Bypassing the Monster: A Faster and Simpler Optimal Algorithms for Contextual Bandits under Realizability

Yunzong Xu
MIT

Joint work with David Simchi-Levi
Outline

• Motivation and Research Question
• Technical Hurdles and Our Contributions
• The Algorithm
• The Analysis
• Extensions
Contextual Bandits

• For round $t = 1, \cdots, T$
  • Nature generates a random context $x_t \in \mathcal{X}$ according to a fixed unknown distribution
  • Learner observes $x_t$ and makes a decision $a_t \in \{1, \ldots, K\}$
  • Nature generates a random reward $r_t(a_t) \in [0,1]$ according to an unknown distribution with conditional mean
    \[
    \mathbb{E}[r_t(a_t)|x_t = x, a_t = a] = f^*(x,a)
    \]

• We call $f^*$ the ground-truth reward function
• In statistical learning, people use a function class $\mathcal{F}$ to approximate $f^*$. Examples of $\mathcal{F}$:
  • Linear class / high-dimension linear class / generalized linear models
  • Non-parametric class / reproducing kernel Hilbert space
  • Regression trees
  • Neural networks

• Regret: $\mathbb{E}\left[\sum_{t \in [T]} r_t(\pi^*(x_t)) - \sum_{t \in [T]} r_t(a_t)\right]$, where $\pi^*(x) = \arg \max_{a \in [K]} f^*(x,a)$
Why is the problem important and interesting?

- Contextual bandits capture two essential features of sequential decision making under uncertainty
  - **Bandit feedback**: for each context $x_t$, learner only observes the reward for her chosen action $a_t$; no other rewards are observed
    - Learner faces a trade-off between exploration and exploitation
  - **Heterogeneity**: the effectiveness of each action depends on the context
    - The context space is huge --- learner has to learn across contexts
      - This incorporates the challenge of generalization in statistical learning theory

- The exploration-exploitation trade-off in contextual bandits is more challenging than in MAB
Literature

• Algorithms:
  • **Upper Confidence Bounds** (Filippi et al. 2010, Rigollet and Zeevi 2010, Abbasi-Yadkori et al. 2011, Chu et al. 2011, Li et al. 2017, ...)
  • **Thompson Samplings** (Agrawal and Goyal 2013, Russo et al. 2018, ...)
  • **Exponential Weighting** (Auer et al. 2002, McMahan and Streeter 2009, Beygelzimer et al. 2011, ...)

• Applications:
  • **Recommender systems** (Li et al. 2010, Agarwal et al. 2016, ...)
  • **Ride-hailing platforms** (Chen et al. 2019, ...)
  • **Healthcare** (Tewari and Murphy 2017, Bastani and Bayati 2020, ...)
Examples of Applications

• Product recommendation:
  • $K$ products
  • $T$ customers arriving in a sequential manner. Each customer has a feature $x_t$ describing gender, age, shopping history, device type, etc.
  • The task is to recommend a product $a_t$ (based on $x_t$) that generates revenue as high as possible
  • The revenue distribution is unknown, with its conditional mean $f^*(x_t, a_t)$ to be learned

• Personalized medicine:
  • $K$ treatments / dose levels
  • $T$ patients arriving in a sequential manner. Each patient has a feature $x_t$ describing her demographics, diagnosis, genes, etc.
  • The task is to pick a personalized treatment (or dose level) $a_t$ (based on $x_t$) that is as effective as possible
  • The efficacy is random and unknown, with the efficacy rate $f^*(x_t, a_t)$ to be learned
The Challenges

• We are interested in contextual bandits with a general function class $F$
• Realizability assumption:
  \[ f^* \in F \]

• **Statistical challenge**: How to achieve the optimal regret for any general function class $F$?
• **Computational challenge**: How to make the algorithm computational efficient?

• Classical contextual bandits approaches fail to simultaneously address the above two challenges in practice, as they typically
  • Become **statistically suboptimal** for general $F$ (e.g., UCB variants and Thompson Sampling)
  • Become **computationally intractable** for large $F$ (e.g., Exponential weighting)
Research Question

• Observation: The statistical and computational aspects of “offline regression with a general $F$” are very well-studied in ML. Given i.i.d. offline data, advances in ML (e.g., kernel methods, boosting, random forests, deep learning) enable us to find a predictor $\hat{f}$ such that
  • (statistically) $\hat{f}$ achieves low estimation error
  • (computationally) $\hat{f}$ can be efficiently computed

• Can we reduce general contextual bandits to general offline regression?

