Contextual Bandits

Sikander Randhawa

Machine Learning Reading Group

July 17, 2019

Recap: Experts problem

- K experts; sequence of rewards $r_1, r_2, \ldots$ with $r_i \in [0,1]^K$.
- Each day, you must choose a distribution over experts, p_t.
- At the end of the day, you see the quality of *every* expert.
- Want performance comparable to the single best expert after T days.

$$Regret(T) := \max_{i \in [K]} \sum_{t=1}^{T} R_{t,i} - \sum_{t=1}^{T} < p_t, R_t >$$

- **Application:** Portfolio management.
- **Main algorithm:** Multiplicative weights update.
- **Standard Regret Bound:** $\mathcal{O}\left(\sqrt{T \log K}\right)$.

Recap: Adversarial Bandits

- K slot machines; sequence of rewards $R_1, R_2, \ldots$ with $R_i \in [0,1]^K$.
- At every time step, select a probability distribution over slot machines, p_t.
    - See only reward of chosen slot machine A_t, R_{t,A_t}.
- Same performance benchmark as experts setting, but harder to achieve.
    - We learn less about machines at each step.

$$Regret(T) := \max_{i \in [K]} \sum_{t=1}^{T} R_{t,i} - \sum_{t=1}^{T} < p_t, R_t >$$

- **Application:** Clinical trials.
- **Main algorithm:** Exp3.
- **Standard Regret Bound:** $\mathcal{O}\left(\sqrt{TK \log K}\right)$.

MWU: Standard algorithm for experts problem

K experts, rewards in [0,1]; step size η.

```
Initialize:
• $p_1 \leftarrow \left(\frac{1}{k}, \frac{1}{k}, \ldots, \frac{1}{k}\right)$ (Uniform distribution over experts)
• $R_0 \leftarrow (0, 0, \ldots, 0)$ (Cumulative rewards)
For $t = 1, \ldots, T$, do:
• Follow expert $j$ with probability $p_{t,j}$.
• Expected reward is $\langle p_t, r_t \rangle$.
• Observe $r_{t,j}$ for each expert $j$.
• Update:
    • $R_{t,j} \leftarrow R_{t-1,j} + r_{t,j}$ for every $j$.
    • $p_{t,j} \leftarrow \exp(\eta R_{t,j}) / Z_t$, $(Z_t = \sum_i \exp(\eta R_{t,i}))$
```

Exp3: Standard algorithm for bandits problem

K "arms", rewards in [0,1]; step size η.

```
Initialize:
• $p_1 \leftarrow \left(\frac{1}{k}, \frac{1}{k}, \dots, \frac{1}{k}\right)$ (Uniform distribution over experts)
• $R_0 \leftarrow (0, 0, \dots, 0)$    (Cumulative rewards)
For $t = 1, \dots, T$, do:
• Select arm $j$ with probability $p_{t,j}$.
• Expected reward is $\langle p_t, r_t \rangle$.
• Observe only $r_{t,j}$ for the chosen arm $j$.
• $\widetilde{r_{t,j}} = \frac{r_{t,j}}{p_{t,j}}$ if chosen expert was arm $j$, 0 otherwise.
• Update:
    • $R_{t,j} \leftarrow R_{t-1,j} + \widetilde{r_{t,j}}$ for every $j$.
    • $p_{t,j} \leftarrow \exp(\eta R_{t,j}) / Z_t$, $(Z_t = \sum_i \exp(\eta R_{t,i}))$
```

Exp3: Standard algorithm for bandits problem

K "arms", rewards in [0,1]; step size η.

```
Initialize:
• $p_1 \leftarrow \left(\frac{1}{k}, \frac{1}{k}, \ldots, \frac{1}{k}\right)$ (Uniform distribution over experts)
• $R_0 \leftarrow (0, 0, \ldots, 0)$ (Cumulative rewards)
For $t = 1, \ldots, T$, do:
• Select arm $j$ with probability $p_{t,j}$.
• Expected reward is $\langle p_t, r_t \rangle$.
• Observe only $r_{t,j}$ for the chosen expert $j$.
• $\widetilde{r_{t,j}} = \frac{r_{t,j}}{p_{t,j}}$ if chosen expert was expert $j$, 0 otherwise.
• Update:
    • $R_{t,j} \leftarrow R_{t-1,j} + \widetilde{r_{t,j}}$ for every $j$.
    • $p_{t,j} \leftarrow \exp(\eta R_{t,j}) / Z_t$, $(Z_t = \sum_i \exp(\eta R_{t,i}))$
```

