Partial Monitoring

Kartik (CS20S020)

Paper 1: Regret Minimization Under Partial Monitoring

Full-information setting

In each round $t = 1,\ 2, \ldots, T$

1. Algorithm chooses action $I_t \in \mathcal{X}$
2. adversary choose outcome $y_t \in \mathcal{Y}$
3. Compute loss $\ell(I_t, y_t)$
4. Algorithm receives feedback y_t
   - Also, for each arm i where $\{i : i \neq I_t\}$, we get losses $\ell(i, y_t)$

- Regret

$$\mathbf{R}_T = \sum_{t=1}^{T} \ell(I_t, y_t) - \min_{i=1,\ 2,\ldots,N} \sum_{t=1}^{T} \ell(i, y_t)$$

- Hannan consistency

$$\frac{\mathbf{R}_T}{T} \to 0 \text{ as } T \to \infty$$

Bandit setting

In each round $t = 1,\ 2, \ldots, T$

1. Algorithm chooses action $I_t \in \mathcal{X}$
2. adversary choose outcome $y_t \in \mathcal{Y}$
3. Compute loss $\ell(I_t, y_t)$
4. Algorithm receives feedback y_t
   - ~~Also, for each arm i where $\{i : i \neq I_t\}$, we get losses $\ell(i, y_t)$~~

- Regret

$$\mathbf{R}_T = \sum_{t=1}^{T} \ell(I_t, y_t) - \min_{i=1,\ 2,\ldots,N} \sum_{t=1}^{T} \ell(i, y_t)$$

- Hannan consistency

$$\frac{\mathbf{R}_T}{T} \to 0 \text{ as } T \to \infty$$

Partial Information

- In some games, the Algorithm can't see the environment's action.
- AKA., Partial Information
- Examples
  - Dynamic Pricing
  - Apple Tasting
  - Online Stochastic Shortest path
  - Network Routing
- A generalized version of the bandit problem?

Partial Monitoring: Framework

- → **Parameters:** number of action N, number of outcomes M *loss matrix* $\mathbf{L} = [\ell(i,j)]_{N\times M}$ and *feedback matrix* $\mathbf{H} = [h(i,j)]_{N\times M}$.
- → h is a known *feedback function* that assigns to each action-outcome pair, an element of a finite set $\mathcal{S} = \{s_1, \ldots, s_m\}$ of signals.

In each round $t = 1,\ 2, \ldots, T$

1. Based on probability distribution $\mathbf{p}_t$ over the set of N actions, algorithm chooses draws an action $I_t \in \{1, \ldots, N\}$.
2. Adversary chooses outcome $y_t \in \{1, \ldots, M\}$ without revealing it
3. Algorithm incurs loss $\ell(I_t, y_t)$ and each action i incurs loss $\ell(i, y_t)$, where none of these values is revealed to the algorithm.
4. Algorithm receives feedback $h(I_t, y_t)$

Example I: Dynamic Pricing

We're selling a stream of identical products such as books, tickets etc.

Assumption: No bargaining

- Asking (or selling) price $I \in \mathcal{X} = [0, 1]$
- Customer's price $y \in \mathcal{Y} = [0, 1]$
- Feedback $h(I, y) = \mathbb{I}\{I \leq y\}$
- Loss

$$\ell(I, y) = \begin{cases} y - I & \text{if } y \geq I (\text{sell}), \\ c & \text{if } y < I (\text{no sell}) \end{cases}$$

for some $c > 0$

Example II: MAB

- action $I \in \mathcal{X} = \{1,\ 2, \dots, K\}$
- adversary's outcome $y \in \mathcal{Y} = [0, 1]^K$
- Loss = Feedback $h(I, y) = \ell(I, y) = y_I$
- We can say: $\mathbf{H} = \mathbf{L}$

Question: Under what conditions on the loss and feedback matrices is it possible to achieve Hannan consistency?

Upper Bounds on the regret (Case 1: $\mathbf{L} = \mathbf{KH}$)

Algorithm:

Parameters: matrix of losses $\mathbf{L}$, feedback matrix $\mathbf{H}$, matrix $\mathbf{K}$ such that $\mathbf{L} = \mathbf{K}\,\mathbf{H}$, real numbers $0 < \eta, \gamma < 1$.