• Specifically, for any $F$, given an offline regression oracle, e.g., a least-squares regression oracle

$$\arg\min_{f \in F} \sum_{t=1}^{n} (f(x_t, a_t) - r_t(a_t))^2$$

or its regularized counterparts (e.g., Ridge and Lasso), can we design a contextual bandit algorithm such that
  • (statistically) it achieves the optimal regret whenever the offline regression oracle attains the optimal estimation error
  • (computationally) it requires no more computation than calling the offline regression oracle

• An open problem mentioned in Agarwal et al. (2012), Foster et al. (2018), Foster and Rakhlin (2020)
Outline

• Motivation and Research Question
• Technical Hurdles and Our Contributions
• The Algorithm
• The Analysis
• Extensions
Why is reducing contextual bandits to offline regression difficult?

• Statistical hurdles associated with confidence bounds
  • UCB and successive elimination only have theoretical guarantees when the confidence bounds for each context is shrinking
  • This property does not hold for a general function class $F$

• Statistical hurdles associated with dependent actions
  • Translating offline estimation error guarantees to contextual bandits is a challenge
  • This is because the data collected in the learning process is not i.i.d
    • The action distributions in later rounds depend on the action distributions in previous rounds
  • Foster and Rakhlin (2020) provide an optimal algorithm for contextual bandits, assuming access to an online regression oracle
  • Online regression oracles provide statistical guarantees for an arbitrary data sequence possibly generated by an adaptive adversary
    • This is a much stronger oracle compared with the offline oracle that we assume
    • Optimal and efficient algorithms for online regression is only known for specific $F$

Yunzong Xu (MIT)
Our Contributions

• We provide the first optimal and efficient **black-box reduction** from general contextual bandits to offline regression
  • The algorithm is simpler and faster than existing approaches to general contextual bandits
    • Our algorithm builds on an intriguing sampling rule developed by Abe and Long (1999) and Foster and Rakhlin (2020). We combine this rule with an offline (rather than online) regression oracle, and add a few useful ingredients to the algorithm
    • The design of the algorithm is also motivated by Agarwal et al. (2014)
    • The analysis of the algorithm is highly non-trivial and reveals connections between different historical approaches to contextual bandits
  • Any advances in offline regression immediately translate to contextual bandits, statistically and computationally

Yunzong Xu (MIT)
Our Contributions

• Our algorithm’s computational complexity can be much better than existing algorithms for complicated $F$

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Regret rate</th>
<th>Computational complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regressor Elimination (Agarwal et al. 2012)</td>
<td>optimal</td>
<td>$\Omega(</td>
</tr>
<tr>
<td>ILOVETOCONBANDITS (Agarwal et al. 2014)</td>
<td>optimal</td>
<td>$O(\sqrt{KT/\log</td>
</tr>
<tr>
<td>RegCB (Foster et al. 2018)</td>
<td>suboptimal</td>
<td>$O(T^{3/2})$ calls to an offline least squares oracle</td>
</tr>
<tr>
<td>SquareCB (Foster and Rakhlin 2020)</td>
<td>optimal</td>
<td>$O(T)$ calls to an online regression oracle</td>
</tr>
<tr>
<td>FALCON / FALCON+ (this paper)</td>
<td>optimal</td>
<td>$O(\log T)$ or $O(\log \log T)$ calls to an offline regression oracle*</td>
</tr>
</tbody>
</table>

* Not restricted to least squares; strictly easier to solve than online regression. See §3 for details.
What Does “Monster” Refer To?

  • These authors refer to their paper as the “Monster Paper”
  • Optimal regret but requires “a monster amount of computation”

  • Optimal regret with reduced computational cost
  • Requires using offline classification oracle

• This paper: Bypassing the Monster
  • Under a weak “realizability” assumption: \( f^* \in F \)
Outline

• Motivation and Research Question
• Technical Hurdles and Our Contributions
• The Algorithm
• The Analysis
• Extensions
The Algorithm

• Three components
  • An epoch schedule (to exponentially save computation)
  • Greedy calls to the offline regression oracle (to obtain reward predictor)
  • A sampling rule ("inverse gap weighting") determined by the predictor and an epoch-varying learning rate (to make decisions)
    • The sampling rule was introduced in Abe and Long (1999) and adopted in Foster and Rakhlin (2020)

• The algorithm is fast and we call it FALCON (FAst Least-squares-regression-oracle CONtextual bandits)

Falcon, the fastest animal on earth
Source: Kirstin Fawcett, fakuto.com

Yunzong Xu (MIT)
Component 1: Epoch Schedule

• An epoch schedule $\tau_1, \tau_2, \ldots$

• Take $\tau_m = 2^m$ as an example, then there are $O(\log T)$ epochs over $T$ rounds

• The purpose of the epoch schedule is to save computation
  • The algorithm only calls the regression oracle at the start of each epoch. When $\tau_m = 2^m$, it only makes $O(\log T)$ calls to the oracle over $T$ rounds
  • The oracle is called more and more infrequently as the algorithm proceeds
Component 2: Oracle Calls

• Before the start of each epoch $m$, solves

$$\arg\min_{f \in F} \sum_{t=1}^{t_{m-1}} (f(x_t, a_t) - r_t(a_t))^2$$

via the least square oracle, and obtains a predictor $\hat{f}_m$

• We can replace the least squares oracle by any other offline regression oracles (e.g., Ridge, Lasso, log loss oracle, ...)

