Learning to Rank with Click Models: From Online Algorithms to Offline Evaluations

Shuai LI

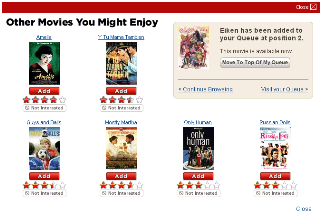
The Chinese University of Hong Kong

Outline

1. Motivation
2. Background
3. Problem Definition – Online
4. Click Models
    - Cascade Model (CM)
        - ICML'2016
        - AAAI'2018
        - IJCAI'2019
    - Dependent Click Model – A co-authored work
    - Position-Based Model
    - General Click Models – A co-authored work, ICML'2019
5. Offline Evaluations – KDD'2018
6. Conclusions

Outline

Motivation – Learning to Rank



Amazon, YouTube, Facebook, Netflix, Taobao

Outline

Background – Multi-armed Bandit Problem

- A special case of reinforcement learning
- There are L arms
  - Each arm a has an unknown reward distribution with unknown mean α_a
  - The best arm is $a^* = \mathrm{argmax}\ \alpha_a$

Background – Multi-armed Bandit Setting

- At each time t
  - The learning agent selects one arm a_t
  - Observe the reward $X_{a_t,t}$

Background – Multi-armed Bandit Setting

- At each time t
  - The learning agent selects one arm a_t
  - Observe the reward $X_{a_t,t}$
- The objective is to minimize the regret in T rounds

$$R(T) = T\alpha^* - \mathbb{E}\left[\sum_{t=1}^{T} \alpha_{a_t}\right]$$

Background – Multi-armed Bandit Setting

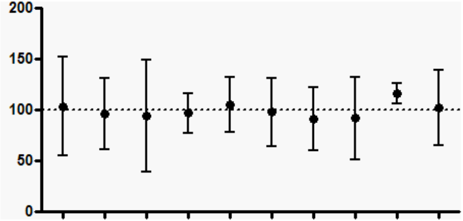
- At each time t
  - The learning agent selects one arm a_t
  - Observe the reward $X_{a_t,t}$
- The objective is to minimize the regret in T rounds

$$R(T) = T\alpha^* - \mathbb{E}\left[\sum_{t=1}^{T} \alpha_{a_t}\right]$$

- Balance the trade-off between exploitation and exploration
  - Exploitation: select arms that yield good results so far
  - Exploration: select arms that have not been tried much before

Background – Upper Confidence Bound

- UCB (Upper Confidence Bound) [ACF'02]



- UCB policy: select

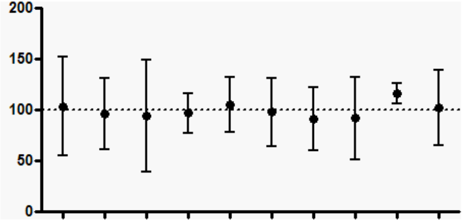
$$a_t = \mathrm{argmax}_a \;\; \hat{\alpha}_{a,t} + \sqrt{\frac{3\ln(t)}{2T_a(t)}}$$

where

- $\hat{\alpha}_{a,t}$ is the empirical mean of arm a in time t — Exploitation
- $T_a(t)$ is the played times of arm a — Exploration

Background – Upper Confidence Bound

- UCB (Upper Confidence Bound) [ACF'02]



  - UCB policy: select

$$a_t = \operatorname{argmax}_a \ \hat{\alpha}_{a,t} + \sqrt{\frac{3\ln(t)}{2T_a(t)}}$$

    where

    - $\hat{\alpha}_{a,t}$ is the empirical mean of arm a in time t — Exploitation
    - $T_a(t)$ is the played times of arm a — Exploration
  - Gap-dependent bound $O(\frac{L}{\Delta}\log(T))$ where $\Delta = \min_{\alpha_a<\alpha^*} \alpha^* - \alpha_a$, match lower bound
  - Gap-free bound $O(\sqrt{LT\log(T)})$ tight up to a factor of $\sqrt{\log(T)}$

Outline

Online Learning to Rank

- There are L items
  - Each item a with an unknown attractiveness $\alpha(a)$
- There are K positions

Online Learning to Rank

- There are L items
  - Each item a with an unknown attractiveness $\alpha(a)$
- There are K positions
- At time t
  - The learning agent selects a list of items $A_t = (a_1^t, \ldots, a_K^t)$
  - Receive the click feedback $C_t \in \{0, 1\}^K$

Online Learning to Rank

- There are L items
  - Each item a with an unknown attractiveness $\alpha(a)$
- There are K positions
- At time t
  - The learning agent selects a list of items $A_t = (a_1^t, \ldots, a_K^t)$
  - Receive the click feedback $C_t \in \{0, 1\}^K$
- The objective is to minimize the regret over T rounds

$$R(T) = T\ r(A^*) - \mathbb{E}\left[\sum_{t=1}^{T} r(A_t)\right]$$

where

- $r(A)$ is the reward of list A
- $A^* = (1, 2, \ldots, K)$ by assuming arms are ordered by $\alpha(1) \geq \alpha(2) \geq \cdots \geq \alpha(L)$