Initialization: $\mathbf{w}_0 = (1, \ldots, 1)$.

For each round $t = 1, 2, \ldots$

(1) draw an action $I_t \in \{1, \ldots, N\}$ according to the distribution

$$p_{i,t} = (1-\gamma)\frac{w_{i,t-1}}{W_{t-1}} + \frac{\gamma}{N} \qquad i = 1, \ldots, N\,;$$

(2) get feedback $h_t = h(I_t, Y_t)$ and compute $\widetilde{\ell}_{i,t} = k(i, I_t)h_t / p_{I_t,t}$ for all $i = 1, \ldots, N$;

(3) compute $w_{i,t} = w_{i,t-1}e^{-\eta\tilde{\ell}(i,Y_t)}$ for all $i = 1, \ldots, N$.

Upper Bounds on the regret (Case 1: $\mathbf{L} = \mathbf{KH}$)

Regret:

$$\mathbf{R}_T \leq 3T^{\frac{2}{3}} (k^* N)^{\frac{2}{3}} (\log_e N)^{\frac{1}{3}}$$

→ Loss estimates are unbiased.

$$\mathbb{E}_t[\tilde{\ell}(i, y_t)] = \sum_{k=1}^{N} \frac{k(i,k)h(k,y_t)}{p_{k,t}} p_{k,t} = \sum_{k=1}^{T} k(i,k)h(k,y_t) = \ell(i, y_t)$$

where $i = 1, \ldots, N$.

→ Upper bound with high probability

- $\mathbf{R}_T \leq 13(k^* N)^{\frac{2}{3}} (\log_e N)^{\frac{1}{3}} (n+1)^{\frac{2}{3}} \sqrt{\log_e \frac{1}{\delta}}$

→ Rate of $T^{-1/3}$ is significantly slower than the best rate $T^{-1/2}$ obtained in the "full information" case.

→ The price paid for having access to only some feedback except for the actual outcomes is the deterioration in the rate of convergence.

Upper Bounds on the regret (Case 2: $\mathbf{L} \neq \mathbf{KH}$)

- Can we achieve similar Hannan consistent bounds when $\mathbf{L} \neq \mathbf{KH}$?
- An example:
  - Let $M = N = 3$

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } \mathbf{H} = \begin{bmatrix} a & b & c \\ d & d & d \\ e & e & e \end{bmatrix}$$

  - Rank of $\mathbf{H}$ is at most two.

- An action i is said to be revealing for a feedback matrix $\mathbf{H}$ if all entries in the i^{th} row of $\mathbf{H}$ are different.

Upper Bounds on the regret (Case 2: $\mathbf{L} \neq \mathbf{KH}$)

Algorithm:

Parameters: $0 \leq \varepsilon \leq 1$ and $\eta > 0$. Action r is revealing.

Initialization: $\mathbf{w}_0 = (1, \ldots, 1)$.

For each round $t = 1, 2, \ldots,$

(1) draw an action J_t from $\{1, \ldots, N\}$ according to the distribution $p_{i,t} = w_{i,t-1}/(w_{1,t-1} + \cdots + w_{N,t-1})$ for $i = 1, \ldots, N$;

(2) draw a Bernoulli random variable Z_t such that $\mathbb{P}[Z_t = 1] = \varepsilon$;

(3) if $Z_t = 1$, then play the revealing action, $I_t = r$, observe Y_t, and compute

$$w_{i,t} = w_{i,t-1} e^{-\eta\, \ell(i,Y_t)/\varepsilon} \qquad \text{for each } i = 1, \ldots, N;$$

(4) otherwise, play $I_t = J_t$ and let $w_{i,t} = w_{i,t-1}$ for each $i = 1, \ldots, N$.

