Reducing contextual bandits to supervised learning

Daniel Hsu
Columbia University

Based on joint work with A. Agarwal, S. Kale,
J. Langford, L. Li, and R. Schapire

Learning to interact: example #1

Practicing physician

Learning to interact: example #1

Practicing physician

Loop:

1. Patient arrives with symptoms, medical history, genome . . .

Learning to interact: example #1

Practicing physician

Loop:

1. Patient arrives with symptoms, medical history, genome . . .
2. Prescribe treatment.

Learning to interact: example #1

Practicing physician

Loop:

1. Patient arrives with symptoms, medical history, genome . . .
2. Prescribe treatment.
3. Observe impact on patient's health (*e.g.*, improves, worsens).

Learning to interact: example #1

Practicing physician
Loop:

1. Patient arrives with symptoms, medical history, genome . . .
2. Prescribe treatment.
3. Observe impact on patient's health (*e.g.*, improves, worsens).

Goal: prescribe treatments that yield good health outcomes.

Learning to interact: example #2

Website operator

Learning to interact: example #2

Website operator

Loop:

1. User visits website with profile, browsing history . . .

Learning to interact: example #2

Website operator

Loop:

1. User visits website with profile, browsing history . . .
2. Choose content to display on website.

Learning to interact: example #2

Website operator

Loop:

1. User visits website with profile, browsing history . . .
2. Choose content to display on website.
3. Observe user reaction to content (*e.g.*, click, "like").

Learning to interact: example #2

Website operator

Loop:

1. User visits website with profile, browsing history . . .
2. Choose content to display on website.
3. Observe user reaction to content (*e.g.*, click, "like").

Goal: choose content that yield desired user behavior.

Contextual bandit problem

For $t = 1, 2, \ldots, T$:

Contextual bandit problem

For $t = 1, 2, \ldots, T$:

1. Observe context $x_t \in \mathcal{X}$. [e.g., user profile, search query]

Contextual bandit problem

For $t = 1, 2, \ldots, T$:

1. Observe context $x_t \in \mathcal{X}$. [e.g., user profile, search query]
2. Choose action $a_t \in \mathcal{A}$. [e.g., ad to display]

Contextual bandit problem

For $t = 1, 2, \ldots, T$:

1. Observe context $x_t \in \mathcal{X}$. [e.g., user profile, search query]
2. Choose action $a_t \in \mathcal{A}$. [e.g., ad to display]
3. Collect reward $r_t(a_t) \in [0, 1]$. [e.g., 1 if click, 0 otherwise]

Contextual bandit problem

For $t = 1, 2, \ldots, T$:

0. Nature draws $(x_t, \boldsymbol{r}_t)$ from dist. $\mathcal{D}$ over $\mathcal{X} \times [0,1]^{\mathcal{A}}$.
1. Observe context $x_t \in \mathcal{X}$. [e.g., user profile, search query]
2. Choose action $a_t \in \mathcal{A}$. [e.g., ad to display]
3. Collect reward $r_t(a_t) \in [0,1]$. [e.g., 1 if click, 0 otherwise]

Contextual bandit problem

For $t = 1, 2, \ldots, T$:

0. Nature draws $(x_t, \boldsymbol{r}_t)$ from dist. $\mathcal{D}$ over $\mathcal{X} \times [0,1]^{\mathcal{A}}$.
1. Observe context $x_t \in \mathcal{X}$. [e.g., user profile, search query]
2. Choose action $a_t \in \mathcal{A}$. [e.g., ad to display]
3. Collect reward $r_t(a_t) \in [0,1]$. [e.g., 1 if click, 0 otherwise]

Task: choose a_t's that yield high expected reward (w.r.t. $\mathcal{D}$).

Contextual bandit problem

For $t = 1, 2, \ldots, T$:

0. Nature draws $(x_t, \boldsymbol{r}_t)$ from dist. $\mathcal{D}$ over $\mathcal{X} \times [0,1]^{\mathcal{A}}$.
1. Observe context $x_t \in \mathcal{X}$. [e.g., user profile, search query]
2. Choose action $a_t \in \mathcal{A}$. [e.g., ad to display]
3. Collect reward $r_t(a_t) \in [0, 1]$. [e.g., 1 if click, 0 otherwise]

Task: choose a_t's that yield high expected reward (w.r.t. $\mathcal{D}$).

Contextual: use features x_t to choose good actions a_t.

Contextual bandit problem

For $t = 1, 2, \dots, T$:

0. Nature draws $(x_t, \boldsymbol{r}_t)$ from dist. $\mathcal{D}$ over $\mathcal{X} \times [0,1]^{\mathcal{A}}$.
1. Observe context $x_t \in \mathcal{X}$. [e.g., user profile, search query]
2. Choose action $a_t \in \mathcal{A}$. [e.g., ad to display]
3. Collect reward $r_t(a_t) \in [0,1]$. [e.g., 1 if click, 0 otherwise]

Task: choose a_t's that yield high expected reward (w.r.t. $\mathcal{D}$).

Contextual: use features x_t to choose good actions a_t.

Bandit: $r_t(a)$ for $a \neq a_t$ is not observed.

(Non-bandit setting: whole reward vector $\boldsymbol{r}_t \in [0,1]^{\mathcal{A}}$ is observed.)

Challenges

Challenges

1. Exploration vs. exploitation.
    - Use what you've already learned (exploit), but also learn about actions that could be good (explore).
    - Must balance to get good statistical performance.

Challenges

1. Exploration vs. exploitation.
   - Use what you've already learned (exploit), but also learn about actions that could be good (explore).
   - Must balance to get good statistical performance.
2. Must use context.
   - Want to do as well as the best **policy** (i.e., decision rule)

     $$\pi : \text{context } x \;\mapsto\; \text{action } a$$

     from some **policy class** Π (a set of decision rules).
   - Computationally constrained w/ large Π (e.g., all decision trees).

Challenges

1. Exploration vs. exploitation.
   - Use what you've already learned (exploit), but also learn about actions that could be good (explore).
   - Must balance to get good statistical performance.
2. Must use context.
   - Want to do as well as the best **policy** (i.e., decision rule)

     $$\pi : \text{context } x \;\mapsto\; \text{action } a$$

     from some **policy class** Π (a set of decision rules).
   - Computationally constrained w/ large Π (e.g., all decision trees).
3. Selection bias, especially while *exploiting*.