• What to do next for making decisions?

• If we directly follow the predictor to choose greedy actions, then the algorithm does not explore at all, and may perform poor
  • The exploration-exploitation dilemma can be addressed by inverse gap weighting
Component 3: Sampling Rule

• For each epoch $m$, we have a learning rate $\gamma_m$.

• At round $t$, we do the following:

  Observe context $x_t \in \mathcal{X}$.
  Compute $\hat{f}_m(x_t, a)$ for each action $a \in \mathcal{A}$. Let $\hat{a}_t = \max_{a \in \mathcal{A}} \hat{f}_m(x_t, a)$. Define
  \[
  p_t(a) = \begin{cases} 
  \frac{1}{K + \gamma_m (\hat{f}_m(x_t, \hat{a}_t) - \hat{f}_m(x_t, a))}, & \text{for all } a \neq \hat{a}_t, \\
  1 - \sum_{a \neq \hat{a}_t} p_t(a), & \text{for } a = \hat{a}_t.
  \end{cases}
  \]
  Sample $a_t \sim p_t(\cdot)$ and observe reward $r_t(a_t)$.

  The greedy action is the highest. This corresponds to “exploitation”.

  The probability of selecting each non-greedy action is inversely proportional to the predicted gap between this action and the greedy action, as well as the learning rate $\gamma_m$. This corresponds to “exploration”.

  The learning rate balances between exploration and exploitation. The algorithm explores more at the beginning and gradually exploits more.

  \[
  \gamma_m = \sqrt{\tau_{m-1}} \cdot c(F)
  \]
Outline

• Motivation and Research Question
• Technical Hurdles and Our Contributions
• The Algorithm
• The Analysis
• Extensions
General Theoretical Guarantees (informal)

• **Statistically**, we show that the algorithm can always convert the oracle’s offline estimation guarantee to the best-possible online regret guarantee. This means that for any function class $F$, as long as we can find an optimal offline estimator, we are able to achieve the optimal regret.

• **Computationally**, the algorithm only calls the offline oracle for $O(\log T)$ times over $T$ periods if $T$ is unknown. The number of calls can be reduced to $O(\log\log T)$ if $T$ is known. This means that the online problem is computationally “no harder” than an offline problem.
Theorem

- **Offline guarantee**: Given $n$ i.i.d. samples $(x_i, a_i; r_i) \sim D$, an offline regression oracle returns an estimator $\hat{f}$. The estimation error guarantee ensures that $\forall D$,

$$ \mathbb{E}_{(x,a;r) \sim D} \left[ \left( \hat{f}(x,a) - f^*(x,a) \right)^2 \right] \leq Er(n; F) $$

where $Er(n; F)$ depends on the number of samples $n$ and the complexity of $F$.

- **Theorem**: Given an offline regression oracle with estimation error $Er(n; F)$ for $n$ samples, FALCON guarantees regret of

$$ \tilde{O} \left( \sqrt{K Er(T; F) T} \right) $$

through $O(\log T)$ calls to the offline regression oracle.
  - Plugging in the rate-optimal $Er(n; F)$ ensures that the regret is optimal in terms of $T$, which matches the regret lower bound proved in Foster and Rakhlin (2020).
  - The number of oracle calls can be reduced to $O(\log \log T)$ if $T$ is known in advance.
Proof Sketch

• Translating offline estimation error guarantees to contextual bandits is a challenge
• This is because the data collected in the learning process is **not i.i.d**
  - Offline guarantees provide upper bounds on the “distance” between \( \hat{f} \) and \( f^* \) for a fixed action distribution
  - But in contextual bandits, the action distribution is changing, and the action distribution in later rounds **depend on** the action distributions in previous rounds
• We need to find a way to measure the effectiveness of \( \hat{f} \) irrespective of the action distributions, and incorporate it into the exploration-exploitation analysis