Outline

Contents

Click Models

- Click models describe how users interact with a list of items

Click Models

- Click models describe how users interact with a list of items
- Cascade Model (CM)
  - Assumes the user checks the list from position 1 to position K, clicks at the first satisfying item and stops

Click Models

- Click models describe how users interact with a list of items
- Cascade Model (CM)
  - Assumes the user checks the list from position 1 to position K, clicks at the first satisfying item and stops
  - At most 1 click

Click Models

- Click models describe how users interact with a list of items
- Cascade Model (CM)
  - Assumes the user checks the list from position 1 to position K, clicks at the first satisfying item and stops
  - At most 1 click
  - $r(A) = 1 - \prod_{k=1}^{K}(1 - \alpha(a_k)) = \mathrm{OR}(\alpha(a_1), \ldots, \alpha(a_K))$

Click Models

- Click models describe how users interact with a list of items
- Cascade Model (CM)
  - Assumes the user checks the list from position 1 to position K, clicks at the first satisfying item and stops
  - At most 1 click
  - $r(A) = 1 - \prod_{k=1}^{K}(1 - \alpha(a_k)) = \mathrm{OR}(\alpha(a_1), \ldots, \alpha(a_K))$
  - The meaning of received feedback $(0, 0, 1, 0, 0)$

Click Models

- Click models describe how users interact with a list of items
- Cascade Model (CM)
  - Assumes the user checks the list from position 1 to position K, clicks at the first satisfying item and stops
  - At most 1 click
  - $r(A) = 1 - \prod_{k=1}^{K}(1 - \alpha(a_k)) = \mathrm{OR}(\alpha(a_1), \ldots, \alpha(a_K))$
  - The meaning of received feedback $(0, 0, 1, 0, 0)$

✗

✗

✓

?

?

Click Models

- Click models describe how users interact with a list of items
- Cascade Model (CM)
  - Assumes the user checks the list from position 1 to position K, clicks at the first satisfying item and stops
  - At most 1 click
  - $r(A) = 1 - \prod_{k=1}^{K}(1 - \alpha(a_k)) = \mathrm{OR}(\alpha(a_1), \dots, \alpha(a_K))$
  - The meaning of received feedback $(0, 0, 1, 0, 0)$

✗
✗
✓
?
?

	Click Model	Regret
[KSWA, 2015]	CM	$O(\frac{L}{\Delta}\log(T))$

Outline

Contextual Bandit Setting

- Contexts
  - User profiles, search keywords
  - Important for search and recommendations

Contextual Bandit Setting

- Contexts
  - User profiles, search keywords
  - Important for search and recommendations
- Assume each item a is represented by $x_{t,a} \in \mathbb{R}^d$

Contextual Bandit Setting

- Contexts
  - User profiles, search keywords
  - Important for search and recommendations
- Assume each item a is represented by $x_{t,a} \in \mathbb{R}^d$
- Assume the attractiveness for item a

$$\alpha_t(a) = \theta^\top x_{t,a}$$

by a fixed but unknown weight vector θ

Contextual Bandit Setting

- Contexts
  - User profiles, search keywords
  - Important for search and recommendations
- Assume each item a is represented by $x_{t,a} \in \mathbb{R}^d$
- Assume the attractiveness for item a

$$\alpha_t(a) = \theta^\top x_{t,a}$$

  by a fixed but unknown weight vector θ
- When $x_{t,a}$'s are one-hot representations, and $\theta = (\alpha(1), \dots, \alpha(L))$, it returns to multi-armed bandit setting.

Contextual Combinatorial Cascading Bandits[LWZC, ICML'2016] – Algorithm

- C^3-UCB Algorithm
  - Initialization: $\hat{\theta} = 0 \in \mathbb{R}^{d \times 1}$, $V = \lambda I \in \mathbb{R}^{d \times d}$, $b = 0 \in \mathbb{R}^{d \times 1}$

Contextual Combinatorial Cascading Bandits[LWZC, ICML'2016] – Algorithm

- C^3-UCB Algorithm
  - Initialization: $\hat{\theta} = 0 \in \mathbb{R}^{d \times 1}$, $V = \lambda I \in \mathbb{R}^{d \times d}$, $b = 0 \in \mathbb{R}^{d \times 1}$
  - For time $t = 1, 2, \ldots$

Contextual Combinatorial Cascading Bandits[LWZC, ICML'2016] – Algorithm

- C^3-UCB Algorithm
  - Initialization: $\hat{\theta} = 0 \in \mathbb{R}^{d \times 1}$, $V = \lambda I \in \mathbb{R}^{d \times d}$, $b = 0 \in \mathbb{R}^{d \times 1}$
  - For time $t = 1, 2, \ldots$
    - Obtain items $\{x_{t,a}\}_{a \in E} \subset \mathbb{R}^{d \times 1}$