Learning objective

Regret (*i.e.*, relative performance) to a policy class Π:

$$\underbrace{\max_{\pi \in \Pi} \frac{1}{T} \sum_{t=1}^{T} r_t(\pi(x_t))}_{\text{average reward of best policy}} \quad - \quad \underbrace{\frac{1}{T} \sum_{t=1}^{T} r_t(a_t)}_{\text{average reward of learner}}$$

Learning objective

Regret (*i.e.*, relative performance) to a policy class Π:

$$\underbrace{\max_{\pi \in \Pi} \frac{1}{T} \sum_{t=1}^{T} r_t(\pi(x_t))}_{\text{average reward of best policy}} \quad - \quad \underbrace{\frac{1}{T} \sum_{t=1}^{T} r_t(a_t)}_{\text{average reward of learner}}$$

Strong benchmark if Π contains a policy w/ high expected reward!

Learning objective

Regret (*i.e.*, relative performance) to a policy class Π:

$$\underbrace{\max_{\pi \in \Pi} \frac{1}{T} \sum_{t=1}^{T} r_t(\pi(x_t))}_{\text{average reward of best policy}} \quad - \quad \underbrace{\frac{1}{T} \sum_{t=1}^{T} r_t(a_t)}_{\text{average reward of learner}}$$

Strong benchmark if Π contains a policy w/ high expected reward!

Goal: regret $\to 0$ as fast as possible as $T \to \infty$.

Our result

New fast and simple algorithm for contextual bandits.

Our result

New fast and simple algorithm for contextual bandits.

- Operates via reduction to supervised learning
  (with computationally-efficient reduction).

Our result

New fast and simple algorithm for contextual bandits.

- Operates via reduction to supervised learning
  (with computationally-efficient reduction).
- Statistically (near) optimal regret bound.

Need for exploration

No-exploration approach:

Need for exploration

No-exploration approach:

1. Using historical data, learn a "reward predictor" for each action $a \in \mathcal{A}$ based on context $x \in \mathcal{X}$:

$$\hat{r}(a \mid x).$$

Need for exploration

No-exploration approach:

1. Using historical data, learn a "reward predictor" for each action $a \in \mathcal{A}$ based on context $x \in \mathcal{X}$:

$$\hat{r}(a \mid x)\,.$$

2. Then deploy policy $\hat{\pi}$, given by

$$\hat{\pi}(x) \;:=\; \arg\max_{a \in \mathcal{A}} \hat{r}(a \mid x)\,,$$

and collect more data.

Need for exploration

No-exploration approach:

1. Using historical data, learn a "reward predictor" for each action $a \in \mathcal{A}$ based on context $x \in \mathcal{X}$:

$$\hat{r}(a \mid x)\,.$$

2. Then deploy policy $\hat{\pi}$, given by

$$\hat{\pi}(x) \;:=\; \arg\max_{a \in \mathcal{A}} \hat{r}(a \mid x)\,,$$

and collect more data.

Suffers from **selection bias**.

Using no-exploration

Example: two contexts $\{X, Y\}$, two actions $\{A, B\}$.

Suppose initial policy says $\hat{\pi}(X) = A$ and $\hat{\pi}(Y) = B$.

Using no-exploration

Example: two contexts $\{X, Y\}$, two actions $\{A, B\}$.

Suppose initial policy says $\hat{\pi}(X) = A$ and $\hat{\pi}(Y) = B$.

Observed rewards

	A	B
X	0.7	—
Y	—	0.1

Using no-exploration

Example: two contexts $\{X, Y\}$, two actions $\{A, B\}$.

Suppose initial policy says $\hat{\pi}(X) = A$ and $\hat{\pi}(Y) = B$.

Observed rewards

	A	B
X	0.7	—
Y	—	0.1

Reward estimates

	A	B
X	**0.7**	0.5
Y	**0.5**	0.1

Using no-exploration

Example: two contexts $\{X, Y\}$, two actions $\{A, B\}$.

Suppose initial policy says $\hat{\pi}(X) = A$ and $\hat{\pi}(Y) = B$.

Observed rewards

	A	B
X	0.7	—
Y	—	0.1

Reward estimates

	A	B
X	**0.7**	0.5
Y	**0.5**	0.1

New policy: $\hat{\pi}'(X) = \hat{\pi}'(Y) = A$.

Using no-exploration

Example: two contexts $\{X, Y\}$, two actions $\{A, B\}$.

Suppose initial policy says $\hat{\pi}(X) = A$ and $\hat{\pi}(Y) = B$.

Observed rewards

	A	B
X	0.7	—
Y	—	0.1

Reward estimates

	A	B
X	**0.7**	0.5
Y	**0.5**	0.1

New policy: $\hat{\pi}'(X) = \hat{\pi}'(Y) = A$.

Observed rewards

	A	B
X	0.7	—
Y	0.3	0.1

Using no-exploration

Example: two contexts $\{X, Y\}$, two actions $\{A, B\}$.

Suppose initial policy says $\hat{\pi}(X) = A$ and $\hat{\pi}(Y) = B$.

Observed rewards

	A	B
X	0.7	—
Y	—	0.1

Reward estimates

	A	B
X	**0.7**	0.5
Y	**0.5**	0.1

New policy: $\hat{\pi}'(X) = \hat{\pi}'(Y) = A$.

Observed rewards

	A	B
X	0.7	—
Y	0.3	0.1

Reward estimates

	A	B
X	**0.7**	0.5
Y	**0.3**	0.1

Using no-exploration

Example: two contexts $\{X, Y\}$, two actions $\{A, B\}$.

Suppose initial policy says $\hat{\pi}(X) = A$ and $\hat{\pi}(Y) = B$.

Observed rewards

	A	B
X	0.7	—
Y	—	0.1

Reward estimates

	A	B
X	**0.7**	0.5
Y	**0.5**	0.1

New policy: $\hat{\pi}'(X) = \hat{\pi}'(Y) = A$.

Observed rewards

	A	B
X	0.7	—
Y	0.3	0.1

Reward estimates

	A	B
X	**0.7**	0.5
Y	**0.3**	0.1

Never try action B in context X.

Using no-exploration

Example: two contexts $\{X, Y\}$, two actions $\{A, B\}$.

Suppose initial policy says $\hat{\pi}(X) = A$ and $\hat{\pi}(Y) = B$.

