Nonstochastic Bandits with Unrestricted Delays

Nicolò Cesa-Bianchi[1] Yevgeny Seldin[2] Tobias Thune[2]

[1]Università degli Studi di Milano

[2]University of Copenhagen

The bandit problem

- K actions
- Unknown deterministic assignment of losses to actions

$$\ell_t = \left(\ell_t(1), \ldots, \ell_t(K)\right) \in [0,1]^K \quad \text{for t=1,2,...}$$

The bandit problem

- K actions
- Unknown deterministic assignment of losses to actions

$$\ell_t = \big(\ell_t(1), \ldots, \ell_t(K)\big) \in [0,1]^K \quad \text{for t=1,2,...}$$

For $t = 1, 2, \ldots$

1. Player selects an action I_t (possibly using randomization) and incurs loss $\ell_t(I_t)$

The bandit problem

- K actions
- Unknown deterministic assignment of losses to actions

$$\ell_t = \big(\ell_t(1), \ldots, \ell_t(K)\big) \in [0,1]^K \quad \text{for t=1,2,...}$$

For $t = 1, 2, \ldots$

1. Player selects an action I_t (possibly using randomization) and incurs loss $\ell_t(I_t)$
2. Player observes $\ell_t(I_t)$ and updates selection strategy

Regret

The average loss of the player should converge to that of the best action

$$R_T \overset{\text{def}}{=} \underbrace{\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(I_t)\right]}_{\text{player's loss}} - \underbrace{\min_{i=1,\dots,K} \sum_{t=1}^{T} \ell_t(i)}_{\text{best action's loss}} \overset{\text{want}}{=} o(T)$$

Regret

The average loss of the player should converge to that of the best action

$$R_T \overset{\text{def}}{=} \underbrace{\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(I_t)\right]}_{\text{player's loss}} - \underbrace{\min_{i=1,\dots,K} \sum_{t=1}^{T} \ell_t(i)}_{\text{best action's loss}} \overset{\text{want}}{=} o(T)$$

- Expectation is with respect to the player's internal randomization
- Random play typically results in linear regret

The Exp3 algorithm

At time t draw action I_t with probability $p_t(i) \stackrel{\text{def}}{=} \mathbb{P}\big(I_t = i \,\big|\, I_1, \ldots, I_{t-1}\big)$

The Exp3 algorithm

At time t draw action I_t with probability $p_t(i) \stackrel{\text{def}}{=} \mathbb{P}\left(I_t = i \,\middle|\, I_1, \ldots, I_{t-1}\right)$

- $p_t(i) \quad \propto \quad \exp\left(-\eta \sum_{s=1}^{t-1} \widehat{\ell}_s(i)\right) \qquad i = 1, \ldots, K$

The Exp3 algorithm

At time t draw action I_t with probability $p_t(i) \stackrel{\text{def}}{=} \mathbb{P}\big(I_t = i \,\big|\, I_1, \ldots, I_{t-1}\big)$

- $p_t(i) \quad \propto \quad \exp\left(-\eta \sum_{s=1}^{t-1} \widehat{\ell}_s(i)\right) \qquad i = 1, \ldots, K$
- $\widehat{\ell}_t(i) = \begin{cases} \dfrac{\ell_t(i)}{p_t(i)} & \text{if } I_t = i \\ 0 & \text{otherwise} \end{cases}$

only one non-zero component

The Exp3 algorithm

At time t draw action I_t with probability $p_t(i) \stackrel{\text{def}}{=} \mathbb{P}\left(I_t = i \,\middle|\, I_1, \ldots, I_{t-1}\right)$

- $p_t(i) \quad \propto \quad \exp\left(-\eta \sum_{s=1}^{t-1} \widehat{\ell}_s(i)\right) \qquad i = 1, \ldots, K$
- $\widehat{\ell}_t(i) = \begin{cases} \dfrac{\ell_t(i)}{p_t(i)} & \text{if } I_t = i \\ 0 & \text{otherwise} \end{cases}$

only one non-zero component

Exp3 regret bound

[Auer et al., 2002]

$$R_T \leqslant \sqrt{(\ln K) \sum_t \left\|\ell_t\right\|^2} = \mathcal{O}\left(\sqrt{TK \ln K}\right)$$

