Fighting Boredom in Recommender Systems with Linear Reinforcement Learning

Tiandi Ye 2020.07.31

Contents

Introduction

- Existence of an Optimal Fixed Strategy
  - matrix factorization
  - multi-armed bandit (MAB)
  - A/B testing
- Boredom Effect
  - movie recommendation problem
  - meal taste

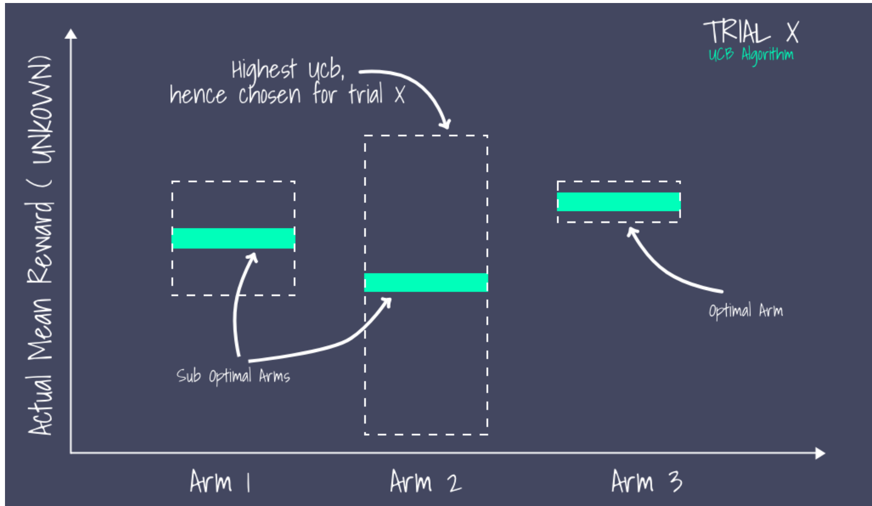
Related Work

- Once an arm is pulled, *its* reward decreases due to loss of interest and never increases again.
- Rewards continuously decrease whether the arm is selected or not.
- MDP-based RS, next item reward depends on previously k selected items without any underlying model assumption. Without considering exploration-exploitation trade-off and directly solving an estimated MDP leads to linear regret.
- Two possible states sensitization and boredom.

Multi-Armed Bandit(MAB)

Context-Free

- $\epsilon - Greedy$
- $Softmax$
  - $P(k) = \frac{e^{\frac{Q(k)}{\tau}}}{\sum_{i=1}^{K} e^{\frac{Q(i)}{\tau}}}$ $\tau \to 0, \text{exploitation} - \text{only}; \tau \to \infty, \text{exploration} - \text{only}$
- $Upper\ Confidence\ Bound(UCB)$
  - Reward upper confidence bound: $I_i = u_i + \sqrt{\frac{2\ln(n)}{n_i}}$



Multi-Armed Bandit(MAB)

Contextual Bandit

- *LinUCB*

A contextual-bandit algorithm **A** proceeds in discrete trials t = 1, 2, 3, . . .

In trial t:

1. The algorithm observes the current user u_t and a set $\mathcal{A}_t$ of arms or actions together with their feature vectors $X_{t,a}$ for $a \in \mathcal{A}_t$. The vector $X_{t,a}$ summarizes information of both the user u_t and arm a, and will be referred to as the context.
2. Based on observed payoffs in previous trials, **A** chooses an arm $a_t \in \mathcal{A}_t$, and receives payoff r_{t,a_t} whose expectation depends on both the user u_t and the arm a_t.
3. The algorithm then improves its arm-selection strategy with the new observation, $\left(X_{t,a_t}, a_t, r_{t,a_t}\right)$.

Multi-Armed Bandit(MAB)

Contextual Bandit

- *LinUCB*

we assume the expected payoff of an arm a is linear in its d-dimensional feature $X_{t,a}$ with some unknown coefficient vector θ_a^*, namely, for all t:

$$E\left[r_{t,a}|X_{t,a}\right] = X_{t,a}^T\theta_a^*$$

$$L = (c_a - D_a\theta_a)^2 + I_d$$

$$\hat{\theta}_a = argmin_{\theta_a}L = (D_a^T D_a + I_d)^{-1}D_a^T c_a$$

$$a_t \stackrel{\text{def}}{=} argmax_{a\in\mathcal{A}_t}\left(X_{t,a}^T\hat{\theta}_a + \alpha\sqrt{X_{t,a}^T A_a^{-1}X_{t,a}}\right)$$

