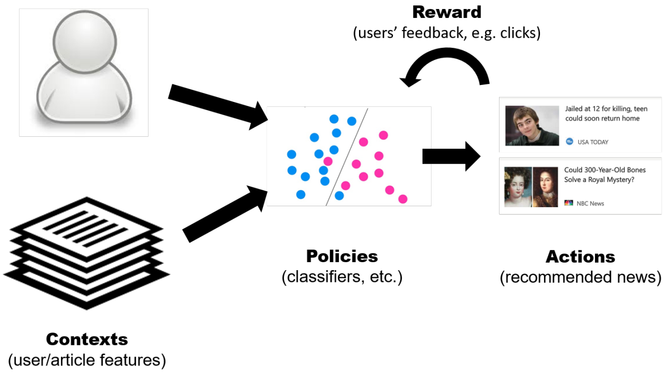
Efficient Contextual Bandits in Non-stationary Worlds

Haipeng Luo[1], **Chen-Yu Wei[1]**, Alekh Agarwal[2], John Langford[2]

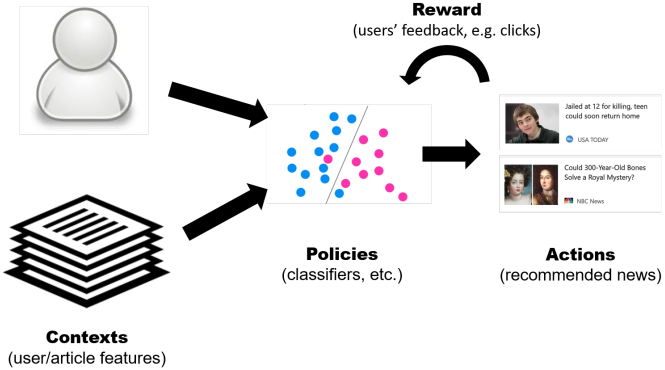
[1]University of Southern California, [2]Microsoft Research (New York City)

Contextual Bandits



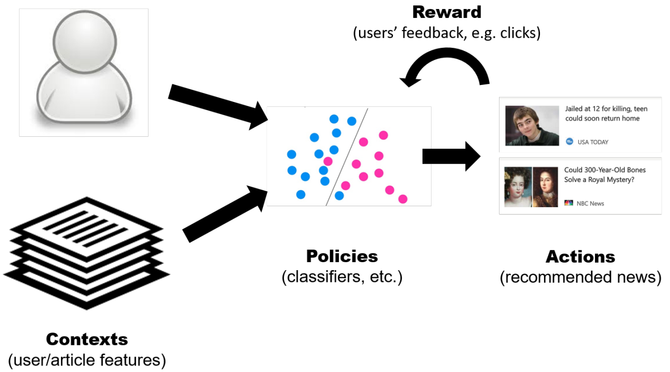
Goal: earn as much reward as possible in the long run.

Contextual Bandits



Goal: earn as much reward as possible in the long run.
Only see whether the recommended news is clicked or not (but not other news)

Contextual Bandits



Goal: earn as much reward as possible in the long run.

Only see whether the recommended news is clicked or not (but not other news)

Different from multi-armed bandit.

Contextual Bandits

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

- see a **context** $x_t \in \mathcal{X}$
- pick an **action** $a_t \in \{1, 2, \ldots, K\}$
- observe **reward** $r_t(a_t) \in [0, 1]$ (but not $r_t(a)$ for $a \neq a_t$)

Contextual Bandits

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

- see a **context** $x_t \in \mathcal{X}$
- pick an **action** $a_t \in \{1, 2, \ldots, K\}$
- observe **reward** $r_t(a_t) \in [0, 1]$ (but not $r_t(a)$ for $a \neq a_t$)

Goal: Given a policy class $\Pi = \{\pi : \mathcal{X} \to [K]\}$ (e.g., neural nets, trees), the goal is to minimize

$$\text{regret} = \sum_{t=1}^{T} r_t(\pi^*(x_t)) - \sum_{t=1}^{T} r_t(a_t), \qquad \pi^* = \text{best policy in } \Pi$$

