Interactive machine learning via reductions to supervised learning

Daniel Hsu
Columbia University

July 27, 2016

Non-interactive machine learning

"Cartoon" of (non-interactive) machine learning

(a.k.a. **supervised learning**)

Non-interactive machine learning

"Cartoon" of (non-interactive) machine learning

(a.k.a. **supervised learning**)

1. Get labeled data $\{(\text{input}_i, \text{output}_i)\}_{i=1}^{n}$.

Non-interactive machine learning

"Cartoon" of (non-interactive) machine learning

(a.k.a. **supervised learning**)

1. Get labeled data $\{(\mathrm{input}_i, \mathrm{output}_i)\}_{i=1}^n$.
2. **Learn** prediction function $\hat{f}$ (e.g., classifier, regressor, policy) such that

$$\hat{f}(\mathrm{input}_i) \approx \mathrm{output}_i$$

for most $i = 1, 2, \ldots, n$.

Non-interactive machine learning

"Cartoon" of (non-interactive) machine learning

(a.k.a. **supervised learning**)

1. Get labeled data $\{(\mathrm{input}_i, \mathrm{output}_i)\}_{i=1}^n$.
2. **Learn** prediction function $\hat{f}$ (e.g., classifier, regressor, policy) such that

$$\hat{f}(\mathrm{input}_i) \approx \mathrm{output}_i$$

   for most $i = 1, 2, \ldots, n$.

Goal: $\hat{f}(\mathrm{input}) \approx \mathrm{output}$ for future $(\mathrm{input}, \mathrm{output})$ pairs.

Non-interactive machine learning

"Cartoon" of (non-interactive) machine learning

(a.k.a. **supervised learning**)

1. Get labeled data $\{(\mathrm{input}_i, \mathrm{output}_i)\}_{i=1}^n$.
2. **Learn** prediction function $\hat{f}$ (e.g., classifier, regressor, policy) such that

   $$\hat{f}(\mathrm{input}_i) \approx \mathrm{output}_i$$

   for most $i = 1, 2, \dots, n$.

Goal: $\hat{f}(\mathrm{input}) \approx \mathrm{output}$ for future $(\mathrm{input}, \mathrm{output})$ pairs.

Many **applications** supported by **mature technologies** and explained/motivated by **mathematical theories**.

Interactive machine learning: example #1

Practicing physician

Interactive machine learning: example #1

Practicing physician

Loop:

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

Interactive machine learning: example #1

Practicing physician

Loop:

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

Interactive machine learning: 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).

Interactive machine learning: 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.

Interactive machine learning: example #2

Website operator

Interactive machine learning: example #2

Website operator

Loop:

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

Interactive machine learning: example #2

Website operator

Loop:

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

Interactive machine learning: 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”).

Interactive machine learning: 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.

Interactive machine learning: example #3

E-mail service provider

Interactive machine learning: example #3

E-mail service provider

Loop:

1. Receive e-mail messages for users (spam or not).

Interactive machine learning: example #3

E-mail service provider

Loop:

1. Receive e-mail messages for users (spam or not).
2. Ask users to provide labels for some borderline cases.

Interactive machine learning: example #3

E-mail service provider

Loop:

1. Receive e-mail messages for users (spam or not).
2. Ask users to provide labels for some borderline cases.
3. Improve spam filter using newly labeled messages.

Interactive machine learning: example #3

E-mail service provider
Loop:

1. Receive e-mail messages for users (spam or not).
2. Ask users to provide labels for some borderline cases.
3. Improve spam filter using newly labeled messages.

Goal: maximize accuracy of spam filter, minimize queries to users.

Characteristics of interactive machine learning problems

Characteristics of interactive machine learning problems

1. Learning agent (a.k.a. "learner") interacts with the world (e.g., patients, users) to gather data.

Characteristics of interactive machine learning problems

1. Learning agent (a.k.a. "learner") interacts with the world (e.g., patients, users) to gather data.
2. Learner's performance based on learner's decisions.

Characteristics of interactive machine learning problems

1. Learning agent (a.k.a. "learner") interacts with the world (e.g., patients, users) to gather data.
2. Learner's performance based on learner's decisions.
3. Data available to learner depends on learner's decisions.

Characteristics of interactive machine learning problems

1. Learning agent (a.k.a. "learner") interacts with the world (e.g., patients, users) to gather data.
2. Learner's performance based on learner's decisions.
3. Data available to learner depends on learner's decisions.
4. State of the world depends on learner's decisions.

Characteristics of interactive machine learning problems

1. Learning agent (a.k.a. "learner") interacts with the world (e.g., patients, users) to gather data.
2. Learner's performance based on learner's decisions.
3. Data available to learner depends on learner's decisions.
4. ~~State of the world depends on learner's decisions.~~

This talk: ~~two~~ interactive machine learning problems

1. contextual bandit learning ~~(≈ 80% of rest-of-talk)~~
2. ~~active learning (≈ 20% of rest-of-talk)~~

\+ how to solve them using existing methods for non-interactive ML.