Contextual Combinatorial Cascading Bandits[LWZC, ICML'2016] – Algorithm

- C^3-UCB Algorithm
  - Initialization: $\hat{\theta} = 0 \in \mathbb{R}^{d \times 1}$, $V = \lambda I \in \mathbb{R}^{d \times d}$, $b = 0 \in \mathbb{R}^{d \times 1}$
  - For time $t = 1, 2, \ldots$
    - Obtain items $\{x_{t,a}\}_{a \in E} \subset \mathbb{R}^{d \times 1}$
    - With high probability

$$\left\| \hat{\theta} - \theta \right\|_V \leq \beta_t$$

thus with high probability

$$\alpha_t(a) \in \hat{\theta}^\top x_{t,a} \pm \beta_t \left\| x_{t,a} \right\|_{V^{-1}}$$

Contextual Combinatorial Cascading Bandits[LWZC, ICML'2016] – Algorithm

- C^3-UCB Algorithm
  - Initialization: $\hat{\theta} = 0 \in \mathbb{R}^{d\times 1}$, $V = \lambda I \in \mathbb{R}^{d\times d}$, $b = 0 \in \mathbb{R}^{d\times 1}$
  - For time $t = 1, 2, \dots$
    - Obtain items $\{x_{t,a}\}_{a\in E} \subset \mathbb{R}^{d\times 1}$
    - With high probability

$$\left\|\hat{\theta} - \theta\right\|_V \le \beta_t$$

      thus with high probability

$$\alpha_t(a) \in \hat{\theta}^\top x_{t,a} \pm \beta_t \left\|x_{t,a}\right\|_{V^{-1}}$$

    - Select the list A_t by UCBs of arms $U_t(a) = \hat{\theta}^\top x_{t,a} + \beta_t \left\|x_{t,a}\right\|_{V^{-1}}$

Contextual Combinatorial Cascading Bandits[LWZC, ICML'2016] – Algorithm

- C^3-UCB Algorithm
  - Initialization: $\hat{\theta} = 0 \in \mathbb{R}^{d\times 1}$, $V = \lambda I \in \mathbb{R}^{d\times d}$, $b = 0 \in \mathbb{R}^{d\times 1}$
  - For time $t = 1, 2, \dots$
    - Obtain items $\{x_{t,a}\}_{a\in E} \subset \mathbb{R}^{d\times 1}$
    - With high probability

$$\left\| \hat{\theta} - \theta \right\|_V \leq \beta_t$$

thus with high probability

$$\alpha_t(a) \in \hat{\theta}^\top x_{t,a} \pm \beta_t \left\| x_{t,a} \right\|_{V^{-1}}$$

    - Select the list A_t by UCBs of arms $U_t(a) = \hat{\theta}^\top x_{t,a} + \beta_t \left\| x_{t,a} \right\|_{V^{-1}}$
    - Receive feedback $C_t \in \{0,1\}^K$
    - Compute the stopping position $K_t = \min\{k : C_t(k) = 1\} \cup \{K\}$ and update

$$V \leftarrow V + \sum_{k=1}^{K_t} x_{t,a_k^t} x_{t,a_k^t}^\top, \quad b \leftarrow b + \sum_{k=1}^{K_t} x_{t,a_k^t} C_t(k)$$

$$\hat{\theta} = V^{-1} b$$

Contextual Combinatorial Cascading Bandits[LWZC, ICML'2016] – Results

- We prove a regret bound

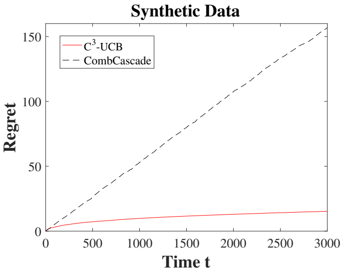
$$R(T) = O\left(\frac{d}{p^*}\sqrt{TK}\ln(T)\right)$$

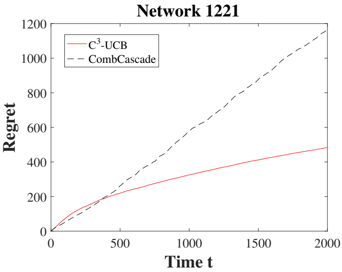
Contextual Combinatorial Cascading Bandits[LWZC, ICML'2016] – Results

- We prove a regret bound

$$R(T) = O\left(\frac{d}{p^*}\sqrt{TK}\ln(T)\right)$$

- Experimental results —Ours —CombCascade





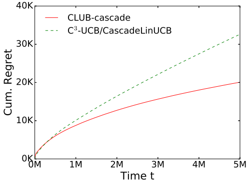
Summary on Bandits with Click Models

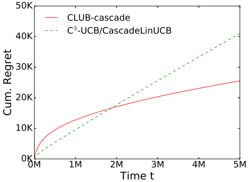
	Context	Click Model	Regret
[KSWA, 2015]	-	CM	$O(\frac{L}{\Delta}\log(T))$
[LWZC, ICML'2016]	Linear	CM	$O(\frac{d}{p^*}\sqrt{TK}\log(T))$

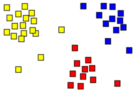
Outline

Online Clustering of Contextual Cascading Bandits [LZ, AAAI'2018]

- Find clustering over users as well as recommending
- The attractiveness function is generalized linear (GL)
- Improve the regret results
- Experiments —Ours · · · C^3-UCB