Note: $\mathbb{E}[\widetilde{r_t}] = r_t$

Recap: Stochastic Bandits

- K slot machines; sequence of rewards $R_1, R_2, \ldots$ with $R_i \in [0,1]^K$, $R_t \sim \mathcal{D}$.
- At every time step, select a probability distribution over slot machines, p_t.
  - See only reward of chosen slot machine A_t, R_{t,A_t}.
- Same performance benchmark as experts setting, but harder to achieve.
  - We learn less about machines at each step.

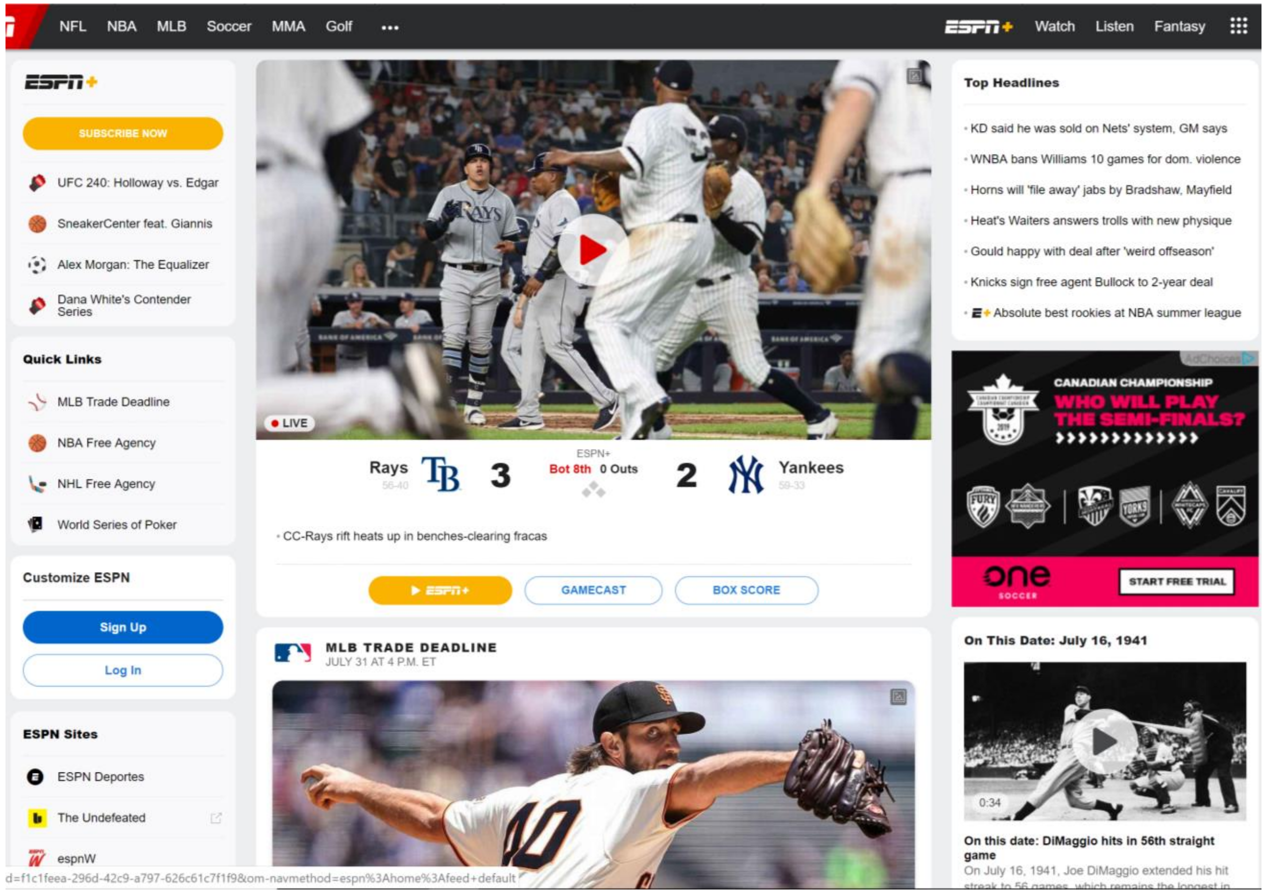
- Slot machine i has mean μ_i.

$$Regret(T) := T\mu^* - \mathbb{E}\left[\sum_{t=1}^{T} R_{t,A_t}\right]$$

- **Application:** Clinical trials.
- **Main algorithm:** UCB, ϵ-greedy, ETC.
- **Standard Regret Bounds:** $o(T)$.

