Between the Theory & Practice of Contextual Bandits in Recommender Systems

Lihong Li
lihongli.cs@gmail.com

6.7920: Reinforcement Learning: Foundations and Methods
11/14/2023

Contextual bandits

- The most widely deployed form of RL
  - Everyday use in major RS like Amazon, Facebook, Google, Netflix, Spotify, …
  - Commercialized tools on the cloud: AWS, Azure, …

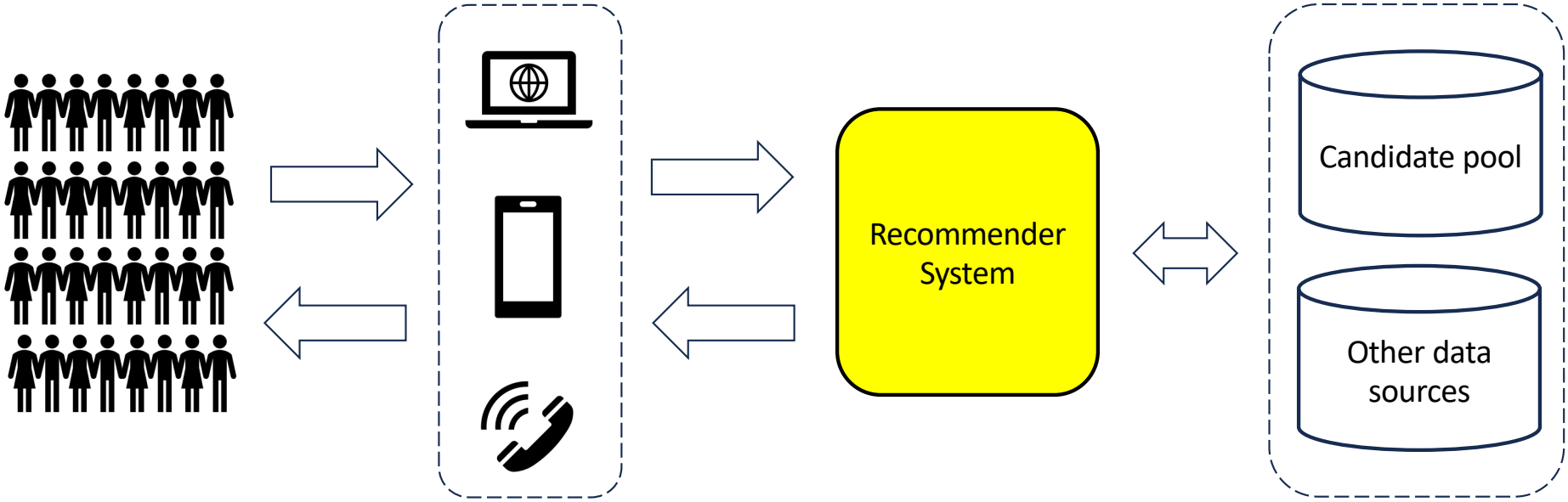
- This lecture:
  - A little shift towards practical challenges from theoretical discussions
  - Examines RS and similar applications
  - Focuses on limitations of the basic theory, and example solutions
  - Not intended to be an extensive overview

Outline

- Bandit for RS recap
- Challenging the assumptions
- Handling the complexities
- What is a good algorithm
- Q&A

Recommender systems

- Over 3 decades of research [G92, BS97]
- Everywhere on & off the Web (thanks to vast volume of data & mobile)



Example: Personalized news recommendation

"Featured Article"

A small pool of articles chosen by editors

www.yahoo.com

User-item rating matrix
(we may have numerical ratings instead of thumb up/down)

	Few drugs for super bacteria	Awkward end to Olympics	Colleges with best-paid alums	Best computers of 2010	
Alex	👍		👎		?
Betty			👍	👍	?
Charlie	👎	👎			?