Observed rewards

	A	B
X	0.7	—
Y	—	0.1

Reward estimates

	A	B
X	**0.7**	0.5
Y	**0.5**	0.1

New policy: $\hat{\pi}'(X) = \hat{\pi}'(Y) = A$.

Observed rewards

	A	B
X	0.7	—
Y	0.3	0.1

Reward estimates

	A	B
X	**0.7**	0.5
Y	**0.3**	0.1

True rewards

	A	B
X	0.7	**1.0**
Y	**0.3**	0.1

Never try action B in context X. $\Omega(1)$ **regret**.

Dealing with policies

Feedback in round t: reward of chosen action $r_t(a_t)$.

- Tells us about policies $\pi \in \Pi$ s.t. $\pi(x_t) = a_t$.
- Not informative about other policies!

Dealing with policies

Feedback in round t: reward of chosen action $r_t(a_t)$.

- Tells us about policies $\pi \in \Pi$ s.t. $\pi(x_t) = a_t$.
- Not informative about other policies!

Possible approach: track average reward of each $\pi \in \Pi$.

Dealing with policies

Feedback in round t: reward of chosen action $r_t(a_t)$.

- Tells us about policies $\pi \in \Pi$ s.t. $\pi(x_t) = a_t$.
- Not informative about other policies!

Possible approach: track average reward of each $\pi \in \Pi$.

- **Exp4** (Auer, Cesa-Bianchi, Freund, & Schapire, FOCS 1995).

Dealing with policies

Feedback in round t: reward of chosen action $r_t(a_t)$.

- Tells us about policies $\pi \in \Pi$ s.t. $\pi(x_t) = a_t$.
- Not informative about other policies!

Possible approach: track average reward of each $\pi \in \Pi$.

- **Exp4** (Auer, Cesa-Bianchi, Freund, & Schapire, FOCS 1995).
- Statistically optimal regret bound $O\left(\sqrt{\frac{K \log N}{T}}\right)$
  for $K := |\mathcal{A}|$ actions and $N := |\Pi|$ policies after T rounds.

Dealing with policies

Feedback in round t: reward of chosen action $r_t(a_t)$.

- Tells us about policies $\pi \in \Pi$ s.t. $\pi(x_t) = a_t$.
- Not informative about other policies!

Possible approach: track average reward of each $\pi \in \Pi$.

- **Exp4** (Auer, Cesa-Bianchi, Freund, & Schapire, FOCS 1995).
- Statistically optimal regret bound $O\left(\sqrt{\frac{K \log N}{T}}\right)$
  for $K := |\mathcal{A}|$ actions and $N := |\Pi|$ policies after T rounds.
- Explicit bookkeeping is **computationally intractable** for large N.

Dealing with policies

Feedback in round t: reward of chosen action $r_t(a_t)$.

- Tells us about policies $\pi \in \Pi$ s.t. $\pi(x_t) = a_t$.
- Not informative about other policies!

Possible approach: track average reward of each $\pi \in \Pi$.

- **Exp4** (Auer, Cesa-Bianchi, Freund, & Schapire, FOCS 1995).
- Statistically optimal regret bound $O\left(\sqrt{\frac{K \log N}{T}}\right)$
  for $K := |\mathcal{A}|$ actions and $N := |\Pi|$ policies after T rounds.
- Explicit bookkeeping is **computationally intractable** for large N.

But perhaps policy class Π has some structure ...

Hypothetical “full-information” setting

If we observed rewards for all actions . . .

Hypothetical "full-information" setting

If we observed rewards for <u>all actions</u> . . .

- Like **supervised learning**, have *labeled data* after t rounds:

$$(x_1, \boldsymbol{\rho}_1), \ldots, (x_t, \boldsymbol{\rho}_t) \in \mathcal{X} \times \mathbb{R}^{\mathcal{A}}.$$

Hypothetical "full-information" setting

If we observed rewards for <u>all actions</u> . . .

- Like **supervised learning**, have *labeled data* after t rounds:

$$(x_1, \boldsymbol{\rho}_1), \ldots, (x_t, \boldsymbol{\rho}_t) \in \mathcal{X} \times \mathbb{R}^{\mathcal{A}}.$$

context	$\longrightarrow$	features
actions	$\longrightarrow$	classes
rewards	$\longrightarrow$	$-$costs
policy	$\longrightarrow$	classifier

Hypothetical "full-information" setting

If we observed rewards for <u>all actions</u> . . .

- Like **supervised learning**, have *labeled data* after t rounds:

$$(x_1, \boldsymbol{\rho}_1), \ldots, (x_t, \boldsymbol{\rho}_t) \in \mathcal{X} \times \mathbb{R}^{\mathcal{A}}.$$

context	$\longrightarrow$	features
actions	$\longrightarrow$	classes
rewards	$\longrightarrow$	$-$costs
policy	$\longrightarrow$	classifier

- Can often exploit structure of Π to get tractable algorithms.
  Abstraction: arg max oracle (AMO)

$$\mathrm{AMO}\Big(\big\{(x_i, \boldsymbol{\rho}_i)\big\}_{i=1}^{t}\Big) \;:=\; \arg\max_{\pi \in \Pi} \sum_{i=1}^{t} \rho_i(\pi(x_i)).$$

Hypothetical "full-information" setting

If we observed rewards for all actions . . .

- Like **supervised learning**, have *labeled data* after t rounds:

$$(x_1, \boldsymbol{\rho}_1), \ldots, (x_t, \boldsymbol{\rho}_t) \in \mathcal{X} \times \mathbb{R}^{\mathcal{A}}.$$

context	$\longrightarrow$	features
actions	$\longrightarrow$	classes
rewards	$\longrightarrow$	$-$costs
policy	$\longrightarrow$	classifier

- Can often exploit structure of Π to get tractable algorithms.
  Abstraction: arg max oracle (AMO)

$$\mathrm{AMO}\Big(\{(x_i, \boldsymbol{\rho}_i)\}_{i=1}^{t}\Big) \;:=\; \operatorname*{arg\,max}_{\pi \in \Pi} \sum_{i=1}^{t} \rho_i(\pi(x_i)).$$

Can't directly use this in bandit setting.

Using AMO with some exploration

Explore-then-exploit

Using AMO with some exploration

Explore-then-exploit

1. In first τ rounds, choose $a_t \in \mathcal{A}$ u.a.r. to get *unbiased estimates* $\hat{\boldsymbol{r}}_t$ of $\boldsymbol{r}_t$ for all $t \leq \tau$.