Today: Contextual Bandits

- Usually, there is some context that can help you make a decision.
- For example:
  - Patient data for clinical trials.
  - Consumer data for news/movie recommendation.
- Today, we will see how we can use this "context" to guide our decision making process.



UFC 240: Holloway vs. Edgar
SneakerCenter feat. Giannis
Alex Morgan: The Equalizer
Dana White's Contender Series
MLB Trade Deadline
NBA Free Agency
NHL Free Agency
World Series of Poker
ESPN Deportes
The Undefeated
espnW

Rays 3 — Bot 8th 0 Outs — 2 Yankees

CC-Rays rift heats up in benches-clearing fracas

MLB TRADE DEADLINE
JULY 31 AT 4 P.M. ET

Top Headlines
- KD said he was sold on Nets' system, GM says
- WNBA bans Williams 10 games for dom. violence
- Horns will 'file away' jabs by Bradshaw, Mayfield
- Heat's Waiters answers trolls with new physique
- Gould happy with deal after 'weird offseason'
- Knicks sign free agent Bullock to 2-year deal
- Absolute best rookies at NBA summer league

CANADIAN CHAMPIONSHIP
WHO WILL PLAY THE SEMI-FINALS?

On This Date: July 16, 1941
On this date: DiMaggio hits in 56th straight game
On July 16, 1941, Joe DiMaggio extended his hit streak to 56 games, which remains the longest in

d=f1c1feea-296d-42c9-a797-626c61c7f1f9&om-navmethod=espn%3Ahome%3Afeed+default

News that generates most clicks should be front and center.
However, the "click through rate" depends on the user!

Contextual Bandits: Formal Setup

- K arms, with reward sequence $r, r_2, \ldots$, with $r_i \in [0,1]^K$.
- At time t, algorithm receives context $s_t \in \mathcal{S}$.
  - Call $\mathcal{S}$ the context set.
  - Given the context, the algorithm must choose an arm to follow (or a distribution over arms).
    - If the algorithm selects arm A_t at time t, then r_{t,A_t} is revealed to the algorithm.
- We'd like to compete with the best *"**policy**":*
  - A ***policy*** is a map $g : \mathcal{S} \to [K]$, which tells us which arm to use upon seeing context s.
  - Our goal is to select a sequence of arms so that after T time steps, we've incurred almost as much reward as the best policy in hindsight.

$$Regret(T) := \max_{g:\mathcal{S}\to[K]} \sum_{t=1}^{T} r_{t,g(s_t)} - \mathbb{E}\left[\sum_{t=1}^{T} r_{t,A_t}\right]$$

Contextual Bandits: some context

Consider the following example:

- We are running a sports news website. Today, there are K big sports related news stories.
- Every time a user visits our set, we must decide then and there which headlines to display to her on the front page.
- The goal is to maximize the number of clicks.

Bob just visited our website!

1. Bob loves the Toronto Raptors.
2. Hates the Los Angeles Clippers.

Which headline should we show to Bob?

Context

Arms

1.

2.

Towards an algorithm for contextual bandits

Begin with a trick. Recall, we'd like to bound the regret:

$$\mathrm{Regret(T)} = \max_{g:\mathcal{S}\to[K]}\left(\sum_{t=1}^{T} r_{t,g(s_t)} - \mathbb{E}\left[\sum_{t=1}^{T} r_{t,A_t}\right]\right)$$

Towards an algorithm for contextual bandits

Begin with a trick. Recall, we'd like to bound the regret:

$$\text{Regret(T)} = \max_{g:\mathcal{S}\to[K]} \left(\sum_{t=1}^{T} r_{t,g(s_t)} - \mathbb{E}\left[\sum_{t=1}^{T} r_{t,A_t}\right]\right)$$

$$= \max_{g:\,\mathcal{S}\to[K]} \sum_{s\in\mathcal{S}} \left(\sum_{t\,:\,s_t=s} r_{t,g(s)} - \mathbb{E}\left[r_{t,A_t}\right] \right)$$