The Exp3 algorithm

At time t draw action I_t with probability $p_t(i) \stackrel{\text{def}}{=} \mathbb{P}\left(I_t = i \,\middle|\, I_1, \ldots, I_{t-1}\right)$

- $p_t(i) \quad \propto \quad \exp\left(-\eta \sum_{s=1}^{t-1} \widehat{\ell}_s(i)\right) \qquad i = 1, \ldots, K$
- $\widehat{\ell}_t(i) = \begin{cases} \dfrac{\ell_t(i)}{p_t(i)} & \text{if } I_t = i \\ 0 & \text{otherwise} \end{cases}$

only one non-zero component

Exp3 regret bound

[Auer et al., 2002]

$$R_T \leqslant \sqrt{(\ln K) \sum_t \left\|\ell_t\right\|^2} = \mathcal{O}\left(\sqrt{TK \ln K}\right)$$

Tight up to log factors

Learning with delayed losses

- In many bandit settings, like product recommendation, we observe the effects of our actions after some delay

Learning with delayed losses

- In many bandit settings, like product recommendation, we observe the effects of our actions after some delay
- Assume the loss of action I_t is observed at time $t + d_t$, where $d_1, d_2, \ldots$ is an unknown and deterministic sequence of delays

Learning with delayed losses

- In many bandit settings, like product recommendation, we observe the effects of our actions after some delay
- Assume the loss of action I_t is observed at time $t + d_t$, where $d_1, d_2, \ldots$ is an unknown and deterministic sequence of delays

For $t = 1, 2, \ldots$

1. Play I_t

Learning with delayed losses

- In many bandit settings, like product recommendation, we observe the effects of our actions after some delay
- Assume the loss of action I_t is observed at time $t + d_t$, where $d_1, d_2, \ldots$ is an unknown and deterministic sequence of delays

For $t = 1, 2, \ldots$

1. Play I_t
2. Incur loss $\ell_t(I_t)$

Learning with delayed losses

- In many bandit settings, like product recommendation, we observe the effects of our actions after some delay
- Assume the loss of action I_t is observed at time $t + d_t$, where $d_1, d_2, \ldots$ is an unknown and deterministic sequence of delays

For $t = 1, 2, \ldots$

1. Play I_t
2. Incur loss $\ell_t(I_t)$
3. Observe pairs $\big(s, \ell_s(I_s)\big)$ for all $s \leqslant t$ such $s + d_s = t$

Learning with delayed losses

- In many bandit settings, like product recommendation, we observe the effects of our actions after some delay
- Assume the loss of action I_t is observed at time $t + d_t$, where $d_1, d_2, \ldots$ is an unknown and deterministic sequence of delays

For $t = 1, 2, \ldots$

1. Play I_t
2. Incur loss $\ell_t(I_t)$
3. Observe pairs $\big(s, \ell_s(I_s)\big)$ for all $s \leqslant t$ such $s + d_s = t$