Contextual Bandits

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

- see a **context** $x_t \in \mathcal{X}$
- pick an **action** $a_t \in \{1, 2, \ldots, K\}$
- observe **reward** $r_t(a_t) \in [0, 1]$ (but not $r_t(a)$ for $a \neq a_t$)

Goal: Given a policy class $\Pi = \{\pi : \mathcal{X} \to [K]\}$ (e.g., neural nets, trees), the goal is to minimize

$$\text{regret} = \sum_{t=1}^{T} r_t(\pi^*(x_t)) - \sum_{t=1}^{T} r_t(a_t), \qquad \pi^* = \text{best policy in } \Pi$$

- EXP4: sublinear regret, but the complexity is linear in $|\Pi|$

Contextual Bandits

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

- see a **context** $x_t \in \mathcal{X}$
- pick an **action** $a_t \in \{1, 2, \ldots, K\}$
- observe **reward** $r_t(a_t) \in [0, 1]$ (but not $r_t(a)$ for $a \neq a_t$)

Goal: Given a policy class $\Pi = \{\pi : \mathcal{X} \to [K]\}$ (e.g., neural nets, trees), the goal is to minimize

$$\text{regret} = \sum_{t=1}^{T} r_t(\pi^*(x_t)) - \sum_{t=1}^{T} r_t(a_t), \qquad \pi^* = \text{best policy in } \Pi$$

- EXP4: sublinear regret, but the complexity is linear in $|\Pi|$
- ϵ-greedy, ILOVETOCONBANDITS, BISTRO+ are **oracle-efficient** (poly($\ln|\Pi|, K, T$) calls), but make i.i.d. assumptions.

Oracle: input: $\{(x_t, r_t)\}_{t=1}^{\tau}$, output: $\arg\max_{\pi\in\Pi} \sum_{t=1}^{\tau} r_t(\pi(x_t))$ (**ERM**)

Motivation

Motivation

- Observation: previous (oracle) efficient contextual bandit algorithms all make i.i.d. assumptions.

Motivation

- Observation: previous (oracle) efficient contextual bandit algorithms all make i.i.d. assumptions.

  But non-i.i.d. is ubiquitous: preference change on a daily/seasonal basis.

Motivation

- Observation: previous (oracle) efficient contextual bandit algorithms all make i.i.d. assumptions.

  But non-i.i.d. is ubiquitous: preference change on a daily/seasonal basis.
- Can we have (oracle) efficient algorithms that handle non-i.i.d. data?

Motivation

- Observation: previous (oracle) efficient contextual bandit algorithms all make i.i.d. assumptions.
  But non-i.i.d. is ubiquitous: preference change on a daily/seasonal basis.
- Can we have (oracle) efficient algorithms that handle non-i.i.d. data?
- **Dynamic regret**:

$$\text{regret} = \sum_{t=1}^{T} r_t(\pi_t^*(x_t)) - \sum_{t=1}^{T} r_t(a_t) \quad \text{assuming } (x_t, r_t) \sim \mathcal{D}_t$$

$\pi_t^* \triangleq \arg\max_{\pi \in \Pi} \mathbb{E}_{(x,r)\sim\mathcal{D}_t}[r_t(\pi(x_t))]$.

Motivation

- Observation: previous (oracle) efficient contextual bandit algorithms all make i.i.d. assumptions.
  But non-i.i.d. is ubiquitous: preference change on a daily/seasonal basis.
- Can we have (oracle) efficient algorithms that handle non-i.i.d. data?
- **Dynamic regret**:

$$\text{regret} = \sum_{t=1}^{T} r_t(\pi_t^*(x_t)) - \sum_{t=1}^{T} r_t(a_t) \quad \text{assuming } (x_t, r_t) \sim \mathcal{D}_t$$

$\pi_t^* \triangleq \arg\max_{\pi \in \Pi} \mathbb{E}_{(x,r)\sim\mathcal{D}_t}[r_t(\pi(x_t))]$.