Regret:

$$\frac{1}{T}\left(\sum_{t=1}^{T} \tilde{\ell}(I_t, y_t) - \min_{i=1,\,2,\ldots,N} \sum_{t=1}^{T} \ell(i, y_t)\right) \leq 8\, T^{\frac{-1}{3}} \left(\log_e \frac{4N}{\delta}\right)^{\frac{1}{3}}$$

Label Efficient prediction: Lower Bounds

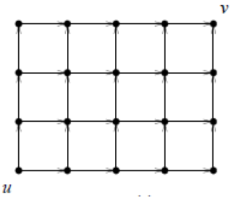
- Can rate of convergence be improved?
- Algorithm can query the outcome y_t only a limited number of times.
- Let $M = 2$ and $N = 3$.

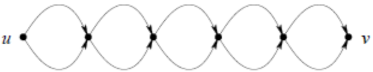
$$\mathbf{L} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } \mathbf{H} = \begin{bmatrix} a & b \\ c & c \\ c & c \end{bmatrix}$$

- For $a \neq b \neq c$, any label-efficient prediction will have an obtained rate of convergence $\mathcal{O}(T^{\frac{-1}{3}})$.
- For any randomized strategy, $\mathbf{R}_T \geq \frac{T^{\frac{2}{3}}}{7}$

Paper2: The On-Line Shortest Path Problem Under Partial Monitoring

SSP





- Consider a finite DAG with a set of edges $E = \{e_1, \dots, e_{|E|}\}$ and a set of vertices V.
- Each edge $e \in E$ is an ordered pair of vertices (v_1, v_2).
- A *path* from u to v is a sequence of edges $e^1, \dots, e^k$ such that $e^1 = (u, v_1), e^j = (v_{j-1}, v_j)$ for all $j = 2, \dots k-1$, and $e^k = (v_{k-1}, v)$.
- Let $\mathcal{P} = \{i_1, \dots, i_N\}$.
- **Assumption:** Every edge in E is on some path from u to v and every vertex in V is an endpoint of an edge.

Defining regret for SSPs

- We say $e \in i$ if edge $e \in E$ belongs to the path $i \in \mathcal{P}$.
- Path loss:

$$\ell(i, t) = \sum_{e \in i} \ell(e, t)$$

- Cumulative loss of algorithm:

$$\hat{L}_t = \sum_{s=1}^{t} \ell(I_s, s) = \sum_{s=1}^{t} \sum_{e \in I_s} \ell(e, s)$$

- Path I_t is chosen randomly according to some distribution $\mathbf{p}_t$ over all paths from u to v.
- **Regret:**

$$\mathbf{R}_T = \frac{1}{T}(\hat{L}_t - \min_{i \in \mathcal{P}} L_{i,t})$$

MAB setup for SSP

- At each time instance t, the algorithm chooses a path $I_t \in \mathcal{P}$ from u to v.
- The adversary assigns loss $\ell_{e,t} \in [0,1]$ to each edge $e \in E$, and the algorithm suffers loss $\ell_{I_t,t}$
- Algorithm only has access to the losses of the edges of chosen path.
- Auer et al. (2002) algorithm's based on exponential weighting:
  - $\mathbf{R}_T \leq 5.5K\sqrt{\frac{N \log_e \frac{N}{\delta}}{T}} + \frac{K \log_e N}{2T}$
  - where N is the total number of paths, and K is the length of the longest path.

Bandit algorithm for SSP

Parameters: real numbers $\beta > 0$, $0 < \eta, \gamma < 1$.
Initialization: Set $w_{e,0} = 1$ for each $e \in E$, $\overline{w}_{i,0} = 1$ for each $i \in \mathcal{P}$, and $\overline{W}_0 = N$. For each round $t = 1, 2, \ldots$

(a) Choose a path I_t at random according to the distribution p_t on $\mathcal{P}$, defined by

$$p_{i,t} = \begin{cases} (1-\gamma)\frac{\overline{w}_{i,t-1}}{\overline{W}_{t-1}} + \frac{\gamma}{|C|} & \text{if } i \in C \\ (1-\gamma)\frac{\overline{w}_{i,t-1}}{\overline{W}_{t-1}} & \text{if } i \notin C. \end{cases}$$

(b) Compute the probability of choosing each edge e as

$$q_{e,t} = \sum_{i:e\in i} p_{i,t} = (1-\gamma)\frac{\sum_{i:e\in i}\overline{w}_{i,t-1}}{\overline{W}_{t-1}} + \gamma\frac{|\{i \in C : e \in i\}|}{|C|}.$$