	Context	Click Model	Regret
[KSWA, 2015]	-	CM	$O(\frac{L}{\Delta}\log(T))$
[LWZC, ICML'2016]	Linear	CM	$O(\frac{d}{p^*}\sqrt{TK}\log(T))$
[LZ, AAAI'2018]	GL	CM	$O(d\sqrt{TK}\log(T))$

Outline

Improved Algorithm on Clustering Bandits [LCLL, IJCAI'2019]

- Arbitrary frequency distribution over users (compared to uniform distribution)
- Prove a regret bound that is free of the minimal frequency over users

$$R(T) = O\left(d\sqrt{mT}\ln(T) + \left(\frac{1}{\gamma_p^2} + \frac{n_u}{\gamma^2\lambda_x^3}\right)\ln(T)\right)$$

(compared to $R(T) = O\left(d\sqrt{mT}\ln(T) + \frac{1}{p_{\min}\gamma^2\lambda_x^3}\ln(T)\right)$)
where n_u is number of users and m is number of clusters

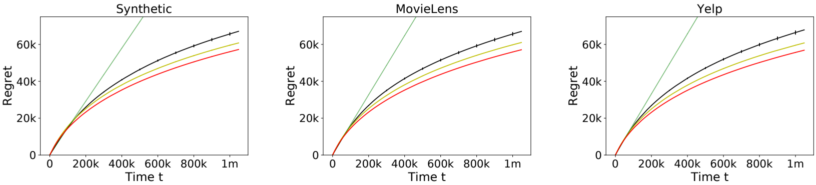
Improved Algorithm on Clustering Bandits [LCLL, IJCAI'2019]

- Arbitrary frequency distribution over users (compared to uniform distribution)
- Prove a regret bound that is free of the minimal frequency over users

$$R(T) = O\left(d\sqrt{mT}\ln(T) + \left(\frac{1}{\gamma_p^2} + \frac{n_u}{\gamma^2\lambda_x^3}\right)\ln(T)\right)$$

(compared to $R(T) = O\left(d\sqrt{mT}\ln(T) + \frac{1}{p_{\min}\gamma^2\lambda_x^3}\ln(T)\right)$)
where n_u is number of users and m is number of clusters

- Experiments —Ours —CLUB —LinUCB-One —LinUCB-Ind



Contents

Dependent Click Model (DCM)

- Allow multiple clicks
- Assumes there is a probability of satisfaction after each click

Dependent Click Model (DCM)

- Allow multiple clicks
- Assumes there is a probability of satisfaction after each click
- $r(A) = 1 - \prod_{k=1}^{K}(1 - \alpha(a_k)\gamma_k)$
  - γ_k: satisfaction probability after click on position k

Dependent Click Model (DCM)

- Allow multiple clicks
- Assumes there is a probability of satisfaction after each click
- $r(A) = 1 - \prod_{k=1}^{K}(1 - \alpha(a_k)\gamma_k)$
  - γ_k: satisfaction probability after click on position k
- The meaning of received feedback $(0, 1, 0, 1, 0)$

✗no click

✓click, not satisfied

✗no click

✓click, satisfied?

?

Dependent Click Model (DCM)

- Allow multiple clicks
- Assumes there is a probability of satisfaction after each click
- $r(A) = 1 - \prod_{k=1}^{K}(1 - \alpha(a_k)\gamma_k)$
  - γ_k: satisfaction probability after click on position k
- The meaning of received feedback $(0, 1, 0, 1, 0)$

✗no click

✓click, not satisfied

✗no click

✓click, satisfied?

?

	Context	Click Model	Regret
[KSWA, 2015]	-	CM	$O(\frac{L}{\Delta}\log(T))$
[LWZC, ICML'2016]	Linear	CM	$O(\frac{d}{p^*}\sqrt{TK}\log(T))$
[LZ, AAAI'2018]	GL	CM	$O(d\sqrt{TK}\log(T))$
[KKSW, 2016]	-	DCM	$O(\frac{L}{\Delta}\log(T))$
[LLZ, COCOON'2018]	GL	DCM	$O(dK\sqrt{TK}\log(T))$

Contents

Position-Based Model (PBM)

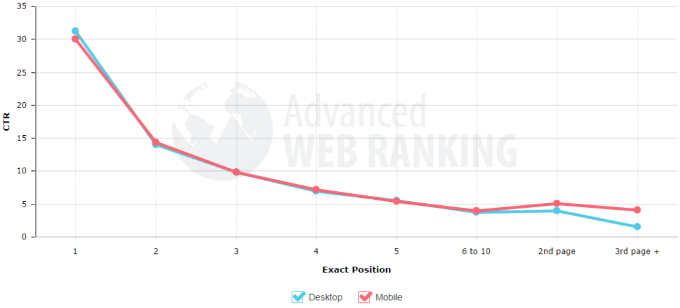
- Most popular model in industry

Position-Based Model (PBM)

- Most popular model in industry
- Assumes the user click probability on an item a of position k can be factored into $\beta_k \cdot \alpha(a)$