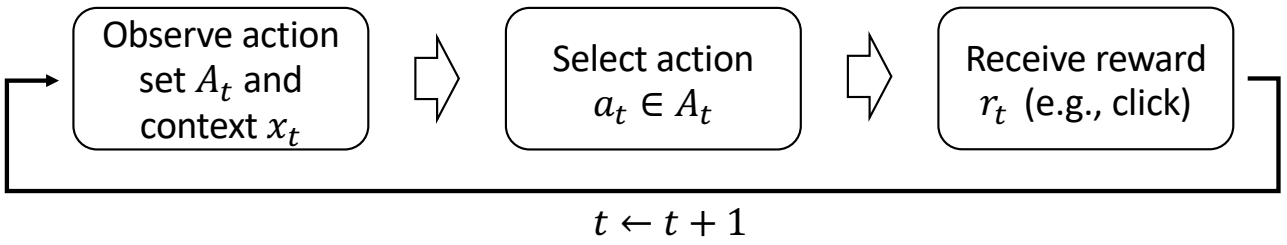
Non-RL approaches

- Traditional approaches try to predict unseen ratings
  - Collaborative filtering (CF)
  - Content-based filtering
- CF: users with similar ratings in the past will be similar in the future
  - Low-rank matrix factorization to fill in missing values in use-item rating matrix
  - Evaluated against RMSE, Precision@K, Recall, …
  - Example (square loss): $\ell(\theta) \coloneqq \frac{1}{n}\sum_i (f(x_i, a_i; \theta) - y_i)^2$
- Highly successful [ACE09, KVC09]
- Limitations
  - Cold-start problem: new items/users don't have enough data required by models; need to actively experiment to improve model prediction
  - Gap between offline *proxy* metrics (RMSE, …) and online metrics (adoption rate, ...): higher offline metric may correlate poorly with online metrics

Bandits came to the rescue

- RL view of RS as *sequential decision making*
  - Observes state/context: user features, page context, query, device, …
  - Takes an action: what to recommend
  - Receives reward: outcome/utility of the recommendation

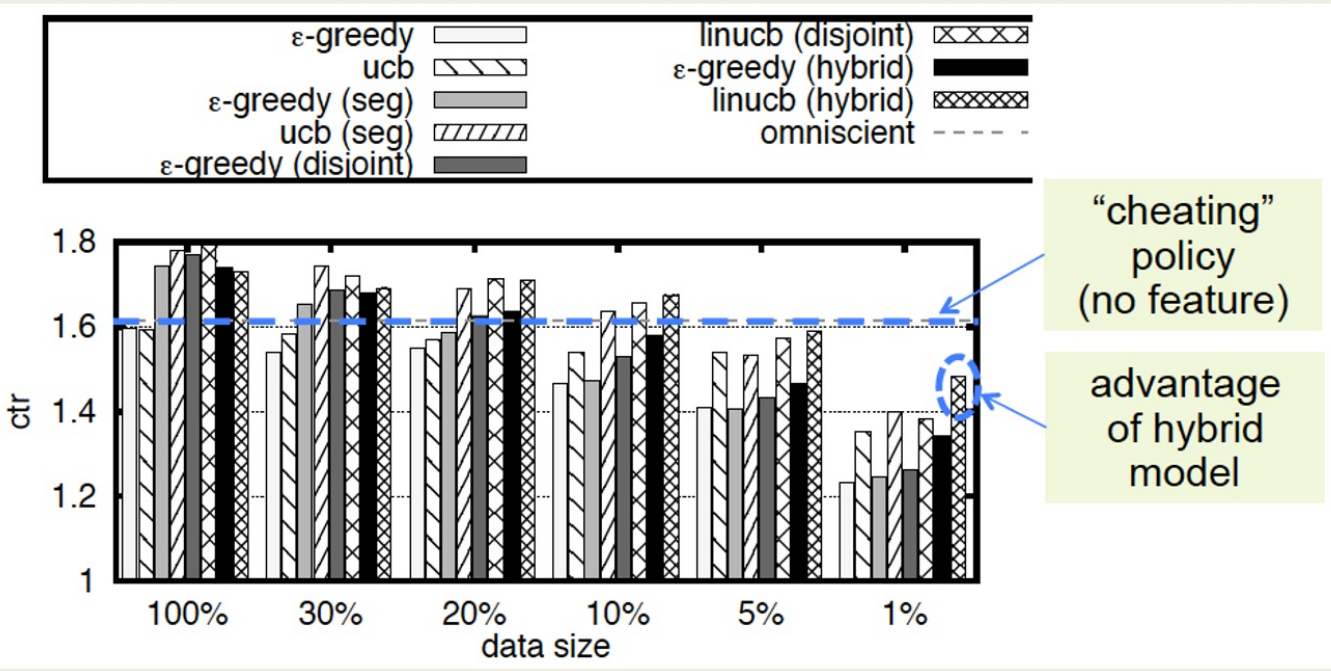
- Simplest RL setting: contextual bandits [LZ08] (aka associative RL)