Using AMO with some exploration

Explore-then-exploit

1. In first τ rounds, choose $a_t \in \mathcal{A}$ u.a.r. to get *unbiased estimates* $\hat{\boldsymbol{r}}_t$ of $\boldsymbol{r}_t$ for all $t \leq \tau$.
2. Get $\hat{\pi} := \mathrm{AMO}(\{(x_t, \hat{\boldsymbol{r}}_t)\}_{t=1}^{\tau})$.

Using AMO with some exploration

Explore-then-exploit

1. In first τ rounds, choose $a_t \in \mathcal{A}$ u.a.r. to get *unbiased estimates* $\hat{\boldsymbol{r}}_t$ of $\boldsymbol{r}_t$ for all $t \leq \tau$.
2. Get $\hat{\pi} := \mathrm{AMO}(\{(x_t, \hat{\boldsymbol{r}}_t)\}_{t=1}^{\tau})$.
3. Henceforth use $a_t := \hat{\pi}(x_t)$, for $t = \tau+1, \tau+2, \ldots, T$.

Using AMO with some exploration

Explore-then-exploit

1. In first τ rounds, choose $a_t \in \mathcal{A}$ u.a.r. to get *unbiased estimates* $\hat{\boldsymbol{r}}_t$ of $\boldsymbol{r}_t$ for all $t \leq \tau$.
2. Get $\hat{\pi} := \mathrm{AMO}(\{(x_t, \hat{\boldsymbol{r}}_t)\}_{t=1}^{\tau})$.
3. Henceforth use $a_t := \hat{\pi}(x_t)$, for $t = \tau+1, \tau+2, \ldots, T$.

Regret bound with best τ: $\sim T^{-1/3}$ (**sub-optimal**).

(Dependencies on $|\mathcal{A}|$ and $|\Pi|$ hidden.)

Previous contextual bandit algorithms

Previous contextual bandit algorithms

Exp4 (Auer, Cesa-Bianchi, Freund, & Schapire, FOCS 1995).
Optimal regret, but explicitly enumerates Π.

Previous contextual bandit algorithms

Exp4 (Auer, Cesa-Bianchi, Freund, & Schapire, FOCS 1995).
Optimal regret, but explicitly enumerates Π.

Greedy (Langford & Zhang, NIPS 2007)
Sub-optimal regret, but one call to AMO.

Previous contextual bandit algorithms

Exp4 (Auer, Cesa-Bianchi, Freund, & Schapire, FOCS 1995).
Optimal regret, but explicitly enumerates Π.

Greedy (Langford & Zhang, NIPS 2007)
Sub-optimal regret, but one call to AMO.

Monster (Dudik, Hsu, Kale, Karampatziakis, Langford, Reyzin, & Zhang, UAI 2011)
Near optimal regret, but $O(T^6)$ calls to AMO.

Our result

Let $K := |\mathcal{A}|$ and $N := |\Pi|$.

Our result: a new, fast and simple algorithm.

- Regret bound: $\tilde{O}\left(\sqrt{\frac{K \log N}{T}}\right)$.
  Near optimal.
- # calls to AMO: $\tilde{O}\left(\sqrt{\frac{TK}{\log N}}\right)$.
  Less than once per round!

Rest of the talk

Components of the new algorithm:

Importance-weighted LOw-Variance Epoch-Timed Oracleized CONtextual BANDITS

1. "Classical" tricks: randomization, inverse propensity weighting.
2. Efficient algorithm for balancing exploration/exploitation.
3. Additional tricks: warm-start and epoch structure.

1. Classical tricks

What would've happened if I had done X?

For $t = 1, 2, \ldots, T$:

0. Nature draws $(x_t, \boldsymbol{r}_t)$ from dist. $\mathcal{D}$ over $\mathcal{X} \times [0,1]^{\mathcal{A}}$.
1. Observe context $x_t \in \mathcal{X}$. [e.g., user profile, search query]
2. Choose action $a_t \in \mathcal{A}$. [e.g., ad to display]
3. Collect reward $r_t(a_t) \in [0,1]$. [e.g., 1 if click, 0 otherwise]

What would've happened if I had done X?

For $t = 1, 2, \ldots, T$:

0. Nature draws $(x_t, \boldsymbol{r}_t)$ from dist. $\mathcal{D}$ over $\mathcal{X} \times [0,1]^{\mathcal{A}}$.
1. Observe context $x_t \in \mathcal{X}$. [e.g., user profile, search query]
2. Choose action $a_t \in \mathcal{A}$. [e.g., ad to display]
3. Collect reward $r_t(a_t) \in [0,1]$. [e.g., 1 if click, 0 otherwise]

Q: How do I learn about $r_t(a)$ for actions a I don't actually take?

What would've happened if I had done X?

For $t = 1, 2, \ldots, T$:

0. Nature draws $(x_t, \boldsymbol{r}_t)$ from dist. $\mathcal{D}$ over $\mathcal{X} \times [0,1]^{\mathcal{A}}$.
1. Observe context $x_t \in \mathcal{X}$. [e.g., user profile, search query]
2. Choose action $a_t \in \mathcal{A}$. [e.g., ad to display]
3. Collect reward $r_t(a_t) \in [0,1]$. [e.g., 1 if click, 0 otherwise]

Q: How do I learn about $r_t(a)$ for actions a I don't actually take?

A: *Randomize*. Draw $a_t \sim \boldsymbol{p}_t$ for some pre-specified prob. dist. $\boldsymbol{p}_t$.