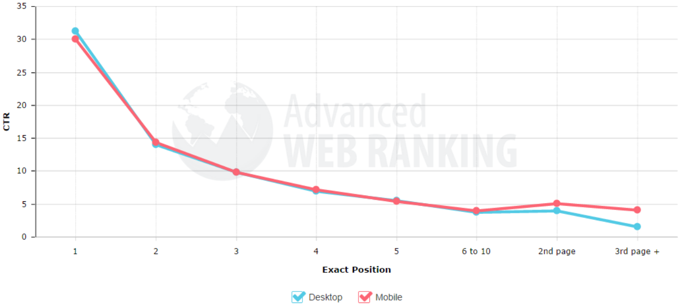
Position-Based Model (PBM)

- Most popular model in industry
- Assumes the user click probability on an item a of position k can be factored into $\beta_k \cdot \alpha(a)$
- β_k is position bias. Usually $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_K$



Position-Based Model (PBM)

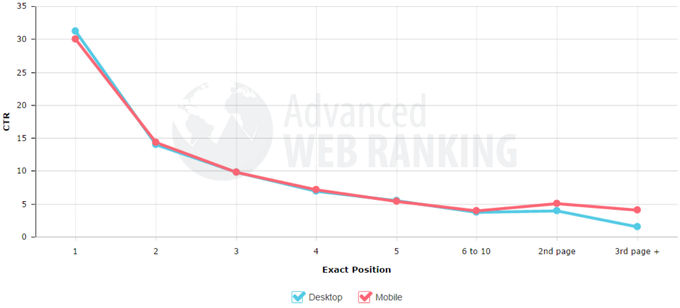
- Most popular model in industry
- Assumes the user click probability on an item a of position k can be factored into $\beta_k \cdot \alpha(a)$
- β_k is position bias. Usually $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_K$



- $r(A) = \sum_{k=1}^{K} \beta_k \alpha(a_k)$

Position-Based Model (PBM)

- Most popular model in industry
- Assumes the user click probability on an item a of position k can be factored into $\beta_k \cdot \alpha(a)$
- β_k is position bias. Usually $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_K$



- $r(A) = \sum_{k=1}^{K} \beta_k \alpha(a_k)$
- The meaning of received feedback $(0, 1, 0, 1, 0)$

Summary on Bandits with Click Models

	Context	Click Model	Regret
[KSWA, 2015]	-	CM	$O(\frac{L}{\Delta}\log(T))$
[LWZC, ICML'2016]	Linear	CM	$O(\frac{d}{p^*}\sqrt{TK}\log(T))$
[LZ, AAAI'2018]	GL	CM	$O(d\sqrt{TK}\log(T))$
[KKSW, 2016]	-	DCM	$O(\frac{L}{\Delta}\log(T))$
[LLZ, COCOON'2018]	GL	DCM	$O(dK\sqrt{TK}\log(T))$
[LVC, 2016]	-	PBM with β	$O(\frac{L}{\Delta}\log(T))$

Contents

General Click Models

- Common observations for click models
  - The click-through-rate (CTR) of list A on position k can be factored into

$$\mathrm{CTR}(A, k) = \chi(A, k)\ \alpha(a_k)$$

$\chi(A, k)$ is the examination probability of list A on position k

General Click Models

- Common observations for click models
  - The click-through-rate (CTR) of list A on position k can be factored into

$$\text{CTR}(A, k) = \chi(A, k)\ \alpha(a_k)$$

    $\chi(A, k)$ is the examination probability of list A on position k
  - E.g. $\chi(A, k) = \prod_{i=1}^{k-1}(1 - \alpha(a_i))$ in Cascade Model and $\chi(A, k) = \beta_k$ in Position Based Model

General Click Models

- Common observations for click models
  - The click-through-rate (CTR) of list A on position k can be factored into

$$\mathrm{CTR}(A, k) = \chi(A, k)\ \alpha(a_k)$$

    $\chi(A, k)$ is the examination probability of list A on position k
  - E.g. $\chi(A, k) = \prod_{i=1}^{k-1}(1 - \alpha(a_i))$ in Cascade Model and $\chi(A, k) = \beta_k$ in Position Based Model
- Difficulties on General Click Models
  - χ depends on both click models and lists

Summary on Bandits with Click Models

	Context	Click Model	Regret
[KSWA, 2015]	-	CM	$O(\frac{L}{\Delta}\log(T))$
[LWZC, ICML'2016]	Linear	CM	$O(\frac{d}{p^*}\sqrt{TK}\log(T))$
[LZ, AAAI'2018]	GL	CM	$O(d\sqrt{TK}\log(T))$
[KKSW, 2016]	-	DCM	$O(\frac{L}{\Delta}\log(T))$
[LLZ, COCOON'2018]	GL	DCM	$O(dK\sqrt{TK}\log(T))$
[LVC, 2016]	-	PBM with β	$O(\frac{L}{\Delta}\log(T))$
[ZTGKSW, 2017]	-	General	$O(\frac{K^3L}{\Delta}\log(T))$
[LKLS, NIPS'2018]	-	General	$O\left(\frac{KL}{\Delta}\log(T)\right)$ $O\left(\sqrt{K^3LT\log(T)}\right)$ $\Omega\left(\sqrt{KLT}\right)$