Note: The learner knows that $\ell_s(I_s)$ refers to action I_s played at time s

Variable delayed feedback

Consider variable delays $d_1, d_2, \ldots$

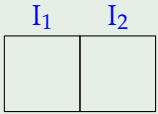
Example

I_1

Variable delayed feedback

Consider variable delays $d_1, d_2, \ldots$

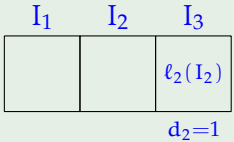
Example



Variable delayed feedback

Consider variable delays $d_1, d_2, \ldots$

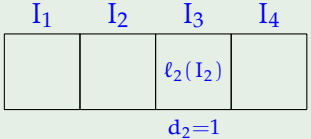
Example



Variable delayed feedback

Consider variable delays $d_1, d_2, \ldots$

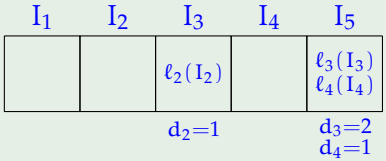
Example



Variable delayed feedback

Consider variable delays $d_1, d_2, \ldots$

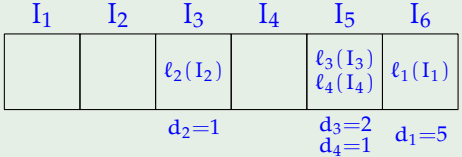
Example



Variable delayed feedback

Consider variable delays $d_1, d_2, \ldots$

Example



Note that $\sum_t d_t$ can be of order T^2 (all feedbacks received at the end)

Learning with delayed losses

How does the regret depend on the total delay $D = \sum_t d_t$?

Learning with delayed losses

How does the regret depend on the total delay $D = \sum_t d_t$?

Lower bound

$$\max\Big\{ \underbrace{\sqrt{KT}}_{\substack{\text{bandit}\\ \text{lower bound}}} \;,\; \underbrace{\sqrt{(T+D)\ln K}}_{\substack{\text{delayed experts}\\ \text{lower bound}}} \Big\} = \Omega\left(\sqrt{KT + D\ln K}\right)$$

Even with fixed delays $d_t = d$ so that $D = dT$

Learning with delayed losses

How does the regret depend on the total delay $D = \sum_t d_t$?

Lower bound

$$\max\Big\{ \underbrace{\sqrt{KT}}_{\substack{\text{bandit}\\ \text{lower bound}}} \;,\; \underbrace{\sqrt{(T+D)\ln K}}_{\substack{\text{delayed experts}\\ \text{lower bound}}} \Big\} = \Omega\left(\sqrt{KT + D\ln K}\right)$$

Even with fixed delays $d_t = d$ so that $D = dT$

Upper bound for fixed delays: $R_T = \mathcal{O}\left(\sqrt{(K+d)T\ln K}\right)$ [C-B et al., 2016]

Related works

- Online learning with delays
  [Weinberger and Ordentlich, 2002], [Mesterharm, 2005], [Joulani, György, and Szepesvári, 2016], [Ghosh and Ramchandran, 2018]

Related works

- Online learning with delays
  [Weinberger and Ordentlich, 2002], [Mesterharm, 2005], [Joulani, György, and Szepesvári, 2016], [Ghosh and Ramchandran, 2018]
- Bandits with delayed feedback
  [Neu, Antos, György, and Szepesvári, 2010], [Julani, György, and Szepesvári, 2013], [C-B, Gentile, and Mansour, 2016], [Vernade, Cappé, and Perchet, 2017], [Li, Chen, and Giannakis, 2019]

Related works

- Online learning with delays
  [Weinberger and Ordentlich, 2002], [Mesterharm, 2005], [Joulani, György, and Szepesvári, 2016], [Ghosh and Ramchandran, 2018]
- Bandits with delayed feedback
  [Neu, Antos, György, and Szepesvári, 2010], [Julani, György, and Szepesvári, 2013], [C-B, Gentile, and Mansour, 2016], [Vernade, Cappé, and Perchet, 2017], [Li, Chen, and Giannakis, 2019]
- Contextual bandits with delayed feedback
  [Arya and Yang, 2019]

Related works

- Online learning with delays
  [Weinberger and Ordentlich, 2002], [Mesterharm, 2005], [Joulani, György, and Szepesvári, 2016], [Ghosh and Ramchandran, 2018]
- Bandits with delayed feedback
  [Neu, Antos, György, and Szepesvári, 2010], [Julani, György, and Szepesvári, 2013], [C-B, Gentile, and Mansour, 2016], [Vernade, Cappé, and Perchet, 2017], [Li, Chen, and Giannakis, 2019]
- Contextual bandits with delayed feedback
  [Arya and Yang, 2019]
- Bandits with delayed anonymous feedback
  [Pike-Burke, Agrawal, Szepesvári, and Grunewalder, 2017], [C-B, Gentile, and Mansour, 2018]