Inverse propensity weighting (Horvitz & Thompson, JASA 1952)

Importance-weighted estimate of reward from round t:

$$\forall a \in \mathcal{A}. \quad \hat{r}_t(a) := \frac{r_t(a_t) \cdot \mathbb{1}\{a = a_t\}}{p_t(a_t)}$$

Inverse propensity weighting (Horvitz & Thompson, JASA 1952)

Importance-weighted estimate of reward from round t:

$$\forall a \in \mathcal{A}. \quad \hat{r}_t(a) := \frac{r_t(a_t) \cdot \mathbb{1}\{a = a_t\}}{p_t(a_t)} = \begin{cases} \dfrac{r_t(a_t)}{p_t(a_t)} & \text{if } a = a_t\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

Inverse propensity weighting (Horvitz & Thompson, JASA 1952)

Importance-weighted estimate of reward from round t:

$$\forall a \in \mathcal{A}. \quad \hat{r}_t(a) := \frac{r_t(a_t) \cdot \mathbb{1}\{a = a_t\}}{p_t(a_t)} = \begin{cases} \dfrac{r_t(a_t)}{p_t(a_t)} & \text{if } a = a_t \,, \\[2ex] 0 & \text{otherwise} \,. \end{cases}$$

Unbiasedness:

$$\mathbb{E}_{a_t \sim \boldsymbol{p}_t}\left[\hat{r}_t(a)\right] = \sum_{a' \in \mathcal{A}} p_t(a') \cdot \frac{r_t(a') \cdot \mathbb{1}\{a = a'\}}{p_t(a')} = r_t(a) \,.$$

Inverse propensity weighting (Horvitz & Thompson, JASA 1952)

Importance-weighted estimate of reward from round t:

$$\forall a \in \mathcal{A}. \quad \hat{r}_t(a) := \frac{r_t(a_t) \cdot \mathbb{1}\{a = a_t\}}{p_t(a_t)} = \begin{cases} \dfrac{r_t(a_t)}{p_t(a_t)} & \text{if } a = a_t\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

Unbiasedness:

$$\mathbb{E}_{a_t \sim \boldsymbol{p}_t}\left[\hat{r}_t(a)\right] = \sum_{a' \in \mathcal{A}} p_t(a') \cdot \frac{r_t(a') \cdot \mathbb{1}\{a = a'\}}{p_t(a')} = r_t(a)\,.$$

Range and variance: upper-bounded by $\frac{1}{p_t(a)}$.

Inverse propensity weighting (Horvitz & Thompson, JASA 1952)

Importance-weighted estimate of reward from round t:

$$\forall a \in \mathcal{A}. \quad \hat{r}_t(a) \ := \ \frac{r_t(a_t) \cdot \mathbb{1}\{a = a_t\}}{p_t(a_t)} \ = \ \begin{cases} \dfrac{r_t(a_t)}{p_t(a_t)} & \text{if } a = a_t \,, \\[2ex] 0 & \text{otherwise} \,. \end{cases}$$

Unbiasedness:

$$\mathbb{E}_{a_t \sim \boldsymbol{p}_t}\left[\hat{r}_t(a)\right] \ = \ \sum_{a' \in \mathcal{A}} p_t(a') \cdot \frac{r_t(a') \cdot \mathbb{1}\{a = a'\}}{p_t(a')} \ = \ r_t(a) \,.$$

Range and variance: upper-bounded by $\frac{1}{p_t(a)}$.

Estimate avg. reward of policy: $\widehat{\text{Rew}}_t(\pi) := \frac{1}{t} \sum_{i=1}^{t} \hat{r}_i(\pi(x_i))$.

Inverse propensity weighting (Horvitz & Thompson, JASA 1952)

Importance-weighted estimate of reward from round t:

$$\forall a \in \mathcal{A}. \quad \hat{r}_t(a) \;:=\; \frac{r_t(a_t) \cdot \mathbb{1}\{a = a_t\}}{p_t(a_t)} \;=\; \begin{cases} \dfrac{r_t(a_t)}{p_t(a_t)} & \text{if } a = a_t\,, \\[2ex] 0 & \text{otherwise}\,. \end{cases}$$

Unbiasedness:

$$\mathbb{E}_{a_t \sim \boldsymbol{p}_t}\left[\hat{r}_t(a)\right] \;=\; \sum_{a' \in \mathcal{A}} p_t(a') \cdot \frac{r_t(a') \cdot \mathbb{1}\{a = a'\}}{p_t(a')} \;=\; r_t(a)\,.$$

Range and variance: upper-bounded by $\frac{1}{p_t(a)}$.

Estimate avg. reward of policy: $\widehat{\mathrm{Rew}}_t(\pi) := \frac{1}{t}\sum_{i=1}^{t} \hat{r}_i(\pi(x_i))$.

How should we choose the $\boldsymbol{p}_t$?

Hedging over policies

Get action distributions via policy distributions.

$$\underbrace{(\boldsymbol{Q}, x)}_{(\text{policy distribution}, \text{context})} \quad \longmapsto \quad \underbrace{\boldsymbol{p}}_{\text{action distribution}}$$

Hedging over policies

Get action distributions via policy distributions.

$$\underbrace{(\boldsymbol{Q}, x)}_{(\text{policy distribution, context})} \quad \longmapsto \quad \underbrace{\boldsymbol{p}}_{\text{action distribution}}$$

Policy distribution: $\boldsymbol{Q} = (Q(\pi) : \pi \in \Pi)$
probability dist. over policies π in the policy class Π

Hedging over policies

Get action distributions via policy distributions.

$$\underbrace{(\boldsymbol{Q}, x)}_{\text{(policy distribution, context)}} \quad \longmapsto \quad \underbrace{\boldsymbol{p}}_{\text{action distribution}}$$

Pick initial distribution $\boldsymbol{Q}_1$ over policies Π.
for round $t = 1, 2, \ldots$ **do**
  Nature draws $(x_t, \boldsymbol{r}_t)$ from dist. $\mathcal{D}$ over $\mathcal{X} \times [0, 1]^{\mathcal{A}}$.
  Observe context x_t.
  Compute distribution $\boldsymbol{p}_t$ over $\mathcal{A}$ (using $\boldsymbol{Q}_t$ and x_t).
  Pick action $a_t \sim \boldsymbol{p}_t$.
  Collect reward $r_t(a_t)$.
  Compute new distribution $\boldsymbol{Q}_{t+1}$ over policies Π.
end for

2. Efficient construction of good policy distributions

Our approach

Q: How do we choose $\boldsymbol{Q}_t$ for good exploration/exploitation?

Our approach

Q: How do we choose $\boldsymbol{Q}_t$ for good exploration/exploitation?

Caveat: $\boldsymbol{Q}_t$ must be efficiently computable + representable!

Our approach

Q: How do we choose $\boldsymbol{Q}_t$ for good exploration/exploitation?

Caveat: $\boldsymbol{Q}_t$ must be efficiently computable + representable!

Our approach:

1. Define convex feasibility problem (over distributions $\boldsymbol{Q}$ on Π) such that solutions yield (near) optimal regret bounds.