Online Learning to Rank with Features [LLS, ICML'2019] – Preparation

Recall

- Each item a is represented by a feature vector $x_a \in \mathbb{R}^d$
- The attractiveness of item a is $\alpha(a) = \theta^\top x_a$

Online Learning to Rank with Features [LLS, ICML'2019] – Preparation

Recall

- Each item a is represented by a feature vector $x_a \in \mathbb{R}^d$
- The attractiveness of item a is $\alpha(a) = \theta^\top x_a$

We bring up an algorithm called RecurRank (Recursive Ranking)

Online Learning to Rank with Features [LLS, ICML'2019] – Preparation

Recall

- Each item a is represented by a feature vector $x_a \in \mathbb{R}^d$
- The attractiveness of item a is $\alpha(a) = \theta^\top x_a$

We bring up an algorithm called RecurRank (Recursive Ranking)

- G-optimal design
  - Minimize the covariance of the least-squares estimator

Online Learning to Rank with Features [LLS, ICML'2019] – Preparation

Recall

- Each item a is represented by a feature vector $x_a \in \mathbb{R}^d$
- The attractiveness of item a is $\alpha(a) = \theta^\top x_a$

We bring up an algorithm called RecurRank (Recursive Ranking)

- G-optimal design
  - Minimize the covariance of the least-squares estimator
  - $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$

Online Learning to Rank with Features [LLS, ICML'2019] – Preparation

Recall

- Each item a is represented by a feature vector $x_a \in \mathbb{R}^d$
- The attractiveness of item a is $\alpha(a) = \theta^\top x_a$

We bring up an algorithm called RecurRank (Recursive Ranking)

- G-optimal design
    - Minimize the covariance of the least-squares estimator
    - $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$
    - For any distribution $\pi : X \to [0, 1]$, let $Q(\pi) = \sum_{x \in X} \pi(x) x x^\top$

Online Learning to Rank with Features [LLS, ICML'2019] – Preparation

Recall

- Each item a is represented by a feature vector $x_a \in \mathbb{R}^d$
- The attractiveness of item a is $\alpha(a) = \theta^\top x_a$

We bring up an algorithm called RecurRank (Recursive Ranking)

- G-optimal design
  - Minimize the covariance of the least-squares estimator
  - $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$
  - For any distribution $\pi : X \to [0,1]$, let $Q(\pi) = \sum_{x \in X} \pi(x) x x^\top$
  - By the Kiefer–Wolfowitz theorem there exists a π called the G-optimal design such that

$$\max \det(Q(\pi)) \text{ or equivalently } \max_{x \in X} \|x\|^2_{Q(\pi)^\dagger} \leq d$$

Online Learning to Rank with Features [LLS, ICML'2019] – Preparation

Recall

- Each item a is represented by a feature vector $x_a \in \mathbb{R}^d$
- The attractiveness of item a is $\alpha(a) = \theta^\top x_a$

We bring up an algorithm called RecurRank (Recursive Ranking)

- G-optimal design
  - Minimize the covariance of the least-squares estimator
  - $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$
  - For any distribution $\pi : X \to [0,1]$, let $Q(\pi) = \sum_{x \in X} \pi(x) x x^\top$
  - By the Kiefer–Wolfowitz theorem there exists a π called the G-optimal design such that

$$\max \det(Q(\pi)) \text{ or equivalently } \max_{x \in X} \|x\|^2_{Q(\pi)^\dagger} \le d$$

  - John's theorem implies that π may be chosen so that $|\{x : \pi(x) > 0\}| \le d(d+3)/2$

Online Learning to Rank with Features [LLS, ICML'2019] – Algorithm

- RecurRank Algorithm

Online Learning to Rank with Features [LLS, ICML'2019] – Algorithm

- RecurRank Algorithm
  - Each instantiation is called with three arguments:
    1. A phase number $\ell \in \{1, 2, \ldots\}$;
    2. An ordered tuple of items $\mathcal{A} = (a_1, a_2, \ldots, a_n)$;
    3. A tuple of positions $\mathcal{K} = (k, \ldots, k + m - 1)$ and $m \leq n$.

Online Learning to Rank with Features [LLS, ICML'2019] – Algorithm

- RecurRank Algorithm
  - Each instantiation is called with three arguments:
    1. A phase number $\ell \in \{1, 2, \dots\}$;
    2. An ordered tuple of items $\mathcal{A} = (a_1, a_2, \dots, a_n)$;
    3. A tuple of positions $\mathcal{K} = (k, \dots, k + m - 1)$ and $m \leq n$.
  - The algorithm is first called with $\ell = 1$, a random order over all items $\{1, \dots, L\}$, and $\mathcal{K} = (1, \dots, K)$