A simple solution

Delayed Exp3

At time t, make importance-weighted updates using the old probabilities

$$\widehat{\ell}_t(i) = \begin{cases} \dfrac{\ell_s(i)}{p_s(i)} & \text{if } s + d_s = t \text{ and } I_s = i \\ 0 & \text{otherwise} \end{cases}$$

A simple solution

Delayed Exp3

At time t, make importance-weighted updates using the old probabilities

$$\widehat{\ell}_t(i) = \begin{cases} \dfrac{\ell_s(i)}{p_s(i)} & \text{if } s + d_s = t \text{ and } I_s = i \\ 0 & \text{otherwise} \end{cases}$$

- Regret bound: $R_T = \mathcal{O}\left(\sqrt{(KT + D)\ln K}\right)$

A simple solution

Delayed Exp3

At time t, make importance-weighted updates using the old probabilities

$$\widehat{\ell}_t(i) = \begin{cases} \dfrac{\ell_s(i)}{p_s(i)} & \text{if } s + d_s = t \text{ and } I_s = i \\ 0 & \text{otherwise} \end{cases}$$

- Regret bound: $R_T = \mathcal{O}\left(\sqrt{(KT + D)\ln K}\right)$
- Optimal to within $\ln K$ factors ☺

A simple solution

Delayed Exp3

At time t, make importance-weighted updates using the old probabilities

$$\widehat{\ell}_t(i) = \begin{cases} \dfrac{\ell_s(i)}{p_s(i)} & \text{if } s + d_s = t \text{ and } I_s = i \\ 0 & \text{otherwise} \end{cases}$$

- Regret bound: $R_T = \mathcal{O}\left(\sqrt{(KT + D)\ln K}\right)$
- Optimal to within $\ln K$ factors ☺
- Holds only for certain sequences $d_1, d_2, \ldots$ of delays ☹

A simple solution

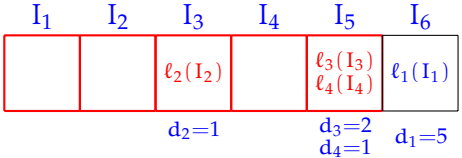
Delayed Exp3

At time t, make importance-weighted updates using the old probabilities

$$\widehat{\ell}_t(i) = \begin{cases} \dfrac{\ell_s(i)}{p_s(i)} & \text{if } s + d_s = t \text{ and } I_s = i \\ 0 & \text{otherwise} \end{cases}$$

- Regret bound: $R_T = \mathcal{O}\left(\sqrt{(KT + D)\ln K}\right)$
- Optimal to within $\ln K$ factors ☺
- Holds only for certain sequences $d_1, d_2, \ldots$ of delays ☹
- Requires tuning based on unfathomable quantities ☹

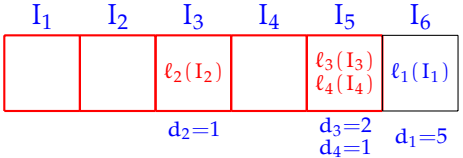
Why does it work?



3 feedbacks received in the interval $[1, 1+5)$

- The stability span N of a delay sequence $d_1, d_2, \ldots$ is the largest amount of feedback received in any interval $[t, t+d_t)$

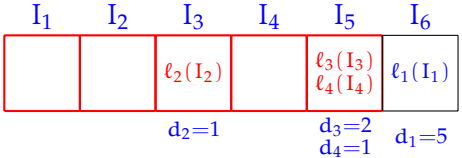
Why does it work?



3 feedbacks received in the interval $[1, 1+5)$

- The stability span N of a delay sequence $d_1, d_2, \ldots$ is the largest amount of feedback received in any interval $[t, t+d_t)$
- If $\eta \leqslant \dfrac{1}{2eN}$ then $p_{t+d_t}(i) \leqslant e\, p_t(i)$ stability

Why does it work?