(Our models for these problems have all but the last of above characteristics.)

1. Contextual bandit learning

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

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.

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:

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

Strong benchmark when Π has a policy w/ high expected reward.

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

Our result (informally)

New fast and simple algorithm for contextual bandits

(Agarwal, Hsu, Kale, Langford, Li, & Schapire, ICML 2014).

Our result (informally)

New fast and simple algorithm for contextual bandits

(Agarwal, Hsu, Kale, Langford, Li, & Schapire, ICML 2014).

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

Our result (informally)

New fast and simple algorithm for contextual bandits

(Agarwal, Hsu, Kale, Langford, Li, & Schapire, ICML 2014).

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

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 $\pi \in \Pi$.

- **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 all actions . . .

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

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

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

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)

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

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.

Our result (formally)

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

Our result: a new, fast and simple algorithm
(Agarwal, Hsu, Kale, Langford, Li, & Schapire, ICML 2014).

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

Coming up

Components of the new contextual bandits algorithm:

1. "Classical" tricks: randomization, inverse propensity weighting.
2. Efficient algorithm for balancing exploration/exploitation.

2. Classical tricks for contextual bandits

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:

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

Unbiasedness:

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

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:

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

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:

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

Unbiasedness:

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

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

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

Inverse propensity weighting (Horvitz & Thompson, JASA 1952)

Importance-weighted estimate of reward from round t:

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

Unbiasedness:

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

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

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

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

Hedging over policies

Get action distributions via policy distributions.

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

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.

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

```
Pick initial distribution Q_1 over policies Π.
for round t = 1, 2, ... do
    Nature draws (x_t, r_t) from dist. D over X × [0,1]^A.
    Observe context x_t.
    Compute distribution p_t over A (using Q_t and x_t).
    Pick action a_t ~ p_t.
    Collect reward r_t(a_t).
    Compute new distribution Q_{t+1} over policies Π.
end for
```

3. Algorithm for constructing 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 Q

The "good policy distribution" problem

Convex feasibility problem for policy distribution Q

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

The "good policy distribution" problem

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

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

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

The "good policy distribution" problem

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

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

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

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

The "good policy distribution" problem

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

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

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

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

But N variables and $N+1$ 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., $\boldsymbol{Q} := \mathbf{0}$), 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

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

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

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

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

Iteration bound

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

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

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

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

Iteration bound

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

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

(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 $(1 - \varepsilon) \cdot \boldsymbol{Q} + \varepsilon \cdot$ **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.

Our analysis shows that, in round t,

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

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.

Our analysis shows that, in round t,

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

With a few additional tricks:

$$\text{Total } \#\text{ calls to } \mathrm{AMO} \text{ over all } T \text{ rounds} \;=\; \tilde{O}\left(\sqrt{\frac{KT}{\log N}}\right),$$

i.e., only about once every $\sqrt{T}$ rounds.

Summary

Summary

1. New statistically optimal algorithm for contextual bandits.

Summary

1. New statistically optimal algorithm for contextual bandits.
2. Accesses policy class Π only via AMO (i.e., supervised learner).

Summary

1. New statistically optimal algorithm for contextual bandits.
2. Accesses policy class Π only via AMO (i.e., supervised learner).
3. Algorithm uses AMO (sparingly!) to solve convex feasibility problem over policy distributions that balances exploration and exploitation.

4. Wrap-up

Wrap-up

Wrap-up

- Interactive machine learning (e.g., contextual bandits, active learning) confronts challenges in how machine learning is used in real applications.

Wrap-up

- Interactive machine learning (e.g., contextual bandits, active learning) confronts challenges in how machine learning is used in real applications.
- Sampling bias is a pervasive issue:
  direct application of non-interactive ML methods fail.

Wrap-up

- Interactive machine learning (e.g., contextual bandits, active learning) confronts challenges in how machine learning is used in real applications.
- Sampling bias is a pervasive issue:
  direct application of non-interactive ML methods fail.
- **Future directions**:
  - Richer forms of interaction / more powerful queries
  - Interactive algorithms for solving other data analysis tasks (e.g., clustering, error profiling, debugging)

Wrap-up

- Interactive machine learning (e.g., contextual bandits, active learning) confronts challenges in how machine learning is used in real applications.
- Sampling bias is a pervasive issue:
  direct application of non-interactive ML methods fail.
- **Future directions**:
  - Richer forms of interaction / more powerful queries
  - Interactive algorithms for solving other data analysis tasks (e.g., clustering, error profiling, debugging)

Thanks!

References:
Contextual bandits: `http://arxiv.org/abs/1402.0555`
Active learning `http://arxiv.org/abs/1506.08669`