$$A_a \stackrel{\text{def}}{=} D_a^T D_a + I_d$$

D_a : a design matrix of dimension $m \times d$ at trial t

I_d: $d \times d$ identity matrix

c_a: corresponding response vector

Problem Formulation

Determinininistic MDP: $M = \langle S, [K], f, r \rangle$

- Action: $a \in \{1, \dots, K\} = [K]$
- State: $s_t = (a_{t-w}, \dots, a_{t-1})$
  - recency function $\rho(s_t, a) = \sum_{\tau=1}^{w} \mathbb{I}\{a_{t-\tau} = a\}/\tau$,
  - e.g. $a_1 a_2 a_1$,$\rho(s_t, a_1) = \frac{1}{1} + \frac{0}{2} + \frac{1}{3}$
- Transition function f: $S \times [K] \rightarrow S$, drops the action selected w steps ago and appends the last action to the state.
  - e.g. $\{a_1 a_2 a_3\} \times a_4 = \{a_2 a_3 a_4\}$
- $r(s_t, a) = \sum_{j=0}^{d} \theta_{a,j}^{*} \rho\left(s_{t,} a\right)^{j} = x_{s,a}^{T} \theta_{a}^{*}$
  - context vector for action a $x_{s,a} = [1, \rho(s,a), \dots, \rho(s,a)^d] \in \mathbb{R}^{d+1}$
  - Unknown vector $\theta_a^* \in \mathbb{R}^{d+1}$

Value Iteration (θ_a^* *were known*)

- $u_{i+1}(s) = \max_{a \epsilon [K]} \left[r(s,a) + u_i\big(f(s,a)\big) \right]$
- regret: $\Delta(T) = T\eta^* - \sum_{t=1}^{T} r(s_t, a_t)$

Model Validation on Real Data(movielens-100k)

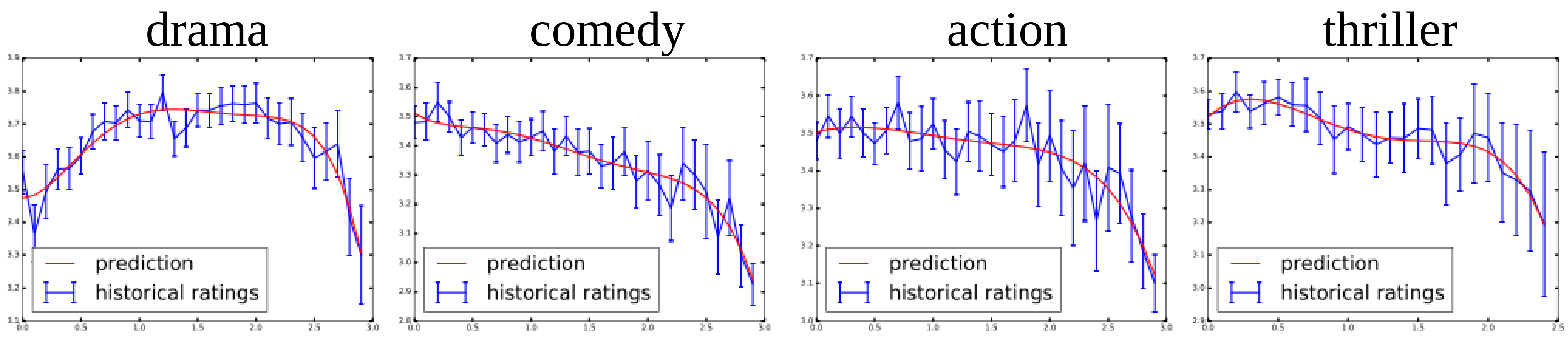


Figure 1: Average rating as a function of the recency for different genre of movies ($w = 10$) and predictions of our model for $d = 5$ in red. From left to right, *drama*, *comedy*, *action* and *thriller*. The confidence intervals are constructed based on the amount of samples available at each state s and the red curves are obtained by fitting the data with the model in Eq. $\boxed{1}$.

$$\underline{(drama, r_1), (comedy, r_2)}, (comedy, r_3), \underline{(thcomedy, r_4), (action, r_5)}, (comedy, r_6)$$

Model Validation on Real Data

Genre	$d = 1$	$d = 2$	$d = 3$	$d = 4$	$d = 5$	$d = 6$
action	0.55	0.74	0.79	0.81	0.81	0.82
comedy	0.77	0.85	0.88	0.90	0.90	0.91
drama	0.0	0.77	0.80	0.83	0.86	0.87
thriller	0.74	0.81	0.83	0.91	0.91	0.91

Table 1: R^2 for the different genres and values of d on *movielens-100k* and a window $w = 10$.