• **Step 0.** A **dual interpretation:** our algorithm adaptively maintains a distribution over policies in the **universal policy space** \( \Psi = [K]^X \)
  - A policy \( \pi: X \mapsto [K] \) is a deterministic decision function
    - Let \( \pi_f \) be the true optimal policy, and \( \pi_{f_m} \) be the greedy policy
  - At epoch \( m \), \( \hat{f}_m \) and \( \gamma_m \) induce a distribution over policies \( Q_m(\cdot) \)
    - \( Q_m(\pi) = \prod_{x \in X} p_m(\pi(x)|x) \), where \( p_m(\pi(x)|x) \) is the probability that the sampling rule selects action \( \pi(x) \) given context \( x \)

Yunzong Xu (MIT)
Proof Sketch

Step 1. Per-round property: At each epoch $m$, given $\hat{f}_m$ and $\gamma_m$, the distribution $Q_m$ ensures that

$$\sum_{\pi \in \Psi} Q_m(\pi) \mathbb{E}_X \left[ \hat{f}_m \left(x, \pi_{\hat{f}_m} (x) \right) - \hat{f}_m \left(x, \pi(x) \right) \right] = O(K/\gamma_m)$$

Estimated per-round expected regret of $\pi$

Step 2. Proof by induction: At each epoch $m$, $Q_1, \ldots, Q_{m-1}$ ensure that for all $\pi \in \Psi$,

$$\mathbb{E}_X \left[ f^* \left(x, \pi_{f^*}(x) \right) - f^* \left(x, \pi(x) \right) \right] \leq 2 \mathbb{E}_X \left[ \hat{f}_m \left(x, \pi_{\hat{f}_m} (x) \right) - \hat{f}_m \left(x, \pi(x) \right) \right] + O(K/\gamma_m)$$

True per-round expected regret of $\pi$

Step 3. Putting together: At each epoch $m$, our per-round expected regret is

$$\sum_{\pi \in \Psi} Q_m(\pi) \mathbb{E}_X \left[ f^* \left(x, \pi_{f^*}(x) \right) - f^* \left(x, \pi(x) \right) \right] \leq 2 \sum_{\pi \in \Psi} Q_m(\pi) \mathbb{E}_X \left[ \hat{f}_m \left(x, \pi_{\hat{f}_m} (x) \right) - \hat{f}_m \left(x, \pi(x) \right) \right] + O(K/\gamma_m) = O(K/\gamma_m)$$

by step 1

We choose $\{\gamma_m\}$ such that Step 2 holds and Step 3 leads to the optimal accumulated regret

Yunzong Xu (MIT)
A closer look at Step 2

Step 2. **Proof by induction:** At each epoch $m$, $Q_1, ..., Q_{m-1}$ ensure that for all $\pi \in \Psi$,

$$\mathbb{E}_{D_X} \left[ f^* \left( x, \pi f^*(x) \right) - f^*(x, \pi(x)) \right] \leq 2 \mathbb{E}_{D_X} \left[ \hat{f}_m \left( x, \pi \hat{f}_m(x) \right) - \hat{f}_m \left( x, \pi(x) \right) \right] + O(K/\gamma_m)$$

**True per-round expected regret of $\pi$**

**Estimated per-round expected regret of $\pi$**

- It connects $\hat{f}_m$ and $f^*$ without specifying an action distribution
  - It connects the “estimated world” and the “true world”
- It holds for non-iid and dependent decision process (of $\{a_t\}$)
  - The induction argument shows how exploration in early rounds benefit exploitation in later rounds
- It utilizes the iid properties of $\{x_t\}$
- It establishes a bridge from offline estimation guarantees to online decision making guarantees
  - The analysis is general and does not rely on any refined property of $F$

Yunzong Xu (MIT)
Outline

- Motivation and Research Question
- Technical Hurdles and Our Contributions
- The Algorithm
- The Analysis
- Extensions
Extensions

• Sharper instance-dependent guarantees for contextual bandits, and extensions to reinforcement learning
  • Foster, Rakhlin, Simchi-Levi, and Xu (2020)

• We develop an algorithm called AdaCB, which follows the same general template as FALCON, but with two key differences.
  • First, rather than applying the sampling strategy to all actions, we restrict only to actions which have not been eliminated by our confidence bounds.
  • Second, we choose the learning rate $\gamma_m$ in a more data-driven fashion

• The algorithm always guarantees the minimax regret (when the problem instance does not have good structure), but adapts to superior instance-dependent regret whenever possible
References

• Bypassing the Monster: A Faster and Simpler Optimal Algorithm for Contextual Bandits under Realizability (with D. Simchi-Levi)
  • arXiv 2003.12699

• Instance-Dependent Complexity of Contextual Bandits and Reinforcement Learning: A Disagreement-based Perspective (with D. Foster, A. Rakhlin, and D. Simchi-Levi)
  • arXiv 2010.03104