Towards an algorithm for contextual bandits

Begin with a trick. Recall, we'd like to bound the regret:

$$\text{Regret(T)} = \max_{g:\mathcal{S}\to[K]}\left(\sum_{t=1}^{T} r_{t,g(s_t)} - \mathbb{E}\left[\sum_{t=1}^{T} r_{t,A_t}\right]\right)$$

$$= \max_{g:\,\mathcal{S}\to[K]} \sum_{s\in\mathcal{S}} \left(\sum_{t\,:\,s_t=s} r_{t,g(s)} - \mathbb{E}\left[r_{t,A_t}\right] \right)$$

$$= \sum_{s\in\mathcal{S}} \max_{g:\,\mathcal{S}\to[K]} \left(\sum_{t\,:\,s_t=s} r_{t,g(s)} - \mathbb{E}\left[r_{t,A_t}\right] \right)$$

Towards an algorithm for contextual bandits

Begin with a trick. Recall, we'd like to bound the regret:

$$\text{Regret(T)} = \max_{g:\mathcal{S}\to[K]}\left(\sum_{t=1}^{T} r_{t,g(s_t)} - \mathbb{E}\left[\sum_{t=1}^{T} r_{t,A_t}\right]\right)$$

$$= \max_{g:\mathcal{S}\to[K]}\sum_{s\in\mathcal{S}}\left(\sum_{t\,:\,s_t=s} r_{t,g(s)} - \mathbb{E}\left[r_{t,A_t}\right]\right)$$

$$= \sum_{s\in\mathcal{S}}\max_{g:\mathcal{S}\to[K]}\left(\sum_{t\,:\,s_t=s} r_{t,g(s)} - \mathbb{E}\left[r_{t,A_t}\right]\right)$$

$$= \sum_{s\in\mathcal{S}}\max_{i\in[K]}\left(\sum_{t\,:\,s_t=s} r_{t,i} - \mathbb{E}\left[r_{t,A_t}\right]\right)$$

How can we deal with this?

Towards an algorithm for contextual bandits

Let N_s = "number of steps where $s_t = s$". Then:

$$\sum_{s \in \mathcal{S}} \max_{i \in [K]} \left(\sum_{t \,:\, s_t = s} r_{t,i} - \mathbb{E}\left[r_{t,A_t} \right] \right) \le \sum_{s \in \mathcal{S}} Regret(N_s) \le \sum_{s \in \mathcal{S}} \mathcal{O}\left(\sqrt{N_s K \log K} \right) = \mathcal{O}\left(\sqrt{T |S| K log K} \right)$$

This yields a natural and simple algorithm.

$\mathcal{S}$-exp3 Algorithm

K "arms", rewards in $[\mathbf{0}, \mathbf{1}]$; context set $\mathcal{S}$.

For every context $s \in \mathcal{S}$, run an instance of the exp3 algorithm.

Theorem: The $\mathcal{S}$-exp3 algorithm has regret

$$Regret(T) = \mathcal{O}\left(\sqrt{T|S|K \log K}\right).$$

Compare with $\mathcal{O}\left(\sqrt{TK \log K}\right)$ in the non-contextual setting.

A new setting

- Consider a setting where the player has fixed ahead of time a set of policies that could be used for choosing actions.
- We can view this as an “expert” problem.
    - View each policy as an expert who knows the current context.
    - The advice of the expert is the output of the policy on the current context.
    - The notion of context is now implicit: the context is available through the expert advice.
- How is this different from the expert problem?
    - Following the advice of expert j only reveals feedback for experts who offered the same advice as expert j.
    - This partial feedback issue was also prevalent in the bandit setting.
        - Pulling lever j only revealed information about the j-th slot machine.

Bandits with expert advice: Formal setup

- K “arms”, N “experts”. The rewards of the arms are in [0,1].
- Suppose that $e_j(t)$ is the predicted arm of expert j at time t.
  - Expert j’s reward at time t is the reward of the arm he predicted.
- Using the expert advice, the algorithm must at each time step produce a distribution over arms.
- If the algorithm selects arm A_t at time t, the cost of arm A_t is revealed.
- We’d like to minimize the following:

$$Regret(T) := \max_{j \in [N]} \left(\sum_{t=1}^{T} r_{t,e_j(t)} - \mathbb{E}\left[\sum_{t=1}^{T} r_{t,A_t} \right] \right)$$

Note: we are comparing the algorithms performance to the performance of the best expert.