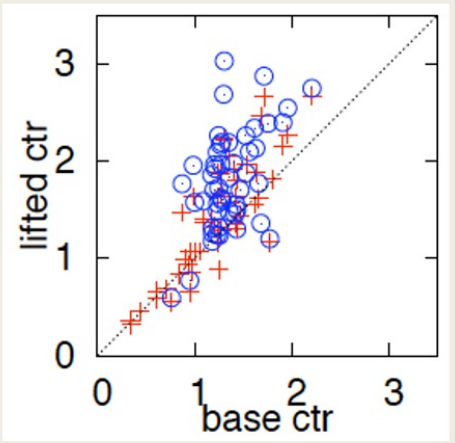


Goal: maximize $\sum_t r_t$

Plots from L+10

- LinUCB applied to personalized news recommendation [L+10]





Article CTR lift (blue for UCB, red for eps-greedy)

Rich context x_t

- Contextual signals: query, webpage, products-in-cart, device, …
- Demographic: age, gender, location, …
- Behavioral signals: previous views, clicks, purchases, visit frequency, …
- Social features: friend connections on Facebook, follows on Twitter, …

- Not all features are available
- Some features are missing for some users
  - Eg, if they don't log in, or if they opt out of personalization

Flexible choice of actions A_t

- Item: product, news, video, music, app, job, health suggestion, …
- System parameter: reserve price, online bidding, …
- Edge/node in a graph: connections on LinkedIn/Facebook
- Email/coupon: marketing promotions
- …

Diverse choice of reward r_t

- Adoption: click, subscription
- Duration: listen/watch time
- Revenue: product sales, real-time bidding
- Satisfaction: web search (use click, navigation, query reformulation etc. to derive implicit satisfaction signal)
- Wellbeing: healthcare measurements

Wide success in practice

- Numerous applications in everyday lives
- Sometimes solving the problem even without knowing it
- Example: strategies to learn new user preferences in RS [R+02]
  - "Random": similar to explore-then-commit
  - "Pure entropy": similar to pure exploration
  - "Balanced strategies": mimicking UCB
- Diversity of scenarios is contrast with the simplicity of the bandit model
  - Is the theory useful? *Yes, proven.*
  - Are there gaps between theory and practice? *Yes, we will see.*
  - How to close the gaps? *We'll see examples, opportunities and open questions.*

Outline

- Bandit for RS recap
- Challenging the assumptions
- Handling the complexities
- What is a good algorithm
- Q&A

Revisit the contextual bandit model

For $t = 1, 2, 3, \ldots$

- Observe context $x_t \sim \nu_X$
- Select one action $a_t \in A_t$
- Receive reward $r_t \sim \nu_R(\cdot \,|x_t, a_t)$

Elegant and useful mathematical model, but ..

lots of simplifying assumptions that almost never hold in practice.