(c) Calculate the estimated gains

$$g'_{e,t} = \begin{cases} \frac{g_{e,t}+\beta}{q_{e,t}} & \text{if } e \in I_t \\ \frac{\beta}{q_{e,t}} & \text{otherwise.} \end{cases}$$

(d) Compute the updated weights

$$\begin{aligned} w_{e,t} &= w_{e,t-1}e^{\eta g'_{e,t}} \\ \overline{w}_{i,t} &= \prod_{e\in i} w_{e,t} = \overline{w}_{i,t-1}e^{\eta g'_{i,t}} \end{aligned}$$

where $g'_{i,t} = \sum_{e\in i} g'_{e,t}$, and the sum of the total weights of the paths

$$\overline{W}_t = \sum_{i\in\mathcal{P}} \overline{w}_{i,t}.$$

Bandit algorithm for SSP

- Gains are estimated for each edge and not for each path.
- Improves upper bound on the performance with the number of edges in place of the number of paths.
- To ensure we visit every edge often, a set C of *covering paths* where for each edge $e \in E$, there is a path $i \in C$ such that $e \in i$ and $|C| \leq |E|$.
- Regret is of the $\mathcal{O}(K\sqrt{|E| \log_e N/n})$ where where $|E|$ is the number of edges of the graph, K is the length of the paths, and N is the total number of paths.

Label-efficient Bandit

- Algorithm only has access to the losses of all the edges of the chosen path for *m* requests.
- Motivated by cognitive packet network.
- Smart packets explore the network but don't carry any useful data.
- Data packets carry information along their paths
- Goal is to send data packets from source to destination with minimum delay of the chosen path.
- Data packets are α times larger than smart packets ($\alpha >> 1$), and ϵ is the proportion of time instances when smart packets are used, then $\frac{\epsilon}{(\epsilon + \alpha(1-\epsilon))}$ is the proportion of bandwidth sacrificed for well-informed routing decision.

Label-efficient Bandit: Algorithm and Regret Bound

Modified step:

(c') Draw a Bernoulli random variable S_t with $\mathbb{P}(S_t = 1) = \varepsilon$, and compute the estimated gains

$$g'_{e,t} = \begin{cases} \frac{g_{e,t}+\beta}{\varepsilon q_{e,t}} S_t & \text{if } e \in I_t \\ \frac{\beta}{\varepsilon q_{e,t}} S_t & \text{if } e \notin I_t \end{cases} .$$

Regret:

$$\mathbf{R}_T \leq \frac{27K}{2}\sqrt{\frac{|E|\log_e \frac{2N}{\delta}}{T\epsilon}}$$

- For $\epsilon = \left(m - \sqrt{2m\log_e \frac{1}{\delta}}\right)/T$, then $\mathbf{R}_T \leq \mathcal{O}\left(K\sqrt{\frac{|E|\log_e \frac{N}{\delta}}{m}}\right)$

Bandit algorithm for tracking the shortest path

- Goal is to perform as well as the best combination of paths which is allowed to change the path m times during time instance $t = 1, \ldots, T$.
- $L(\textsc{Part}(T, m, \mathbf{t}, \underline{\mathbf{i}})) = \sum_{j=0}^{m} \sum_{t=t_j}^{t_{j+1}-1} \sum_{e \in i_j} \ell_{e,t}$
  - "m-partition" prediction scheme: $\textsc{Part}(T, m, \mathbf{t}, \underline{\mathbf{i}})$ where the sequence of paths is partitioned into $m+1$ contiguous segments, and on each segment the scheme assigns exactly one of the N paths.
- Regret:

$$\mathbf{R}_T = \frac{1}{T}\left(\hat{L}_T - \min_{\mathbf{t},\underline{\mathbf{i}}} L(\textsc{Part}(T, m, \mathbf{t}, \underline{\mathbf{i}}))\right)$$

$$\implies \mathbf{R}_T \le \mathcal{O}\left(K\sqrt{\frac{m}{T}|C|\log_e N}\right)$$