Our approach

Q: How do we choose $\boldsymbol{Q}_t$ for good exploration/exploitation?

Caveat: $\boldsymbol{Q}_t$ must be efficiently computable + representable!

Our approach:

1. Define convex feasibility problem (over distributions $\boldsymbol{Q}$ on Π) such that solutions yield (near) optimal regret bounds.
2. Design algorithm that finds a *sparse* solution $\boldsymbol{Q}$.

Our approach

Q: How do we choose $\boldsymbol{Q}_t$ for good exploration/exploitation?

Caveat: $\boldsymbol{Q}_t$ must be efficiently computable + representable!

Our approach:

1. Define convex feasibility problem (over distributions $\boldsymbol{Q}$ on Π) such that solutions yield (near) optimal regret bounds.

2. Design algorithm that finds a *sparse* solution $\boldsymbol{Q}$.

   Algorithm only accesses Π via calls to AMO
   $\Longrightarrow \mathsf{nnz}(\boldsymbol{Q}) = O(\#\ \mathrm{AMO}\ \text{calls})$

The “good policy distribution” problem

Convex feasibility problem for policy distribution $\boldsymbol{Q}$

The "good policy distribution" problem

Convex feasibility problem for policy distribution $\boldsymbol{Q}$

$$\sum_{\pi \in \Pi} Q(\pi) \cdot \widehat{\mathrm{Reg}}_t(\pi) \;\leq\; \sqrt{\frac{K \log N}{t}} \qquad \text{(Low regret)}$$

The "good policy distribution" problem

Convex feasibility problem for policy distribution $\boldsymbol{Q}$

$$\sum_{\pi \in \Pi} Q(\pi) \cdot \widehat{\mathrm{Reg}}_t(\pi) \;\le\; \sqrt{\frac{K \log N}{t}} \qquad \text{(Low regret)}$$

$$\widehat{\mathrm{var}}\left(\widehat{\mathrm{Rew}}_t(\pi)\right) \;\le\; b(\pi) \quad \forall \pi \in \Pi \qquad \text{(Low variance)}$$

The "good policy distribution" problem

Convex feasibility problem for policy distribution $\boldsymbol{Q}$

$$\sum_{\pi \in \Pi} Q(\pi) \cdot \widehat{\mathrm{Reg}}_t(\pi) \ \leq \ \sqrt{\frac{K \log N}{t}} \qquad \text{(Low regret)}$$

$$\widehat{\mathrm{var}}\left(\widehat{\mathrm{Rew}}_t(\pi)\right) \ \leq \ b(\pi) \quad \forall \pi \in \Pi \qquad \text{(Low variance)}$$

Using feasible $\boldsymbol{Q}_t$ in round t gives near-optimal regret.

The "good policy distribution" problem

Convex feasibility problem for policy distribution Q

$$\sum_{\pi \in \Pi} Q(\pi) \cdot \widehat{\mathrm{Reg}}_t(\pi) \;\leq\; \sqrt{\frac{K \log N}{t}} \qquad \text{(Low regret)}$$

$$\widehat{\mathrm{var}}\Big(\widehat{\mathrm{Rew}}_t(\pi)\Big) \;\leq\; b(\pi) \quad \forall \pi \in \Pi \qquad \text{(Low variance)}$$

Using feasible Q_t in round t gives near-optimal regret.

But $|\Pi|$ variables and $>|\Pi|$ constraints, ...

Solving the convex feasibility problem

Solver for "good policy distribution" problem

(Technical detail: Q can be a sub-distribution that sums to less than one.)

Solving the convex feasibility problem

Solver for "good policy distribution" problem

Start with some $\boldsymbol{Q}$ (e.g., $\boldsymbol{Q} := \mathbf{0}$), then repeat:

(Technical detail: $\boldsymbol{Q}$ can be a sub-distribution that sums to less than one.)

Solving the convex feasibility problem

Solver for "good policy distribution" problem

Start with some $\boldsymbol{Q}$ (e.g., $\boldsymbol{Q} := \mathbf{0}$), then repeat:

1. If "low regret" constraint violated, then fix by rescaling:

$$\boldsymbol{Q} \;:=\; c\boldsymbol{Q}$$

   for some $c < 1$.

(Technical detail: $\boldsymbol{Q}$ can be a sub-distribution that sums to less than one.)

Solving the convex feasibility problem

Solver for "good policy distribution" problem
Start with some $\boldsymbol{Q}$ (e.g., $\boldsymbol{Q} := \mathbf{0}$), then repeat:

1. If "low regret" constraint violated, then fix by rescaling:

$$\boldsymbol{Q} \; := \; c\boldsymbol{Q}$$

   for some $c < 1$.

2. Find most violated "low variance" constraint—say, corresponding to policy $\widetilde{\pi}$—and update

$$Q(\widetilde{\pi}) \; := \; Q(\widetilde{\pi}) + \alpha \,.$$

($c < 1$ and $\alpha > 0$ have closed-form formulae.)

(Technical detail: $\boldsymbol{Q}$ can be a sub-distribution that sums to less than one.)

Solving the convex feasibility problem

Solver for "good policy distribution" problem

Start with some $\boldsymbol{Q}$ (e.g., $\boldsymbol{Q} := \boldsymbol{0}$), then repeat:

1. If "low regret" constraint violated, then fix by rescaling:

$$\boldsymbol{Q} \;:=\; c\boldsymbol{Q}$$

   for some $c < 1$.

2. Find most violated "low variance" constraint—say, corresponding to policy $\widetilde{\pi}$—and update

$$Q(\widetilde{\pi}) \;:=\; Q(\widetilde{\pi}) + \alpha\,.$$

   (If no such violated constraint, stop and return $\boldsymbol{Q}$.)

($c < 1$ and $\alpha > 0$ have closed-form formulae.)

(Technical detail: $\boldsymbol{Q}$ can be a sub-distribution that sums to less than one.)

Implementation via AMO

Finding "low variance" constraint violation:

1. Create fictitious rewards for each $i = 1, 2, \ldots, t$:

$$\widetilde{r}_i(a) \;:=\; \hat{r}_i(a) + \frac{\mu}{Q(a|x_i)} \quad \forall a \in \mathcal{A}\,,$$

   where $\mu \approx \sqrt{(\log N)/(Kt)}$.