Good news: in many applications, they are good enough

Stochasticity assumptions of x_t and r_t

- Exogenous factors
  - time of day, day of week, seasonality, macroeconomic, …
  - some can be added as part of context (eg, time)
  - but some are latent variables, so hard to include
- Dependence on history (past actions, as in full RL)
  - Budget in real-time bidding
  - Within-session in search and shopping
  - Repeated exposure
  - Previous medical treatments
- Further subtleties
  - multiple users sharing the same account (eg, Netflix account)
  - same user with multiple devices (“spillover effect”)

Case 1: Linear-reward assumption

- Much earlier bandit work assumed reward function is linear
  - Easier to derive closed-form updates and analyze regret
  - Still be useful in practice, but not ideal
  - Example how things may go wrong?
- Challenges
  - Poor modeling assumption leads to poor model fitting
    - Linear function for 0/1 (eg, click or not) may output -1 or 100
  - Poor fitting invalidates confidence intervals (as in LinUCB), harming exploration efficiency (both theoretically and empirically)
- Efforts
  - Bandits with generalized linear models (next)
  - Bandits with kernels [S+10]
  - Bandits with neural networks [Z+20]

Bandit with generalized linear models

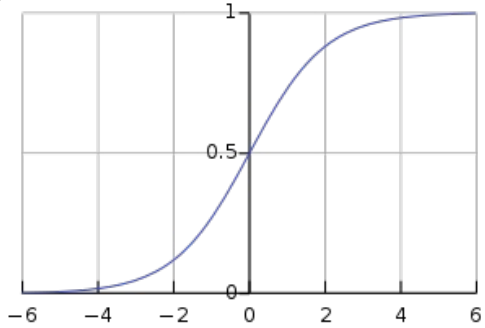
- GLM extends linear models: there exist functions $\{g, h, m\}$ such that

$$p(r|x,a) = \exp\left(\frac{ru - m(u)}{g(\eta)} + h(r,\eta)\right), \qquad u = \phi(x,a)^T\theta^*$$

- This is exponential family of distribution. It's known that

$$\mathbf{E}[r\,|x,a] = \dot{m}(u) = \sigma(u)$$

  for some fixed, strictly increasing link function $\sigma(\cdot)$



- If σ is identity, we recover the original contextual linear bandit
- Popular choice for binary reward is logistic function: $\sigma(u) = \frac{1}{1+\exp(-u)}$
- Challenges
  - No closed form of confidence intervals
  - No closed form of parameter updates

GLM-UCB and regret bound

- Extending from linear (closed form) to GLM (approximate form) [L+17]
- Update: Find maximum-likelihood solution after step t

$$\hat{\theta}_t = \arg\max_{\theta} \log \ell_n(\theta) = \arg\max_{\theta} \sum_{s=1}^{t} r_t \phi(x_t, a_t)^T \theta - m(\phi(x_t, a_t)^T \theta)$$

- Confidence interval: similar to the linear case (under regularity conditions on σ), although analysis is involved

$$\left|\phi(x,a)^T\left(\hat{\theta}_t - \theta^*\right)\right| = O\left(\sqrt{\phi(x,a)^T V_t^{-1} \phi(x,a)}\right)$$

  where $V_t = \sum_{1 \le s < t} \phi(x_s, a_s)\phi(x_s, a_s)^T$

- Regret bound of GLM-UCB: $\tilde{O}\left(\sqrt{dn}\right)$, nearly matching lower bound $\Omega\left(\sqrt{dn}\right)$

Outline

- Bandit for RS recap
- Challenging the assumptions
- Handling the complexities
- What is a good algorithm
- Q&A

Re-revisit the contextual bandit model

For $t = 1, 2, 3, \ldots$

- Observe context $x_t \sim \nu_X$
- Select one action $a_t \in A_t$
- Receive reward $r_t \sim \nu_R(\cdot \,|x_t, a_t)$

Elegant and useful model for studying the fundamental E/E trade-off, but …

- Over-simplifying in many real-world applications
- Sometimes we need to enhance the model to deal with practical complexities

Actions

- There is substantial flexibility in designing the action set
- Enumeration of candidates
- Meta candidate: each arm corresponds to one algorithm
- Combinatorial set: ranking, webpage layout
- Continuous set: RS hyper-parameters

- As size of actions (and dimension of context) increases, exploration also increase. Can we do better?