$$r(s_t, a) = \sum_{j=0}^{d} \theta^*_{a,j} \rho\left(s_{t,} a\right)^j = x^T_{s,a} \theta^*_a$$

$$R^2 = \frac{RSS}{TSS} = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

LinUCRL Linear Upper-Confidence bound for Reinforcement Learning

- $\hat{\theta}_{t,a} = min_{\theta} \sum_{\tau<t:a_\tau=a} \left(x^T_{s_\tau,a}\theta - r_\tau\right)^2 + \lambda\|\theta\|_2$ （ridge regression）
- $\hat{\theta}_{t,a} = V^{-1}_{t,a} X^T_{t,a} R_{t,a}$
  - $R_{a,t}$: vector of rewards obtained up to time t when a was executed
  - $X_{a,t}$: the feature matrix corresponding to the contexts observed so far
  - $V_{t,a} = \left(X^T_{t,a} X_{t,a} + \lambda I\right) \in \mathbb{R}^{(\mathrm{d}+1)\times(d+1)}$
- $\hat{r}_t(s,a) = x^T_{s,a}\hat{\theta}_{t,a}$
- upper-confidence bound: $\widetilde{r_t}(s,a) = \hat{r}_t(s,a) + c_{t,a}\left\|x_{s,a}\right\|_{V^{-1}_{t,a}}$
- $\widetilde{M}_k = \langle S, [K], f, \tilde{r}_k\rangle$ ➔ $\tilde{\pi}_k$

Algorithm

Algorithm 1 The LinUCRL algorithm.

Init: Set $t = 0$, $T_a = 0$, $\widehat{\theta}_a = \mathbf{0} \in \mathbb{R}^{d+1}$, $V_a = \lambda I$
for rounds $k = 1, 2, \cdots$ **do**
  Set $t_k = t$, $\nu_a = 0$
  Compute $\widehat{\theta}_a = V_a^{-1} X_a^{\mathsf{T}} R_a$
  Set optimistic reward $\widetilde{r}_k(s, a) = x_{s,a}^{\mathsf{T}} \widehat{\theta}_a + c_{t,a} \|x_{s,a}\|_{V_a^{-1}}$
  Compute optimal policy $\widetilde{\pi}_k$ for MDP $(S, [K], f, \widetilde{r}_t)$
  while $\forall a \in [K], T_a < \nu_a$ **do**
    Choose action $a_t = \widetilde{\pi}_k(s_t)$
    Observe reward r_t and next state s_{t+1}
    Update $X_{a_t} \leftarrow [X_{a_t}, x_{s_t,a_t}]$, $R_{a_t} \leftarrow [R_{a_t}, r_t]$, $V_{a_t} \leftarrow V_{a_t} + x_{s_t,a_t} x_{s_t,a_t}^{\mathsf{T}}$
    Set $\nu_{a_t} \leftarrow \nu_{a_t} + 1$, $t \leftarrow t + 1$
  end while
  Set $T_a \leftarrow T_a + \nu_a, \forall a \in [K]$
end for

LinUCRL

- Computational complexity：
  - LinUCRL
  - $u_{i+1}(s) = \max_{a\epsilon[K]}\left[r(s,a) + u_i\big(f(s,a)\big)\right]$
  - $O(dSK)$

  - UCRL
  - $u_{i+1}(s) = \max_{a\epsilon\mathcal{A}}\left\{\tilde{r}_k(s,a) + \max_{p(\cdot)\epsilon\mathcal{P}(s,a)}\left\{\sum_{s'\epsilon S} p(s') \cdot u_i(s')\right\}\right\}$
  - $O(S^2K)$

Theoretical Analysis

- Known constants B and R such that $\|\theta_a^*\|_2 \leq$ B for all actions $a \in [K]$ and the noise is sub-Gaussian with parameter R.
- $\ell_\omega = log(\omega) + 1$
- $L_\omega^2 = \frac{1-\ell_\omega^{d+1}}{1-\ell_\omega}$
- $T_{t,a}$: the number of samples collected from action a up to t
- run LINUCRL with the scaling factor

$$c_{t,a} = R\sqrt{(d+1)\log\left(Kt^\alpha\left(1+\frac{T_{t,a}L_w^2}{\lambda}\right)\right)} + \lambda^{1/2}B$$

Theoretical Analysis.