Restricted MAB

- Algorithm receives access only to $\ell_{I_t,t}$.
- The algorithm alternates between choosing a path from a "basis" B to obtain unbiased estimates of the loss, and choosing a path according to exponential weighting based on these estimates.
- Path $i \in \mathcal{P}$ is a binary row vector with $|E|$ components.
- Set of edges spans the set of paths.
  - But they aren't observable!
- Alternatively, choose a subset of $\mathcal{P}$ that forms a *basis*
- Denote by B the $b \times |E|$ matrix whose rows $b^1, \ldots, b^b$ represent basis paths and b is equal to maximum number of LI vector in $\{i : i \in \mathcal{P}\}$.

Restricted MAB

- Let ℓ_t^E denote the column vector of edge losses $\{\ell_{e,t}\}_{e \in E}$ at time t.
- Let $\ell_t^B = (\ell_{b^1,t}, \ldots, \ell_{b^b,t})^T$ be a $b-$dimensional column vector whose components are the losses of the paths in the basis B at time t.
- Expressing path $i \in \mathcal{P} : i = \sum_{k=1}^{b} \alpha_{b^k}^{(i,B)} b^k$.
- $\ell_{i,t} = \langle i, \ell_t^E \rangle = \sum_{k=1}^{b} \alpha_{b^k}^{(i,B)} \ell_{b^k t}$
- We query the loss of a (random) basis vector from time to time, and create unbiased estimates $\hat{\ell}_{b^k,t}$ of the basis paths losses $\ell_{b^k,t}$, which are then transformed into edge-loss estimates.
- Regret:

$$\mathbf{R}_T \leq 9.1 K^2 b \left[Kb \log_e \left(\frac{4bN}{\delta} \right) \right]^{\frac{1}{3}} T^{\frac{-1}{3}}$$

Paper3: Regret Bounds and Minimax Policies under Partial Monitoring

Previous results

- **Aside:** Number of rounds T is denoted as n. Number of arms: K.
- Known regret bounds

	$\inf\sup \overline{R}_n$		$\inf\sup \mathbb{E}R_n$	
	Lower bound	Upper bound	Lower bound	Upper bound
Full information game	$\sqrt{n\log K}$	$\sqrt{n\log K}$	$\sqrt{n\log K}$	$\sqrt{n\log K}$
Label efficient game	$n\sqrt{\frac{\log K}{m}}$	$n\sqrt{\frac{\log K}{m}}$	$n\sqrt{\frac{\log K}{m}}$	$n\sqrt{\frac{\log n}{m}}$
Bandit game	$\sqrt{nK}$	$\sqrt{nK\log K}$	$\sqrt{nK}$	$\sqrt{nK\log n}$
Bandit label efficient game	$n\sqrt{\frac{K}{m}}$	$n\sqrt{\frac{K\log K}{m}}$	$n\sqrt{\frac{K}{m}}$	$n\sqrt{\frac{K\log n}{m}}$

- $\sqrt{\log K}$ gap for the mini-max pseudo-regret in the bandit game as well as the label efficient bandit game.
- $\sqrt{\log n}$ gap for the mini-max expected regret in the bandit game as well as the label efficient bandit game.
- $\sqrt{\frac{\log n}{\log K}}$ gap for the mini-max expected regret in the bandit game as well as the label efficient bandit game.

Implicitly Normalized Forecaster (INF)

- INF: a unified regret analysis in the four games.

INF (Implicitly Normalized Forecaster):

Parameters:

- the continuously differentiable function $\psi : \mathbb{R}_-^* \to \mathbb{R}_+^*$ satisfying (1)
- the estimates $v_{i,t}$ of $g_{i,t}$ based on the (drawn arms and) observed rewards at time t (and before time t)

Let p_1 be the uniform distribution over $\{1,\ldots,K\}$.

For each round $t = 1,2,\ldots,$

(1) Draw an arm I_t from the probability distribution p_t.

(2) Use the observed reward(s) to build the estimate $v_t = (v_{1,t},\ldots,v_{K,t})$ of $(g_{1,t},\ldots,g_{K,t})$ and let: $V_t = \sum_{s=1}^{t} v_s = (V_{1,t},\ldots,V_{K,t})$.