Using exp3

- **Main idea:** because we don't learn about all experts at step t, it is natural to view each expert as a "meta"-arm in a higher level bandit problem.
- Viewing experts as "arms" allows us to apply exp3 to obtain:

Theorem: The exp3 algorithm has regret

$$Regret(T) = \mathcal{O}\left(\sqrt{TN \log N}\right),$$

For the multi-armed bandit problem with expert advice.

Poor performance when there is a large number of experts!

Incomparable to standard bandit regret bounds because no dependence on K

But, we can obtain a bound comparable to standard bandit regret bounds.

Modifying exp3: the exp4 algorithm

- Exp3: “Exponential weights algorithm for exploration and exploitation.”
- Exp4: “Exponential weights algorithm for exploration and exploitation with experts.”

The Exp4 algorithm:

```
K “arms”, rewards in [0,1]; N “experts”; step size $\eta$.
  Initialize:
    • $q_1 \leftarrow (1/N, \dots, 1/N)$   (Uniform distribution over experts)
    • $Y_0 \leftarrow (0,0,\dots,0)$   (Cumulative rewards for experts)
  For $t = 1, \dots, T$, do:
    • Obtain expert advice, $e_1(t), \dots, e_N(t)$.
    • Select arm $A_t$ according to distribution $p_t$, where $p_{t,i} = \sum_{j : e_j(t) = i} q_{t,j}$.
    • Expected reward is $\langle p_t, r_t \rangle$.
    • Observe only $r_{t,j}$ for the chosen arm $j$.
    • $\widetilde{r_{t,j}} \leftarrow \frac{r_{t,j}}{p_{t,j}}$ if chosen expert was arm $j$, 0 otherwise.
    • $\widetilde{y_{t,j}} \leftarrow \widetilde{r_{t,j}}$ if $e_j(t) = A_t$, and 0 otherwise.
    • Update:
        • $Y_{t,j} \leftarrow Y_{t-1,j} + \widetilde{y_{t,j}}$ for every $j$.
        • $q_{t,j} \leftarrow \exp(\eta Y_{t,j}) / Z_t$,  $(Z_t = \sum_i \exp(\eta Y_{t,i}))$
```

The Exp4 algorithm:

Theorem: The exp4 algorithm has regret

$$Regret(T) = \mathcal{O}\left(\sqrt{TN \log K}\right),$$

For the multi-armed bandit problem with expert advice.

Compare with $\mathcal{O}\left(\sqrt{TN \log N}\right)$ in obtained by just running exp3.

This is still slightly incomparable with the non-contextual setting, where the regret is bounded by $\mathcal{O}\left(\sqrt{TK \log K}\right)$. We can get a bound closer to this by modifying the exp4 algorithm (mixing in the uniform distribution), to obtain:

Theorem: The modified exp4 algorithm has regret

$$Regret(T) = \mathcal{O}\left(\sqrt{TK \log N}\right),$$

For the multi-armed bandit problem with expert advice.

Can have exponentially many experts with regret bound similar to the non-contextual setting.

Application: News recommendation

[Chu, Langford, Li, Schapire 2010]

- At time t:
  - The algorithm observes:
    - the current user u_t, the features, x_{t,u_t}, of user u_t.
    - A set of articles, $\mathcal{A}_t$, and the features, $x_{t,a}$, of each article $a \in \mathcal{A}_t$.
    - **Assume:** at each step, the context can be described by d binary features.
  - Based on the features and the number of clicks on previous trials, the algorithm selects an article to display to user u_t. (Can be a randomized choice).
  - Payoff is 1 if the current user clicks on the displayed article, and 0 otherwise. We work in the stochastic setting, that is the **rewards are random.**
    - **Assume:** the expected number of clicks of an article step t is a *linear* function of its features (and the user's).
- **Goal:** minimize regret (with respect to best policy in hindsight).

A first attempt: apply exp4 algorithm

- How many experts are there?
  - Equivalent to asking how many functions are there from the context space to the "arm" space?
    - Since the context can be described by d binary features, the size of the context space is 2^d.
    - There are K arms at round t.
    - So, there are $(K)^{2^d}$ functions from the context space to the arm space.
    - That is, there are $N = (K)^{2^d}$ experts.
- So, what does exp4 give us?
  - Regret is bounded by $\mathcal{O}\left(\sqrt{TK \log N}\right) = \mathcal{O}\left(\sqrt{TK2^d \log K}\right)$, which is exponential in the number of features!
  - What is the runtime of a single iteration of exp4?
    - $\mathcal{O}(N)$ = exponential in the number of features!