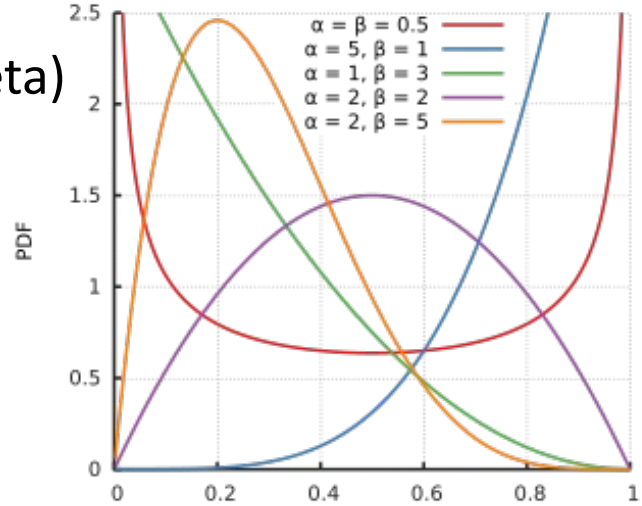
Case #2: Use of prior knowledge

- Many bandit algo & analysis assume little prior domain-knowledge
- Example: walking off the cliff (strong prior to present disasters)
- Example: finding a pizza house nearby
- A standard approach: Gittins index
  - Start with prior over action's reward distribution
  - Every new reward for an action can be used to update the posterior distribution
  - Can construct a MDP accordingly, solved by dynamic programming
  - But complex, not easy to scale beyond simple cases, and may under-explore.
- How to benefit from prior (to reduce unnecessary exploration), in a flexible and scalable way?

Thompson Sampling

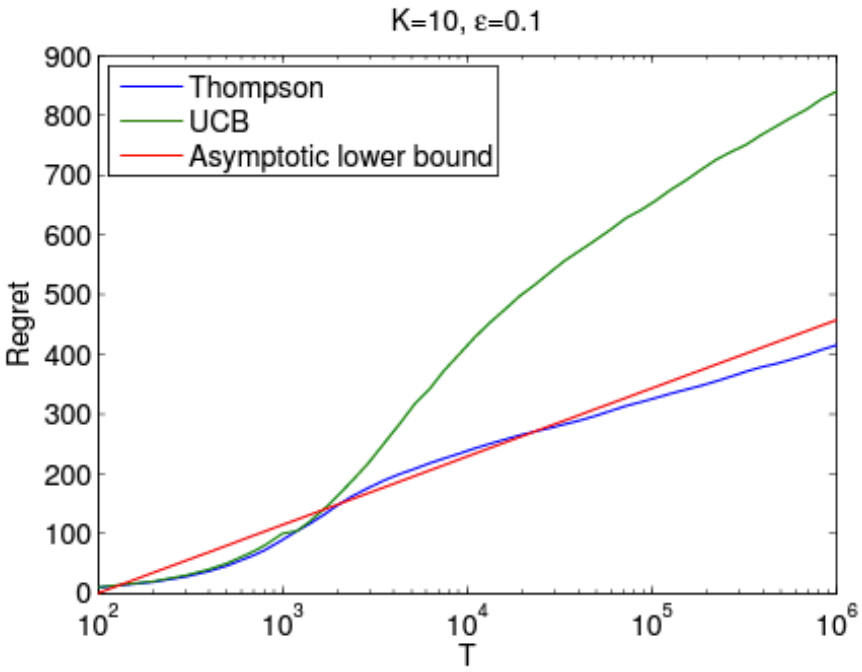
- Key ideas:
  - Start with a prior distribution of reward
  - Choose actions according to posterior probability of the actions being optimal