- Sublinear regret is impossible in general

Motivation

- Observation: previous (oracle) efficient contextual bandit algorithms all make i.i.d. assumptions.
  But non-i.i.d. is ubiquitous: preference change on a daily/seasonal basis.
- Can we have (oracle) efficient algorithms that handle non-i.i.d. data?
- **Dynamic regret**:

$$\text{regret} = \sum_{t=1}^{T} r_t(\pi_t^*(x_t)) - \sum_{t=1}^{T} r_t(a_t) \quad \text{assuming } (x_t, r_t) \sim \mathcal{D}_t$$

$\pi_t^* \triangleq \arg\max_{\pi \in \Pi} \mathbb{E}_{(x,r)\sim\mathcal{D}_t}[r_t(\pi(x_t))]$.

- Sublinear regret is impossible in general
- Previous methods for MAB dynamic regret (e.g., [Besbes et al.'14]) become inefficient for CB

Results

Assumption 1: $\sum_{t=2}^{T} \mathbf{1}\{\mathcal{D}_t \neq \mathcal{D}_{t-1}\} \leq S$
Assumption 2: $\sum_{t=2}^{T} \mathsf{TV}(\mathcal{D}_t, \mathcal{D}_{t-1}) \leq \Delta$

Results

Assumption 1: $\sum_{t=2}^{T} \mathbf{1}\{\mathcal{D}_t \neq \mathcal{D}_{t-1}\} \leq S$
Assumption 2: $\sum_{t=2}^{T} \mathsf{TV}(\mathcal{D}_t, \mathcal{D}_{t-1}) \leq \Delta$

Regret bounds under Assumption 2:

Results

Assumption 1: $\sum_{t=2}^{T} \mathbf{1}\{\mathcal{D}_t \neq \mathcal{D}_{t-1}\} \leq S$
Assumption 2: $\sum_{t=2}^{T} \mathsf{TV}(\mathcal{D}_t, \mathcal{D}_{t-1}) \leq \Delta$

Regret bounds under Assumption 2:

Algorithm	regret

Results

Assumption 1: $\sum_{t=2}^{T} \mathbf{1}\{\mathcal{D}_t \neq \mathcal{D}_{t-1}\} \leq S$
Assumption 2: $\sum_{t=2}^{T} \mathsf{TV}(\mathcal{D}_t, \mathcal{D}_{t-1}) \leq \Delta$

Regret bounds under Assumption 2:

Algorithm	regret
Ada-Greedy	$\Delta^{1/4} T^{3/4}$

Results

Assumption 1: $\sum_{t=2}^{T} \mathbf{1}\{\mathcal{D}_t \neq \mathcal{D}_{t-1}\} \leq S$
Assumption 2: $\sum_{t=2}^{T} \mathsf{TV}(\mathcal{D}_t, \mathcal{D}_{t-1}) \leq \Delta$

Regret bounds under Assumption 2:

Algorithm	regret
Ada-Greedy	$\Delta^{1/4} T^{3/4}$
Ada-ILTCB	$\Delta^{1/3} T^{2/3}$ (optimal)

Results

Assumption 1: $\sum_{t=2}^{T} \mathbf{1}\{\mathcal{D}_t \neq \mathcal{D}_{t-1}\} \leq S$
Assumption 2: $\sum_{t=2}^{T} \mathrm{TV}(\mathcal{D}_t, \mathcal{D}_{t-1}) \leq \Delta$

Regret bounds under Assumption 2:

Algorithm	regret	(if Δ unknown)
Ada-Greedy	$\Delta^{1/4} T^{3/4}$	$\rightarrow \sqrt{\Delta} T^{3/4}$
Ada-ILTCB	$\Delta^{1/3} T^{2/3}$ (optimal)	$\rightarrow \Delta T^{2/3}$

Results

Assumption 1: $\sum_{t=2}^{T} \mathbf{1}\{\mathcal{D}_t \neq \mathcal{D}_{t-1}\} \leq S$
Assumption 2: $\sum_{t=2}^{T} \mathrm{TV}(\mathcal{D}_t, \mathcal{D}_{t-1}) \leq \Delta$