Online Learning to Rank with Features [LLS, ICML'2019] – Algorithm

- RecurRank Algorithm
  - Each instantiation is called with three arguments:
    1. A phase number $\ell \in \{1, 2, \ldots\}$;
    2. An ordered tuple of items $\mathcal{A} = (a_1, a_2, \ldots, a_n)$;
    3. A tuple of positions $\mathcal{K} = (k, \ldots, k + m - 1)$ and $m \leq n$.
  - The algorithm is first called with $\ell = 1$, a random order over all items $\{1, \ldots, L\}$, and $\mathcal{K} = (1, \ldots, K)$
  - Find a G-optimal design $\pi = \textsc{Gopt}(\mathcal{A})$. Then compute

$$T(a) = \left\lceil \frac{d\,\pi(a)}{2\Delta_\ell^2} \log\left(\frac{|\mathcal{A}|}{\delta_\ell}\right) \right\rceil, \qquad \Delta_\ell = 2^{-\ell}$$

Online Learning to Rank with Features [LLS, ICML'2019] – Algorithm

- RecurRank Algorithm
  - Each instantiation is called with three arguments:
    1. A phase number $\ell \in \{1, 2, \ldots\}$;
    2. An ordered tuple of items $\mathcal{A} = (a_1, a_2, \ldots, a_n)$;
    3. A tuple of positions $\mathcal{K} = (k, \ldots, k + m - 1)$ and $m \leq n$.
  - The algorithm is first called with $\ell = 1$, a random order over all items $\{1, \ldots, L\}$, and $\mathcal{K} = (1, \ldots, K)$
  - Find a G-optimal design $\pi = \textsc{Gopt}(\mathcal{A})$. Then compute

$$T(a) = \left\lceil \frac{d\,\pi(a)}{2\Delta_\ell^2} \log\left(\frac{|\mathcal{A}|}{\delta_\ell}\right) \right\rceil, \qquad \Delta_\ell = 2^{-\ell}$$

Hope to satisfy $|\alpha(a) - \hat{\alpha}(a)| \leq \Delta_\ell$ for any $a \in \mathcal{A}$ by the end of this instantiation

Online Learning to Rank with Features [LLS, ICML'2019] – Algorithm

- RecurRank Algorithm
  - Each instantiation is called with three arguments:
    1. A phase number $\ell \in \{1, 2, \ldots\}$;
    2. An ordered tuple of items $\mathcal{A} = (a_1, a_2, \ldots, a_n)$;
    3. A tuple of positions $\mathcal{K} = (k, \ldots, k + m - 1)$ and $m \leq n$.
  - The algorithm is first called with $\ell = 1$, a random order over all items $\{1, \ldots, L\}$, and $\mathcal{K} = (1, \ldots, K)$
  - Find a G-optimal design $\pi = \text{GOPT}(\mathcal{A})$. Then compute

$$T(a) = \left\lceil \frac{d\,\pi(a)}{2\Delta_\ell^2} \log\left(\frac{|\mathcal{A}|}{\delta_\ell}\right) \right\rceil, \qquad \Delta_\ell = 2^{-\ell}$$

    Hope to satisfy $|\alpha(a) - \hat{\alpha}(a)| \leq \Delta_\ell$ for any $a \in \mathcal{A}$ by the end of this instantiation
  - This instantiation runs for $\sum_{a \in \mathcal{A}} T(a)$ times

Online Learning to Rank with Features [LLS, ICML'2019] – Algorithm (Continued)

- RecurRank Algorithm (Continued)
  - Select each item $a \in \mathcal{A}$ exactly $T(a)$ times at position k and put the first $m-1$ items in $\mathcal{A} \setminus \{a\}$ at remaining positions $\{k+1, \ldots, k+m-1\}$
    first position — exploration
    remaining positions — exploitation
    only first position has the same examination probability χ for all lists

Online Learning to Rank with Features [LLS, ICML'2019] – Algorithm (Continued)

- RecurRank Algorithm (Continued)
  - Select each item $a \in \mathcal{A}$ exactly $T(a)$ times at position k and put the first $m-1$ items in $\mathcal{A} \setminus \{a\}$ at remaining positions $\{k+1, \ldots, k+m-1\}$
    first position — exploration
    remaining positions — exploitation
    only first position has the same examination probability χ for all lists
  - E.g. Suppose we have computed $T(a_3) = 100$, then it puts $(a_3, a_1, a_2, a_4, \ldots, a_m)$ on positions $(k, \ldots, k+m-1)$ for 100 rounds

Online Learning to Rank with Features [LLS, ICML'2019] – Algorithm (Continued)

- RecurRank Algorithm (Continued)
  - Select each item $a \in \mathcal{A}$ exactly $T(a)$ times at position k and put the first $m-1$ items in $\mathcal{A} \setminus \{a\}$ at remaining positions $\{k+1, \ldots, k+m-1\}$
    first position — exploration
    remaining positions — exploitation
    only first position has the same examination probability χ for all lists
  - E.g. Suppose we have computed $T(a_3) = 100$, then it puts $(a_3, a_1, a_2, a_4, \ldots, a_m)$ on positions $(k, \ldots, k+m-1)$ for 100 rounds
  - Compute $\hat{\theta}$ only using the feedbacks from first position k and rank items in decreasing order of the estimated attractiveness

$$\hat{\alpha}(\hat{a}_1) \geq \hat{\alpha}(\hat{a}_2) \geq \hat{\alpha}(\hat{a}_3) \geq \cdots \geq \hat{\alpha}(\hat{a}_n)$$