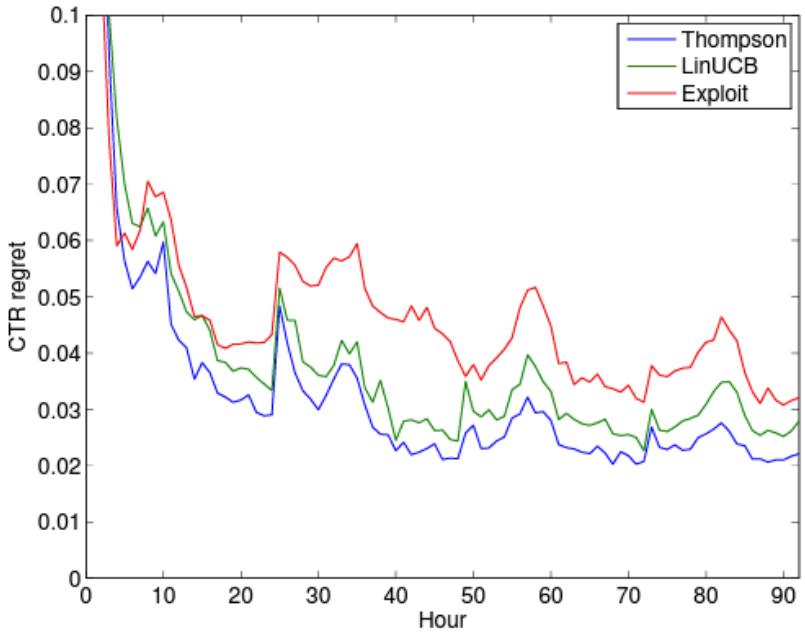
- TS for Bernoulli bandits (where prior/posterior is Beta)
  - Input: prior distribution $\alpha, \beta > 0$
  - $S_a = \alpha, F_a = \beta, \forall a \in [K]$
  - For $t = 1,2, \ldots$
    - Draw $\tilde{\theta}_a \sim \text{Beta}(S_a, F_a)$
    - Choose $a_t = \arg\max_a \tilde{\theta}_a$, and observe reward $r_t \in \{0,1\}$
    - Update: $S_a = S_a + r_t, F_a = F_a + (1 - r_t)$



- TS can be instantiated to contextual bandits and even general RL
- TS also motivates the notion of Bayesian regret

TS: Empirical comparison to UCB





Left: synthetic data. Right: Yahoo! recommendation data. [CL11]

Rewards

- Asking for a reward signal may be impractical
  - We often have user engagement of clicks etc. but not explicit thumb-up/down (next)
- Semi-bandit reward
  - We may have finer grained reward signals (in combinatorial actions)
- Delayed reward
  - Practical RS don't have fully real-time rewards, due to engineering limits, or business constraints. Example: it takes time (minutes, or even days) to lead to a purchase-based reward
- Global constraint
  - Budget of taking certain actions (eg, advertising/marketing).
- Multi-objective bandit
  - Balancing user engagement, content diversity, monetization, etc.
- Pure exploration (next)

Case #3: Absolute vs relative reward

- In ranking or multi-slot recommendation, users often don't give explicit thumb-up and downs.
- We may equate "click" with thumb-up, and "no click" with thumb-down, but this approach is noisy
  - Clicks are affected by factors other than content quality/relevance
  - Clicks are not equal. Absolute feedback is biased, due to position bias, but relative feedback is more reliable [J+07]
- Interleaving to get relative feedback

Dueling bandit

- Dueling bandit relies on relative (not absolute) reward signals [YJ09]
  - $A \subset [-1,1]^d$ is compact & convex, and $\mathbf{0} \in A$
  - A is parameter space of a ranking/recommendation function
  - For $t = 1,2,3, \dots, T$
    - Select 2 actions: $a_t, a_t' \in A$
    - Observe stochastic preference: $P(a_t \succ a_t') = \frac{1}{2} + \epsilon(a_t, a_t')$

      (ϵ is the fraction of users preferring results of a over those of a')
- Regret: $R_T = \sum_t \left(\epsilon(a^*, a_t) + \epsilon(a^*, a_t')\right)$
- Assumption: there is a differentiable & strictly concave utility function $v: A \to \Re$ s.t. $\epsilon(a, a') = \sigma\left(v(a) - v(a')\right) - \frac{1}{2}$, for some link function σ. Example is logistic function: $\sigma(x) = \frac{1}{1+\exp(-x)}$