(3) Compute the normalization constant $C_t = C(V_t)$.

(4) Compute the new probability distribution $p_{t+1} = (p_{1,t+1},\ldots,p_{K,t+1})$ where

$$p_{i,t+1} = \psi(V_{i,t} - C_t).$$

Exponential Weighted Average (EWA) and Poly INF

- For $\Psi(x) = exp(\eta x) + \frac{\gamma}{K}$ and $exp(-\eta C(x)) = \frac{1-\gamma}{\sum_{i=1}^{K} exp(\eta x_i)}$, INF reduces to exponential weighted forecaster:

$$\mathbf{p}_{i,t+1} = (1-\gamma)\frac{exp(\eta V_{i,t})}{\sum_{j=1}^{K} exp(\eta V_{i,t})} + \frac{\gamma}{K}$$

- For $\Psi(x) = \left(\frac{\eta}{-x}\right)^q + \frac{\gamma}{K}$, where $q > 1$, INF reduces to exponential weighted forecaster.
- Poly INF forecaster generates nicer probability updates than the exponentially weighted average forecasters as, for bandits games (label efficient or not), it allows to remove the extraneous log K factor in the pseudo-regret bounds and some regret bounds.

Label efficient Games

- Number of queried reward is constrained either strictly or in expectation.
- Constraint on the expected number of queried reward vectors
  - EWA and Poly INF has bounds: $\bar{R}_n \leq n\sqrt{\frac{\log K}{2m}}$
- Hard constraint on the expected number of queried reward vectors
  - Use high probability bounds as an intermediate step, we can get the $\mathbb{E}R_n$ for non-oblivious opponents, as such an approach gives stronger bounds than pseudo-regret.

Bandit Game

- Main result: By using the Poly INF, we obtain an upper bound of $\mathcal{O}(\sqrt{nK})$ for $\bar{R}_n$
  $\implies \mathcal{O}(\sqrt{nK})$ for $\mathbb{E}\bar{R}_n$ for oblivious adversaries.
- In the general case (non-oblivious opponent), an upper bound of $\mathcal{O}(\sqrt{nK \log K})$ on $\mathbb{E}R_n$.
- Conjecture that this bound cannot be improved because opponent may take advantage of the past to make the algorithm pay a regret with the extra logarithmic factor.
- High probability bound holds for any confidence level.

Label-efficient Bandits and new regret bounds

- Expected number of queried rewards should be less or equal to m
- At each round, we draw a Bernoulli random variable w.p. $\frac{m}{n}$, to decide whether the gain of the chosen arm is revealed or not.

	$\inf\sup \overline{R}_n$	$\inf\sup \mathbb{E}R_n$	High probability bound on R_n
Label efficient game		$n\sqrt{\frac{\log K}{m}}$	$n\sqrt{\frac{\log(K\delta^{-1})}{m}}$
Bandit game with fully oblivious adversary	$\sqrt{nK}$	$\sqrt{nK}$	$\sqrt{nK}\log(\delta^{-1})$
Bandit game with oblivious adversary	$\sqrt{nK}$	$\sqrt{nK}$	$\sqrt{\frac{nK}{\log K}}\log(K\delta^{-1})$
Bandit game with general adversary	$\sqrt{nK}$	$\sqrt{nK\log K}$	$\sqrt{\frac{nK}{\log K}}\log(K\delta^{-1})$
L.E. bandit with deterministic adversary	$n\sqrt{\frac{K}{m}}$	$n\sqrt{\frac{K}{m}}$	$n\sqrt{\frac{K}{m}}\log(\delta^{-1})$
L.E. bandit with oblivious adversary	$n\sqrt{\frac{K}{m}}$	$n\sqrt{\frac{K}{m}}$	$n\sqrt{\frac{K}{m\log K}}\log(K\delta^{-1})$
L.E. bandit with general adversary	$n\sqrt{\frac{K}{m}}$	$n\sqrt{\frac{K\log K}{m}}$	$n\sqrt{\frac{K}{m\log K}}\log(K\delta^{-1})$