Cumulative regret

$$\Delta(\textsc{LinUCRL}, T) \leq Kw \log_2\left(\frac{8T}{K}\right) + 2c_{\max}\sqrt{2KT(d+1)\log\left(1 + \frac{TL_w^2}{\lambda(d+1)}\right)}$$

$c_{max} = max_{t,a} c_{t,a}$

per-step regret Δ/T decreases to zero as $1/\sqrt{T}$

Experiments

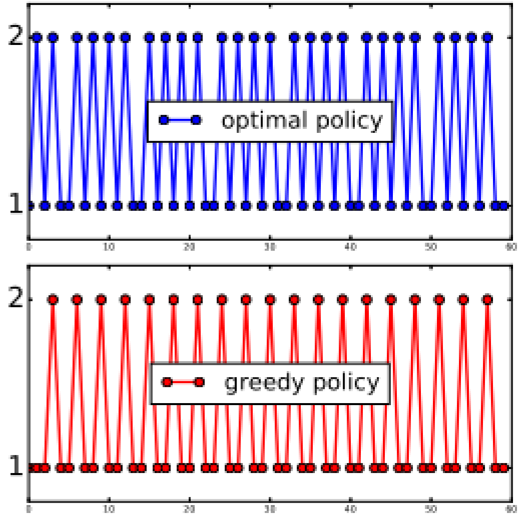
- Toy experiment
- Movielens
- Real-world data from A/B testing

Toy experiment

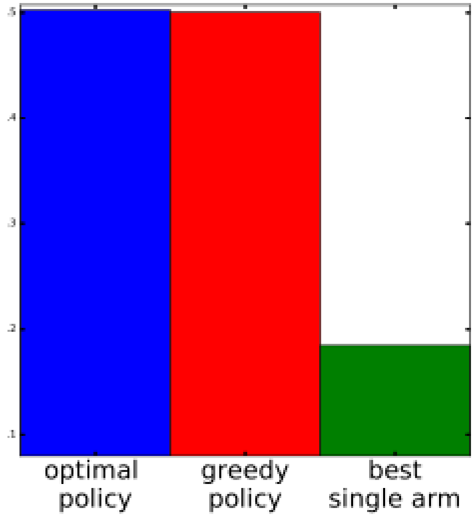
- $K = 2, d = 1, \omega =?$
- $\theta_1^* = (1, c_1), \theta_2^* = (1/\alpha, c_2)$

- optimal policy: maximizing the average reward η
- greedy policy: $a_t = argmax_a r(s_t, a)$
- fixed-action policy: $a_t = argmax\{1, 1/\alpha\}$

$c_1 = 0.3 \approx c_2 = 0.4, \alpha = 1.5$(limited boredom effect)



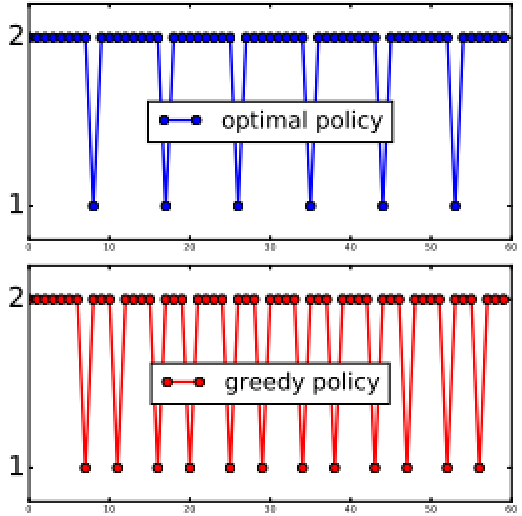
(a) sequence of actions



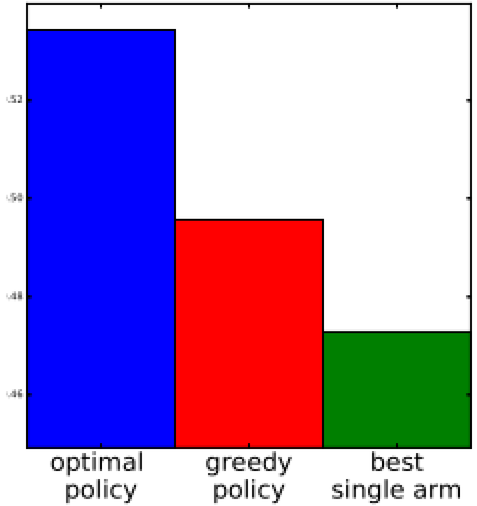
(b) average reward

Difference is very narrow.

$c_1 = 2 \gg c_2 = 0.01, \alpha = 2.0$(strong boredom effect)



(c) sequence of actions



(d) average reward

Greedy policy: 66% for action 1
Optimal policy: 57% for action 1

Movielens-100k

userId::movieId::rating::timestamp

- 100,000 ratings (1-5) from 943 users on 1682 movies
  - Each user has rated at least 20 movies.
- Ratings are made on a 5-star scale, with half-star increments (0.5 stars - 5.0 stars).