Solving dueling bandit by gradient descent

Algorithm 1 Dueling Bandit Gradient Descent

Input: γ, δ, w_1
for query q_t $(t = 1..T)$ **do**
  Sample unit vector u_t uniformly.
  $w_t' \leftarrow \mathbf{P}_{\mathcal{W}}(w_t + \delta u_t)$ *//projected back into $\mathcal{W}$*
  Compare w_t and w_t'
  if w_t' wins **then**
    $w_{t+1} \leftarrow \mathbf{P}_{\mathcal{W}}(w_t + \gamma u_t)$ *//also projected*
  else
    $w_{t+1} \leftarrow w_t$
  end if
end for

- Regret is $O\left(n^{\frac{3}{4}}\right)$
- More recent results [S+18]

Case #4: Pure exploration

- A core challenges in bandits is balancing exploration and exploitation.
- A different scenario: pure exploration (aka best-arm identification)
- Examples
  - Quickly identify the best system parameter and deploy it
  - Quickly find the optimal recommendation strategy and serve users
  - …
- Close connection to experimental design (randomized clinical trials)

Pure exploration MAB

- Given number of rounds n and number of actions K
- For $t = 1,2, \dots, n$
  - Choose action $a_t \in [K]$
  - Receive reward $r_t \sim \mu_{a_t}$ (μ_{a_t} is unknown to the agent)
- Recommend an action $\hat{a}_n \in [K]$

- Recommendation error (assuming unique optimal action a^* with highest reward)

$$e_n = \Pr(\hat{a}_n \neq a^*)$$

UCB-E

- For $t = 1,2, \dots, n$
  - Compute upper confidence bound

$$B_t(a) = \hat{\mu}_t(a) + \sqrt{C_t / T_{t-1}(a)}$$

  - Take action $a_t = \arg\max_a B_t(a)$
  - Observe reward $r_t \sim \nu(a_t)$
  - Update statistics

$$T_t(a_t) = T_{t-1}(a_t) + 1$$

$$\hat{\mu}_t(a_t) = \frac{1}{T_t(a_t)} \sum_{s=1}^{t} r_s \cdot \mathbf{I}(a_s = a_t)$$

- For details and deeper discussion [A+10]

UCB-E regret

- In UCB-1, $C_t \sim t^{-1}$
  - A suboptimal arm is chosen $O(\log n)$ times in n rounds
  - That's how we obtained its regret
- In UCB-E, $C_t \sim t$
  - $e_n = O(n \cdot \exp(-cn))$ for some constant c
  - But it implies the cumulative regret is linear
  - Highlights an interesting and important distinction of pure exploration [BMS11]
    - Difference in objective (need for exploitation or not)
    - UCB-1 over-exploits for the purpose of pure exploration
    - UCB-E under exploits for cumulative regrets

Outline

- Bandit for RS recap
- Challenging the assumptions
- Handling the complexities
- What is a good algorithm
- Q&A

What makes a good algorithm

- Theoretical tools: sample complexity, regret
- Empirical comparison
- Practical considerations

Regret

- Regret analysis offers a beautiful and useful theoretical framework
  - Flexible with different reference point
- Gives a first-order answer, but incomplete
- Go deep into the notation and explain what can go wrong
  - Too loose (even with matching lower bound)
    - Hidden constants in big-O
  - Too coarse
    - Focuses on worse-case scenarios
    - Limits in capturing complex real-world structures/patterns
  - Too optimistic
    - Relies on assumptions that fail to hold

Practical considerations

- Favors simple algorithms
- Transparency is important
- Fewer assumptions implies greater robustness
- Use prior knowledge whenever possible
  - Bayesian prior
  - multi-task/embedding
  - warm-start model with historical data (off-policy RL)

Outline

- Bandit for RS recap
- Challenging the assumptions
- Handling the complexities
- What is a good algorithm

Q & A