M(s_{1:T}) \cdot s_{1:T} + D \sum_{t=1}^{T} |p_t - p_{t-1}|_\infty$$

Pretend state $\tilde{s}_t = s_t + p_t - p_{t-1}$

Note $\tilde{s}_{1:t} = s_{1:t} + p_t$

$$\sum_{t=1}^{T} M(s_{1:t} + p_t) \cdot (s_t + p_t - p_{t-1})$$

$$\leq M(s_{1:T} + p_T) \cdot (s_{1:T} + p_T)$$

$$\leq M(s_{1:T})\cdot(s_{1:T} + p_T)$$

$$= M(s_{1:T})\cdot s_{1:T} + \sum_{t=1}^{T} M(s_{1:T})\cdot(p_t - p_{t-1})$$

$$\therefore \sum_{t=1}^{T} M(s_{1:t} + p_t)\cdot s_t \leq M(s_{1:T})\cdot s_{1:T}$$

$$+ \sum_{t=1}^{T} \left(M(s_{1:T}) - M(s_{1:t}+p_t)\right)\cdot(p_t - p_{t-1})$$

$$\leq M(s_{1:T})\cdot s_{1:T} + D \sum_{t=1}^{T} \|(p_t - p_{t-1})\|_\infty.$$

Now the idea is that $s_{1:t} + p_t$

and $s_{1:t-1} + p_t$

have a similar distribution!

By the lemma, since $p_t = p_1$

$$\sum_{t=1}^{T} M(s_{1:t} + p_t)\cdot s_t \leq M(s_{1:T})\cdot s_{1:T} + D\|p_1\|_\infty$$

Compare LHS with $\sum_{t=1}^{T} M(s_{1:t-1} + p_t)\cdot s_t$

Suppose the distributions of $X = s_{1:t} + p_t$

and $Y = s_{1:t-1} + p_t$

can be <u>coupled</u> so that $X = Y$ with prob q

then

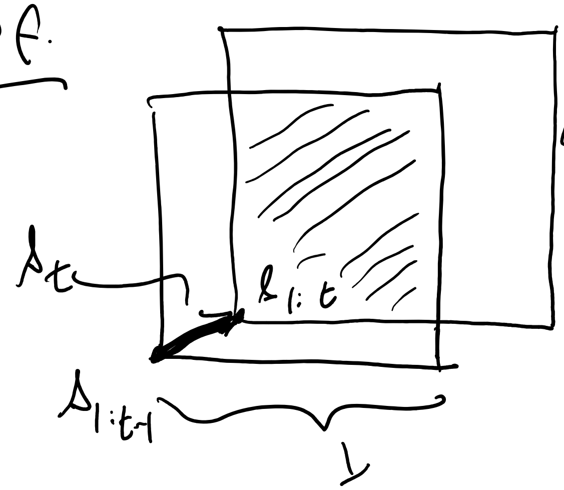
$$\mathbb{E}\left(M(s_{1:t-1} + p_t) \cdot s_t\right) \le \mathbb{E}\left(M(s_{1:t} + p_t) \cdot s_t\right) + (1-q) R$$

So $\mathbb{E}(FPL) \le M(s_{1:T}) \cdot s_{1:T} + \frac{D}{\varepsilon} + (1-q) RT$

<u>Claim</u>, $1 - q \le \varepsilon \|s_t\|_1 \le \varepsilon A$.

$$\Rightarrow \mathbb{E}(FPL) \le \text{mincost} + \frac{D}{\varepsilon} + \varepsilon \cdot RAT.$$

<u>Pf.</u>



$$q = \text{Overlap} = \prod_j \left(1 - \frac{|s_{t\,j}|}{\frac{1}{\varepsilon}}\right)$$

$$\ge e^{-\varepsilon \sum_j |s_{t\,j}|} = e^{-\varepsilon |s_t|_1}$$

$$1 - q \le 1 - e^{-\varepsilon |s_t|_1}$$

$e^x \geq 1+x$

$-e^x \leq -1-x$

$\leq \varepsilon |\Delta_t|_1$

No need to recompute at every stage

"Follow the Lazy Leader"

Round to grid points.

Recompute only if grid point changes.