A better idea: use the structure of the features

- **Main idea:**
  - The click through rate at time t is a linear function of the features.
  - Can look back in time to set up a linear system to solve for each article:

$$D_{a,t}\theta_a = b_{a,t}$$

Data matrix, rows correspond to feature vectors (contexts for article a)

The linear function we wish we knew

Vector corresponding to click/no click feedback

This ends up giving a UCB on the expected payoff of arm a. Choose the arm maximizing the UCB. They call this the LinUCB algorithm.

LinUCB Algorithm: Main result

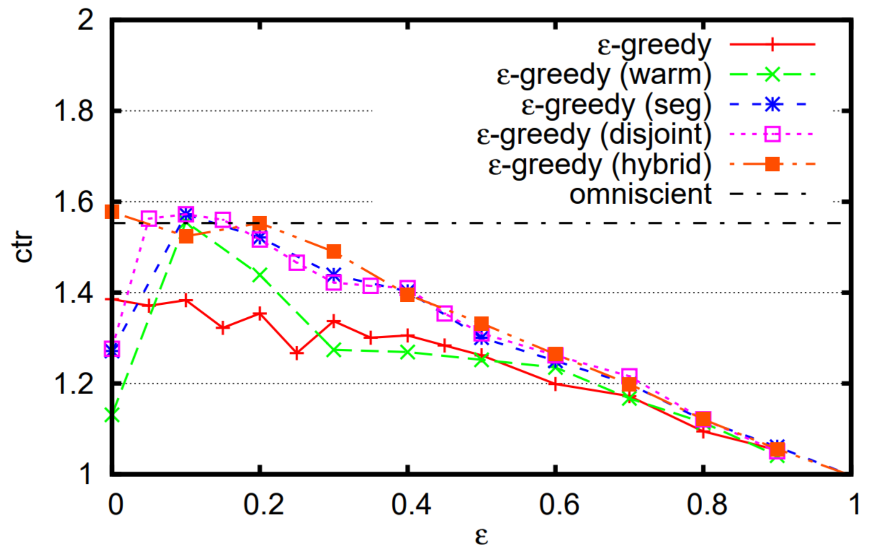
Theorem: Suppose a set of K articles is fixed ahead of time. Then, the LinUCB algorithm has

$$Regret(T) = \tilde{O}\left(\sqrt{TKd}\right),$$

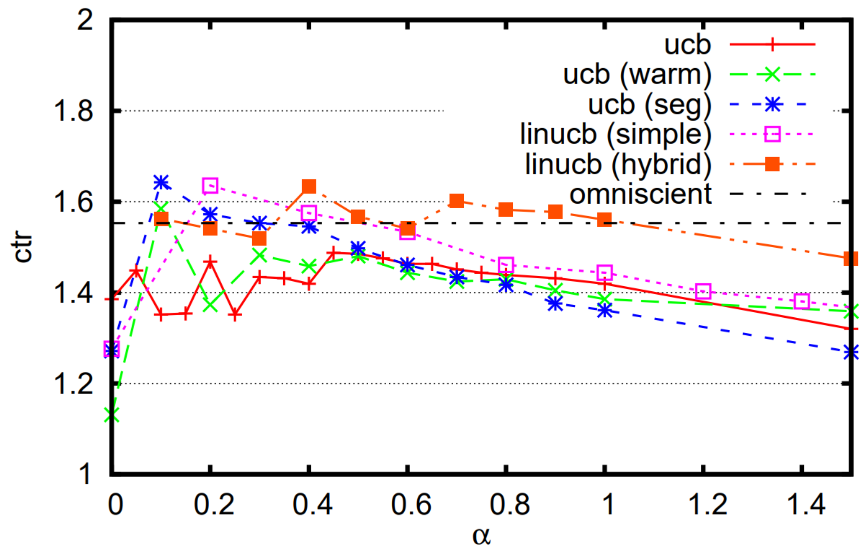
and each iteration takes $O(Kd^3)$ runtime.

Note: this algorithm removes the exponential dependence on d.

Experiments



Doesn't make use of user data



Makes use of user data