Regret bounds under Assumption 2:

Algorithm	regret	(if Δ unknown)
Ada-Greedy	$\Delta^{1/4} T^{3/4}$	$\to \sqrt{\Delta} T^{3/4}$
Ada-ILTCB	$\Delta^{1/3} T^{2/3}$ (optimal)	$\to \Delta T^{2/3}$
Ada-BinGreedy (parameter-free)	$\Delta^{1/5} T^{4/5}$	

Results

Assumption 1: $\sum_{t=2}^{T} \mathbf{1}\{\mathcal{D}_t \neq \mathcal{D}_{t-1}\} \leq S$
Assumption 2: $\sum_{t=2}^{T} \mathsf{TV}(\mathcal{D}_t, \mathcal{D}_{t-1}) \leq \Delta$

Regret bounds under Assumption 2:

Algorithm	regret	(if Δ unknown)
Ada-Greedy	$\Delta^{1/4} T^{3/4}$	$\rightarrow \sqrt{\Delta} T^{3/4}$
Ada-ILTCB	$\Delta^{1/3} T^{2/3}$ (optimal)	$\rightarrow \Delta T^{2/3}$
Ada-BinGreedy (parameter-free)	$\Delta^{1/5} T^{4/5}$	

- Providing a solution to the open problem in [Besbes et al.'14]

Results

Assumption 1: $\sum_{t=2}^{T} \mathbf{1}\{\mathcal{D}_t \neq \mathcal{D}_{t-1}\} \leq S$
Assumption 2: $\sum_{t=2}^{T} \text{TV}(\mathcal{D}_t, \mathcal{D}_{t-1}) \leq \Delta$

Regret bounds under Assumption 2:

Algorithm	regret	(if Δ unknown)
Ada-Greedy	$\Delta^{1/4} T^{3/4}$	$\rightarrow \sqrt{\Delta} T^{3/4}$
Ada-ILTCB	$\Delta^{1/3} T^{2/3}$ (optimal)	$\rightarrow \Delta T^{2/3}$
Ada-BinGreedy (parameter-free)	$\Delta^{1/5} T^{4/5}$	

- Providing a solution to the open problem in [Besbes et al.'14]
- Improving and generalizing the result of [Karnin&Anava'16]
  $\Delta^{0.18} T^{0.82}$ in 2-armed bandit $\rightarrow \min\{S^{1/4} T^{3/4}, \Delta^{1/5} T^{4/5}\}$ in CB

Algorithm: Ada-Greedy

ϵ-**greedy**[Langford&Zhang'08]:
For $t = 1, 2, \ldots, T$:

- with probability ϵ, uniformly explore
- with probability $1 - \epsilon$, follow $\arg\max_{\pi \in \Pi} \sum_{\tau=1}^{t-1} \hat{r}_\tau(\pi(x_\tau))$
- construct importance-weighted estimator $\hat{r}_t$

Algorithm: Ada-Greedy

Ada-greedy:

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

- with probability ϵ, uniformly explore
- with probability $1 - \epsilon$, follow $\arg\max_{\pi \in \Pi} \sum_{\tau=1}^{t-1} \hat{r}_\tau(\pi(x_\tau))$
- construct importance-weighted estimator $\hat{r}_t$
- restart the algorithm if non-stationarity is detected

Algorithm: Ada-Greedy

Ada-greedy:
For $t = 1, 2, \ldots, T$:

- with probability ϵ, uniformly explore
- with probability $1 - \epsilon$, follow $\arg\max_{\pi \in \Pi} \sum_{\tau=1}^{t-1} \hat{r}_\tau(\pi(x_\tau))$
- construct importance-weighted estimator $\hat{r}_t$
- restart the algorithm if non-stationarity is detected