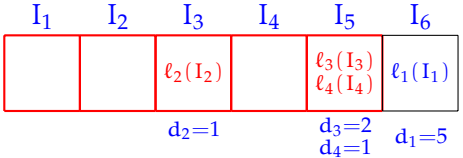


3 feedbacks received in the interval $[1, 1+5)$

- The stability span N of a delay sequence $d_1, d_2, \dots$ is the largest amount of feedback received in any interval $[t, t+d_t)$
- If $\eta \leqslant \dfrac{1}{2eN}$ then $p_{t+d_t}(i) \leqslant e\, p_t(i)$ stability

Regret bound when N is known $R_T \leqslant \max\left\{\dfrac{\ln K}{\eta}, N \ln K\right\} + \eta(KT + D)$

Why does it work?



3 feedbacks received in the interval $[1, 1+5)$

- The stability span N of a delay sequence $d_1, d_2, \dots$ is the largest amount of feedback received in any interval $[t, t+d_t)$
- If $\eta \leqslant \dfrac{1}{2eN}$ then $p_{t+d_t}(i) \leqslant e\, p_t(i)$ stability

Regret bound when N is known $R_T \leqslant \max\left\{\dfrac{\ln K}{\eta}, N \ln K\right\} + \eta(KT + D)$

If $N = \mathcal{O}\left(\sqrt{\dfrac{KT+D}{\ln K}}\right)$ then tuning based on N and D gives $R_T = \mathcal{O}\left(\sqrt{(KT+D)\ln K}\right)$

Dealing with large delays

- Large delays can be ignored by the algorithm paying only constant regret each time

Dealing with large delays

- Large delays can be ignored by the algorithm paying only constant regret each time
- There exists a tradeoff between:
  1. number of skipped updates
  2. sum of non-skipped delays d_t

Dealing with large delays

- Large delays can be ignored by the algorithm paying only constant regret each time
- There exists a tradeoff between:
  1. number of skipped updates
  2. sum of non-skipped delays d_t

The Skipper algorithm

- Skipper wraps Delayed Exp3 ignoring feedback $\ell_t(I_t)$ when $d_t > \delta$ (a parameter)

Dealing with large delays

- Large delays can be ignored by the algorithm paying only constant regret each time
- There exists a tradeoff between:
  1. number of skipped updates
  2. sum of non-skipped delays d_t

The Skipper algorithm

- Skipper wraps Delayed Exp3 ignoring feedback $\ell_t(I_t)$ when $d_t > \delta$ (a parameter)
- Since $N \leqslant 2 \max_t d_t$, this controls the stability span perceived by Delayed Exp3

Dealing with large delays

- Large delays can be ignored by the algorithm paying only constant regret each time
- There exists a tradeoff between:
  1. number of skipped updates
  2. sum of non-skipped delays d_t

The Skipper algorithm

- Skipper wraps Delayed Exp3 ignoring feedback $\ell_t(I_t)$ when $d_t > \delta$ (a parameter)
- Since $N \leqslant 2 \max_t d_t$, this controls the stability span perceived by Delayed Exp3
- As a consequence, we can do away with the knowledge of N

Regret bound for Skipper

Regret bound for Skipper with doubling trick

$$R_T \leqslant S_\delta + \max\left\{\frac{\ln K}{\eta} + \delta \ln K + \eta\left(D_\delta + KT\right)\right\}$$

- S_δ is number of skipped delays at threshold δ
- D_δ is sum of non-skipped delays

Regret bound for Skipper

Regret bound for Skipper with doubling trick

$$R_T \leqslant S_\delta + \max\left\{\frac{\ln K}{\eta} + \delta \ln K + \eta\left(D_\delta + KT\right)\right\}$$

- S_δ is number of skipped delays at threshold δ
- D_δ is sum of non-skipped delays

By tuning δ as a function of $D = \sum_t d_t$ and T only, we get $R_T = \mathcal{O}\left(\sqrt{(KT + D)\ln K}\right)$ without any assumption on $d_1, d_2, \ldots$

No prior knowledge?