K = 10 actions corresponding to different genres of movies

d = 5

w = 5

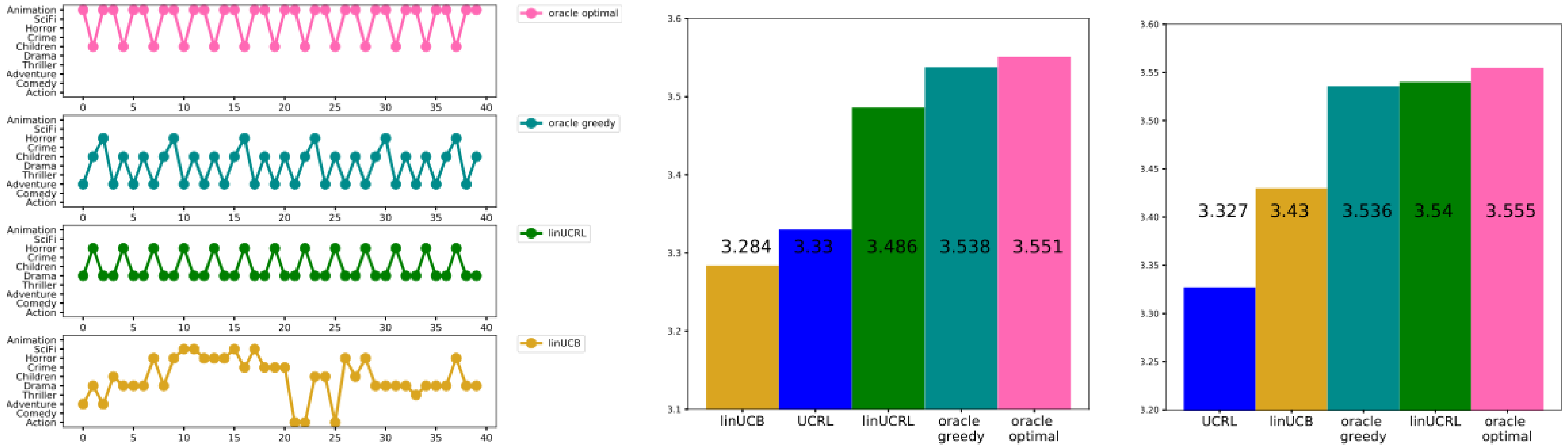
Resulting Parameters

Genre	$\theta^*_{a,0}$	$\theta^*_{a,1}$	$\theta^*_{a,2}$	$\theta^*_{a,3}$	$\theta^*_{a,4}$	$\theta^*_{a,5}$
Action	3.1	0.54	-1.08	0.78	-0.22	0.02
Comedy	3.34	0.54	-1.08	0.78	-0.22	0.02
Adventure	3.51	0.86	-2.7	3.06	-1.46	0.24
Thriller	3.4	1.26	-2.9	2.76	-1.14	0.16
Drama	2.75	1.0	0.94	-1.86	0.94	-0.16
Children	3.52	0.1	0.0	-0.3	0.2	-0.04
Crime	3.37	0.32	1.12	-3.0	2.26	-0.54
Horror	3.54	-0.68	1.84	-2.04	0.82	-0.12
SciFi	3.3	0.64	-1.32	1.1	-0.38	0.02
Animation	3.4	1.38	-3.44	3.62	-1.62	0.24

Table 3: Reward parameters of each genre for the *movielens* experiment.

a constant strategy would always pull the comedy genre since it is the one with the highest “static” reward ?

Results



(a) Last 40 actions (b) Avg. rwd. at $T = 200$ (c) Avg. rwd. at the end

Figure 3: Results of learning experiment based on *movielens* dataset.

Despite the fact that UCRL targets this better performance, the learning process is very slow as the number of states is too large. ?

Large scale A/B testing dataset

350M tuples (user id, timestamp, version, click)

Don't impose any linear assumption on the simulator.

Algorithm	on the T steps	on the last steps
only B	46.0%	46.0%
UCRL	46.5%	46.0%
LINUCRL	66.7%	75.8%
oracle greedy	61.3%	61.3%
oracle optimal	95.2%	95.2%

Table 2: Relative improvement over *only A* of learning experiment based on *large scale A/B testing* dataset.

Conclusion

- Outlook
  - Correlations between actions.
  - Offer personalized “boredom” curves.
  - Different models of the reward as a function of the recency(logistic regression in case of binary rewards).

- Pros
  - Innovation
  - Deterministic MDP
    - Per step regret: $O(1/\sqrt{T})$
    - Computational complexity: $O(dSK)$
- Cons
  - More baselines are necessary.

Thanks for listening.