Stationarity check: For $\mathcal{I} = [t, t-2], [t, t-4], [t, t-8], \ldots$, check if $\frac{1}{t} \sum_{\tau=1}^{t} \hat{r}_\tau(\pi(x_\tau))$ and $\frac{1}{|\mathcal{I}|} \sum_{\tau \in \mathcal{I}} \hat{r}_\tau(\pi(x_\tau))$ are **consistent** for all π (can achieve this with oracle calls)

Algorithm: Ada-Greedy

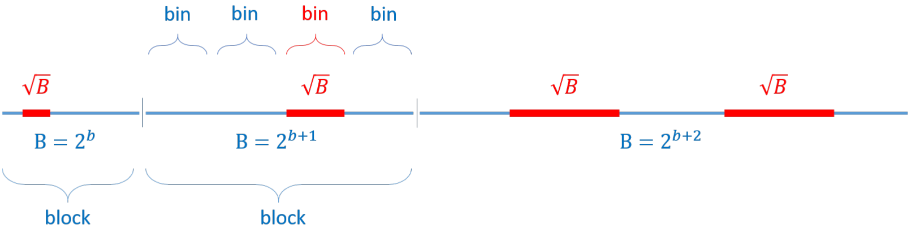
Ada-greedy:
For $t = 1, 2, \ldots, T$:

- with probability ϵ, uniformly explore
- with probability $1 - \epsilon$, follow $\arg\max_{\pi \in \Pi} \sum_{\tau=1}^{t-1} \hat{r}_\tau(\pi(x_\tau))$
- construct importance-weighted estimator $\hat{r}_t$
- restart the algorithm if non-stationarity is detected

Stationarity check: For $\mathcal{I} = [t, t-2], [t, t-4], [t, t-8], \ldots$, check if $\frac{1}{t} \sum_{\tau=1}^{t} \hat{r}_\tau(\pi(x_\tau))$ and $\frac{1}{|\mathcal{I}|} \sum_{\tau \in \mathcal{I}} \hat{r}_\tau(\pi(x_\tau))$ are **consistent** for all π (can achieve this with oracle calls)

- Our Ada-ILTCB algorithm is a from a similar adaptation from ILOVETOCONBANDITS[Agarwal et al.'14], which leads to optimal regret bounds.

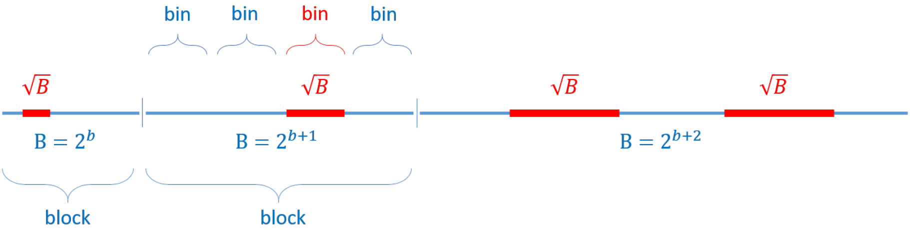
Algorithm: Ada-BinGreedy (Parameter-free)



For $b = 1, 2, \ldots$

- $B \leftarrow 2^b$ (block length)
- partition the next B rounds into $\sqrt{B}$ **bins**, each with length $\sqrt{B}$
- For each bin
  - with probability $B^{-1/4}$ do pure exploration (over actions)
  - otherwise do ϵ-greedy with $\epsilon \approx t^{-1/3}$
- restart the algorithm if non-stationarity is detected

Algorithm: Ada-BinGreedy (Parameter-free)



For $b = 1, 2, \ldots$

- $B \leftarrow 2^b$ (block length)
- partition the next B rounds into $\sqrt{B}$ **bins**, each with length $\sqrt{B}$
- For each bin
  - with probability $B^{-1/4}$ do pure exploration (over actions)
  - otherwise do ϵ-greedy with $\epsilon \approx t^{-1/3}$
- restart the algorithm if non-stationarity is detected

Gives $\min\{S^{1/4}T^{3/4}, \Delta^{1/5}T^{4/5}\}$ regret bound without knowing S or Δ.