2. Obtain $\widetilde{\pi} := \mathrm{AMO}\Big(\big\{(x_i, \widetilde{\boldsymbol{r}}_i)\big\}_{i=1}^{t}\Big)$.

3. $\widetilde{\mathrm{Rew}}_t(\widetilde{\pi}) >$ threshold iff $\widetilde{\pi}$'s "low variance" constraint is violated.

Iteration bound

Solver is **coordinate descent** for minimizing potential function

$$\Phi(\boldsymbol{Q}) \;:=\; c_1 \cdot \widehat{\mathbb{E}}_x\big[\mathrm{RE}(\textbf{uniform} \,\|\, \boldsymbol{Q}(\cdot|x))\big] + c_2 \cdot \sum_{\pi \in \Pi} Q(\pi)\widehat{\mathrm{Reg}}_t(\pi)\,.$$

(Actually use $(1-\varepsilon)\cdot\boldsymbol{Q} + \varepsilon\cdot$ **uniform** inside RE expression.)

Iteration bound

Solver is **coordinate descent** for minimizing potential function

$$\Phi(\boldsymbol{Q}) \; := \; c_1 \cdot \widehat{\mathbb{E}}_x \left[\mathrm{RE}(\mathbf{uniform} \| \boldsymbol{Q}(\cdot|x)) \right] + c_2 \cdot \sum_{\pi \in \Pi} Q(\pi) \widehat{\mathrm{Reg}}_t(\pi) \, .$$

(Partial derivative w.r.t. $Q(\pi)$ is "low variance" constraint for π.)

(Actually use $(1-\varepsilon) \cdot \boldsymbol{Q} + \varepsilon \cdot \mathbf{uniform}$ inside RE expression.)

Iteration bound

Solver is **coordinate descent** for minimizing potential function

$$\Phi(\boldsymbol{Q}) \; := \; c_1 \cdot \widehat{\mathbb{E}}_x\left[\mathrm{RE}(\mathbf{uniform} \| \boldsymbol{Q}(\cdot|x))\right] + c_2 \cdot \sum_{\pi \in \Pi} Q(\pi)\widehat{\mathrm{Reg}}_t(\pi)\,.$$

(Partial derivative w.r.t. $Q(\pi)$ is "low variance" constraint for π.)

Returns a feasible solution after

$$\tilde{O}\left(\sqrt{\frac{Kt}{\log N}}\right) \text{ steps}\,.$$

(Actually use $(1-\varepsilon)\cdot\boldsymbol{Q} + \varepsilon\cdot\mathbf{uniform}$ inside RE expression.)

Algorithm

Pick initial distribution $\boldsymbol{Q}_1$ over policies Π.
for round $t = 1, 2, \ldots$ **do**
    Nature draws $(x_t, \boldsymbol{r}_t)$ from dist. $\mathcal{D}$ over $\mathcal{X} \times [0,1]^{\mathcal{A}}$.
    Observe context x_t.
    Compute action distribution $\boldsymbol{p}_t := \boldsymbol{Q}_t(\,\cdot\,|x_t)$.
    Pick action $a_t \sim \boldsymbol{p}_t$.
    Collect reward $r_t(a_t)$.
    Compute new policy distribution $\boldsymbol{Q}_{t+1}$ using coordinate descent + AMO.
end for

Recap

Recap

Feasible solution to "good policy distribution problem" gives near optimal regret bound.

Recap

Feasible solution to "good policy distribution problem" gives near optimal regret bound.

New coordinate descent algorithm:
repeatedly find a violated constraint and adjust $\boldsymbol{Q}$ to satisfy it.

Recap

Feasible solution to "good policy distribution problem" gives near optimal regret bound.

New coordinate descent algorithm:
repeatedly find a violated constraint and adjust $\boldsymbol{Q}$ to satisfy it.

Analysis:
In round t,

$$\mathrm{nnz}(\boldsymbol{Q}_{t+1}) \;=\; O(\#\ \mathrm{AMO\ calls}) \;=\; \tilde{O}\left(\sqrt{\frac{Kt}{\log N}}\right).$$

3. Additional tricks: warm-start and epoch structure

Total complexity over all rounds

In round t, coordinate descent for computing $\boldsymbol{Q}_{t+1}$ requires

$$\tilde{O}\left(\sqrt{\frac{Kt}{\log N}}\right) \text{ AMO calls.}$$

Total complexity over all rounds

In round t, coordinate descent for computing $\boldsymbol{Q}_{t+1}$ requires

$$\tilde{O}\left(\sqrt{\frac{Kt}{\log N}}\right) \text{ AMO calls.}$$

To compute $\boldsymbol{Q}_{t+1}$ in all rounds $t = 1, 2, \ldots, T$, need

$$\tilde{O}\left(\sqrt{\frac{K}{\log N}}\,T^{1.5}\right) \text{ AMO calls over } T \text{ rounds.}$$

Warm start

To compute $\boldsymbol{Q}_{t+1}$ using coordinate descent, initialize with $\boldsymbol{Q}_t$.

Warm start

To compute $\boldsymbol{Q}_{t+1}$ using coordinate descent, initialize with $\boldsymbol{Q}_t$.

1. Total epoch-to-epoch increase in potential is $\tilde{O}(\sqrt{T/K})$ over all T rounds (w.h.p.—exploiting i.i.d. assumption).

Warm start

To compute $\boldsymbol{Q}_{t+1}$ using coordinate descent, initialize with $\boldsymbol{Q}_t$.

1. Total epoch-to-epoch increase in potential is $\tilde{O}(\sqrt{T/K})$ over all T rounds (w.h.p.—exploiting i.i.d. assumption).

2. Each coordinate descent step decreases potential by $\Omega\left(\frac{\log N}{K}\right)$.

Warm start

To compute $\boldsymbol{Q}_{t+1}$ using coordinate descent, initialize with $\boldsymbol{Q}_t$.

1. Total epoch-to-epoch increase in potential is $\tilde{O}(\sqrt{T/K})$ over all T rounds (w.h.p.—exploiting i.i.d. assumption).

2. Each coordinate descent step decreases potential by $\Omega\left(\frac{\log N}{K}\right)$.

3. Over all T rounds,

$$\text{total \# calls to } \mathrm{AMO} \;\leq\; \tilde{O}\left(\sqrt{\frac{KT}{\log N}}\right)$$

Warm start

To compute $\boldsymbol{Q}_{t+1}$ using coordinate descent, initialize with $\boldsymbol{Q}_t$.