Skipper still needs knowledge of $\sum_t d_t$ to tune the skipping threshold δ

No prior knowledge?

Skipper still needs knowledge of $\sum_t d_t$ to tune the skipping threshold δ

Doubling trick:

1. Run Skipper with δ tuned using an optimistic guess $D' \geqslant \sum_t d_t$

No prior knowledge?

Skipper still needs knowledge of $\sum_t d_t$ to tune the skipping threshold δ

Doubling trick:

1. Run Skipper with δ tuned using an optimistic guess $D' \geqslant \sum_t d_t$
2. Restart Skipper with a doubled guess as soon as $d_1 + \cdots + d_t > D'$

No prior knowledge?

Skipper still needs knowledge of $\sum_t d_t$ to tune the skipping threshold δ

Doubling trick:

1. Run Skipper with δ tuned using an optimistic guess $D' \geqslant \sum_t d_t$
2. Restart Skipper with a doubled guess as soon as $d_1 + \cdots + d_t > D'$

Problem: event $d_1 + \cdots + d_t > D'$ is only known at time $t + d_t$ ☹

Delay known at action time

Assumption: The delay d_t is observed before choosing I_t

Delay known at action time

Assumption: The delay d_t is observed before choosing I_t

Doubling trick for Skipper

For each epoch $m = 1, 2, \ldots$

1. Run Skipper with skipping threshold $\delta \sim 2^m$

Delay known at action time

Assumption: The delay d_t is observed before choosing I_t

Doubling trick for Skipper

For each epoch $m = 1, 2, \ldots$

1. Run Skipper with skipping threshold $\delta \sim 2^m$
2. Restart Skipper based on a (complicated) condition involving:
   - current number of skipped rounds
   - current length of epoch
   - current sum of experienced delays

The oracle bound

Regret bound for Skipper with doubling trick

$$R_T \leqslant \min_{\delta} \left\{ S_\delta + \delta \ln K + \frac{KT + D_\delta}{\delta} \right\} + K \ln K$$

- S_δ is number of skipped delays at threshold δ
- D_δ is sum of non-skipped delays

The oracle bound

Regret bound for Skipper with doubling trick

$$R_T \leqslant \min_{\delta} \left\{ S_\delta + \delta \ln K + \frac{KT + D_\delta}{\delta} \right\} + K \ln K$$

- S_δ is number of skipped delays at threshold δ
- D_δ is sum of non-skipped delays

This bound is at most $\sqrt{(KT + D) \ln K} + K \ln K$ ☺

The oracle bound is better than the tuned Skipper bound

Comparing $\underbrace{\min_{\delta}\left\{S_\delta + \delta \ln K + \frac{KT + D_\delta}{\delta}\right\}}_{\text{Oracle}}$ with $\underbrace{\sqrt{(KT + D)\ln K}}_{\text{Magic tuning}}$

The oracle bound is better than the tuned Skipper bound

Comparing $\underbrace{\min_{\delta}\left\{S_\delta + \delta \ln K + \frac{KT + D_\delta}{\delta}\right\}}_{\text{Oracle}}$ with $\underbrace{\sqrt{(KT + D)\ln K}}_{\text{Magic tuning}}$

- For $\delta = o(T)$ choose delay sequence $d_t = \begin{cases} T - t & \text{if } t < \delta \\ 0 & \text{otherwise} \end{cases}$

The oracle bound is better than the tuned Skipper bound

Comparing $\underbrace{\min_{\delta}\left\{S_\delta + \delta \ln K + \frac{KT + D_\delta}{\delta}\right\}}_{\text{Oracle}}$ with $\underbrace{\sqrt{(KT + D)\ln K}}_{\text{Magic tuning}}$

- For $\delta = o(T)$ choose delay sequence $d_t = \begin{cases} T - t & \text{if } t < \delta \\ 0 & \text{otherwise} \end{cases}$
- Then $\sum_t d_t = \Theta(T\delta)$