1. Total epoch-to-epoch increase in potential is $\tilde{O}(\sqrt{T/K})$ over all T rounds (w.h.p.—exploiting i.i.d. assumption).
2. Each coordinate descent step decreases potential by $\Omega\left(\frac{\log N}{K}\right)$.
3. Over all T rounds,

$$\text{total \# calls to AMO} \;\leq\; \tilde{O}\left(\sqrt{\frac{KT}{\log N}}\right)$$

But still need an AMO call to even check if $\boldsymbol{Q}_t$ is feasible!

Epoch trick

Regret analysis: $\boldsymbol{Q}_t$ has low instantaneous per-round regret—this also crucially relies on i.i.d. assumption.

Epoch trick

Regret analysis: $\boldsymbol{Q}_t$ has low instantaneous per-round regret—this also crucially relies on i.i.d. assumption.

$\Longrightarrow$ same $\boldsymbol{Q}_t$ can be used for $O(t)$ more rounds!

Epoch trick

Regret analysis: $\boldsymbol{Q}_t$ has low instantaneous per-round regret—this also crucially relies on i.i.d. assumption.

$\Longrightarrow$ same $\boldsymbol{Q}_t$ can be used for $O(t)$ more rounds!

Epoch trick: split T rounds into epochs, only compute $\boldsymbol{Q}_t$ at start of each epoch.

Epoch trick

Regret analysis: $\boldsymbol{Q}_t$ has low instantaneous per-round regret—this also crucially relies on i.i.d. assumption.

$\Longrightarrow$ same $\boldsymbol{Q}_t$ can be used for $O(t)$ more rounds!

Epoch trick: split T rounds into epochs, only compute $\boldsymbol{Q}_t$ at start of each epoch.

Doubling: only update on rounds $2^1, 2^2, 2^3, 2^4, \ldots$

$\log T$ updates, so $\tilde{O}(\sqrt{KT/\log N})$ AMO calls overall.

Epoch trick

Regret analysis: $\boldsymbol{Q}_t$ has low instantaneous per-round regret—this also crucially relies on i.i.d. assumption.

$\implies$ same $\boldsymbol{Q}_t$ can be used for $O(t)$ more rounds!

Epoch trick: split T rounds into epochs, only compute $\boldsymbol{Q}_t$ at start of each epoch.

Doubling: only update on rounds $2^1, 2^2, 2^3, 2^4, \ldots$

$\log T$ updates, so $\tilde{O}(\sqrt{KT/\log N})$ AMO calls overall.

Squares: only update on rounds $1^2, 2^2, 3^2, 4^2, \ldots$

$\sqrt{T}$ updates, so $\tilde{O}(\sqrt{K/\log N})$ AMO calls per update, on average.

Warm start + epoch trick

Over all T rounds:

- Update policy distribution on rounds $1^2, 2^2, 3^2, 4^2, \dots,$ i.e., total of $\sqrt{T}$ times.
- Total # calls to AMO:

$$\tilde{O}\left(\sqrt{\frac{KT}{\log N}}\right).$$

- # AMO calls per update (on average):

$$\tilde{O}\left(\sqrt{\frac{K}{\log N}}\right).$$

4. Closing remarks and open problems

Recap

Recap

1. New algorithm for general contextual bandits

Recap

1. New algorithm for general contextual bandits
2. Accesses policy class Π only via AMO.

Recap

1. New algorithm for general contextual bandits
2. Accesses policy class Π only via AMO.
3. Defined convex feasibility problem over policy distributions that are good for exploration/exploitation:

$$\text{Regret} \;\leq\; \tilde{O}\left(\sqrt{\frac{K \log N}{T}}\right).$$

   Coordinate descent finds a $\tilde{O}(\sqrt{KT/\log N})$-sparse solution.

Recap

1. New algorithm for general contextual bandits
2. Accesses policy class Π only via AMO.
3. Defined convex feasibility problem over policy distributions that are good for exploration/exploitation:

   $$\text{Regret} \ \le \ \tilde{O}\left(\sqrt{\frac{K \log N}{T}}\right).$$

   Coordinate descent finds a $\tilde{O}(\sqrt{KT / \log N})$-sparse solution.
4. Epoch structure allows for policy distribution to change very infrequently; combine with warm start for computational improvements.

Open problems

Open problems

1. Empirical evaluation.

Open problems

1. Empirical evaluation.
2. Adaptive algorithm that takes advantage of problem easiness.

Open problems

1. Empirical evaluation.
2. Adaptive algorithm that takes advantage of problem easiness.
3. Alternatives to AMO.

Open problems

1. Empirical evaluation.
2. Adaptive algorithm that takes advantage of problem easiness.
3. Alternatives to AMO.

Thanks!

Projections of policy distributions

Given policy distribution $\boldsymbol{Q}$ and context x,

$$\forall a \in \mathcal{A} \,\mathbf{.} \quad Q(a|x) := \sum_{\pi \in \Pi} Q(\pi) \cdot \mathbb{1}\{\pi(x) = a\}$$

(so $\boldsymbol{Q} \mapsto \boldsymbol{Q}(\cdot|x)$ is a linear map).

Projections of policy distributions

Given policy distribution $\boldsymbol{Q}$ and context x,

$$\forall a \in \mathcal{A} \,.\quad Q(a|x) := \sum_{\pi \in \Pi} Q(\pi) \cdot \mathbb{1}\{\pi(x) = a\}$$

(so $\boldsymbol{Q} \mapsto \boldsymbol{Q}(\cdot|x)$ is a linear map).

We actually use

$$\boldsymbol{p}_t := \boldsymbol{Q}_t^{\mu_t}(\,\cdot\,|x_t) := (1 - K\mu_t)\boldsymbol{Q}_t(\,\cdot\,|x_t) + \mu_t \mathbf{1}$$

so every action has probability at least μ_t (*to be determined*).

The potential function

$$\Phi(\boldsymbol{Q}) := t\mu_t\left(\frac{\widehat{\mathbb{E}}_{x\in H_t}\left[\mathrm{RE}(\textbf{uniform}\|\boldsymbol{Q}^{\mu_t}(\cdot|x))\right]}{1-K\mu_t} + \frac{\sum_{\pi\in\Pi} Q(\pi)\widehat{\mathrm{Reg}}_t(\pi)}{Kt\cdot\mu_t}\right),$$