The oracle bound is better than the tuned Skipper bound

Comparing $\underbrace{\min_{\delta}\left\{S_\delta + \delta \ln K + \frac{KT + D_\delta}{\delta}\right\}}_{\text{Oracle}}$ with $\underbrace{\sqrt{(KT + D)\ln K}}_{\text{Magic tuning}}$

- For $\delta = o(T)$ choose delay sequence $d_t = \begin{cases} T - t & \text{if } t < \delta \\ 0 & \text{otherwise} \end{cases}$
- Then $\sum_t d_t = \Theta(T\delta)$
- ... but $D_\delta = 0$ and $S_\delta < \delta$

The oracle bound is better than the tuned Skipper bound

Comparing $\underbrace{\min_{\delta}\left\{S_\delta + \delta \ln K + \frac{KT + D_\delta}{\delta}\right\}}_{\text{Oracle}}$ with $\underbrace{\sqrt{(KT + D)\ln K}}_{\text{Magic tuning}}$

- For $\delta = o(T)$ choose delay sequence $d_t = \begin{cases} T - t & \text{if } t < \delta \\ 0 & \text{otherwise} \end{cases}$
- Then $\sum_t d_t = \Theta(T\delta)$
- ... but $D_\delta = 0$ and $S_\delta < \delta$

Taking $\delta = \sqrt{KT/(\ln K)}$ the corresponding regret bounds are of order

- Oracle: $\sqrt{T}$
- Magic tuning: $T^{3/4}$

Experiments

Some modeling choices

- Bounded vs. unbounded delay

Some modeling choices

- Bounded vs. unbounded delay
- Time-stamped feedback: $\left(I_s, \ell_s(I_s)\right)$ instead of $\left(s, \ell_s(I_s)\right)$

Some modeling choices

- Bounded vs. unbounded delay
- Time-stamped feedback: $\big(I_s, \ell_s(I_s)\big)$ instead of $\big(s, \ell_s(I_s)\big)$
- Anonymous feedback

I_1	I_2	I_3	I_4	I_5
		$\ell_2(I_2)$		$\ell_1(I_1)$ + $\ell_3(I_3)$

Some modeling choices

- Bounded vs. unbounded delay
- Time-stamped feedback: $\big(I_s, \ell_s(I_s)\big)$ instead of $\big(s, \ell_s(I_s)\big)$
- Anonymous feedback
- Delay known at action vs. observation time

I_1	I_2	I_3	I_4	I_5
		$\ell_2(I_2)$		$\ell_1(I_1)$ + $\ell_3(I_3)$

Summary

Setting	Regret Bound	Reference

Summary

Setting	Regret Bound	Reference
Fixed delay	$\sqrt{(K+d)T \ln K}$	[C-B et al., 2016]

Summary

Setting	Regret Bound	Reference
Fixed delay	$\sqrt{(K+d)T \ln K}$	[C-B et al., 2016]
Action time	$\min_{\delta} \left(S_\delta + \delta \ln K + \frac{KT + D_\delta}{\delta} \right)$	This work

Summary

Setting	Regret Bound	Reference
Fixed delay	$\sqrt{(K+d)T \ln K}$	[C-B et al., 2016]
Action time	$\min_{\delta} \left(S_\delta + \delta \ln K + \frac{KT + D_\delta}{\delta} \right)$	This work
Observation time (known T, D)	$\sqrt{(KT+D) \ln K}$	This work

Summary

Setting	Regret Bound	Reference
Fixed delay	$\sqrt{(K+d)T \ln K}$	[C-B et al., 2016]
Action time	$\min_{\delta} \left(S_\delta + \delta \ln K + \frac{KT + D_\delta}{\delta} \right)$	This work
Observation time (known T, D)	$\sqrt{(KT + D) \ln K}$	This work
Anonymous (known d_{max})	$\sqrt{d_{max} KT \ln K}$	[C-B et al., 2018]