Online Learning to Rank with Features [LLS, ICML'2019] – Algorithm (Continued)

- RecurRank Algorithm (Continued)
  - Eliminate bad arms $\hat{a}_{n'+1}, \ldots, \hat{a}_n$ if

$$\hat{\alpha}(\hat{a}_1) \geq \cdots \geq \underbrace{\hat{\alpha}(\hat{a}_m) \geq \cdots \geq \hat{\alpha}(\hat{a}_{n'}) \geq \hat{\alpha}(\hat{a}_{n'+1})}_{\text{gap } \geq 2\Delta_\ell} \geq \cdots \geq \hat{\alpha}(\hat{a}_n)$$

Online Learning to Rank with Features [LLS, ICML'2019] – Algorithm (Continued)

- RecurRank Algorithm (Continued)
  - Eliminate bad arms $\hat{a}_{n'+1}, \dots, \hat{a}_n$ if

$$\hat{\alpha}(\hat{a}_1) \geq \cdots \geq \underbrace{\hat{\alpha}(\hat{a}_m) \geq \cdots \geq \hat{\alpha}(\hat{a}_{n'}) \geq \hat{\alpha}(\hat{a}_{n'+1})}_{\text{gap } \geq 2\Delta_\ell} \geq \cdots \geq \hat{\alpha}(\hat{a}_n)$$

  - Split the partition for each consecutive gap larger than $2\Delta_\ell$

$$\hat{\alpha}(\hat{a}_1) \geq \cdots \geq \underbrace{\hat{\alpha}(\hat{a}_{k_1}) \,\Big|\, \hat{\alpha}(\hat{a}_{k_1+1})}_{\text{gap } \geq 2\Delta_\ell} \geq \cdots \geq \underbrace{\hat{\alpha}(\hat{a}_{k_2}) \,\Big|\, \hat{\alpha}(\hat{a}_{k_2+1})}_{\text{gap } \geq 2\Delta_\ell} \geq \cdots \geq \hat{\alpha}(\hat{a}_{n'})$$

$$k, \cdots, k+k_1-1 \,\Big|\, k+k_1, \cdots, k+k_2-1 \,\Big|\, k+k_2, \cdots, k+m-1$$

Online Learning to Rank with Features [LLS, ICML'2019] – Algorithm (Continued)

- RecurRank Algorithm (Continued)
  - Eliminate bad arms $\hat{a}_{n'+1}, \ldots, \hat{a}_n$ if

$$\hat{\alpha}(\hat{a}_1) \geq \cdots \geq \underbrace{\hat{\alpha}(\hat{a}_m) \geq \cdots \geq \hat{\alpha}(\hat{a}_{n'}) \geq \hat{\alpha}(\hat{a}_{n'+1})}_{\text{gap } \geq 2\Delta_\ell} \geq \cdots \geq \hat{\alpha}(\hat{a}_n)$$

  - Split the partition for each consecutive gap larger than $2\Delta_\ell$

$$\hat{\alpha}(\hat{a}_1) \geq \cdots \geq \underbrace{\hat{\alpha}(\hat{a}_{k_1}) \,\Big|\, \hat{\alpha}(\hat{a}_{k_1+1})}_{\text{gap } \geq 2\Delta_\ell} \geq \cdots \geq \underbrace{\hat{\alpha}(\hat{a}_{k_2}) \,\Big|\, \hat{\alpha}(\hat{a}_{k_2+1})}_{\text{gap } \geq 2\Delta_\ell} \geq \cdots \geq \hat{\alpha}(\hat{a}_{n'})$$

$$k, \cdots, k + k_1 - 1 \,\Big|\, k + k_1, \cdots, k + k_2 - 1 \,\Big|\, k + k_2, \cdots, k + m - 1$$

  - Call the refined partitions with phase $\ell + 1$

Online Learning to Rank with Features [LLS, ICML'2019] – Results

- Regret bound

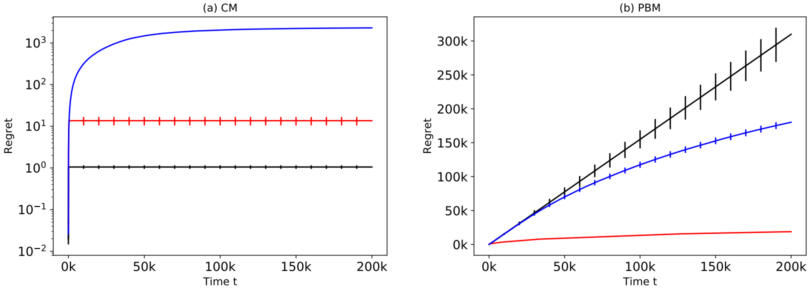
$$R(T) = O(K\sqrt{dT\log(LT)})$$

Online Learning to Rank with Features [LLS, ICML'2019] – Results

- Regret bound

$$R(T) = O(K\sqrt{dT\log(LT)})$$

- Experiments —RecurRank(Ours) —C^3-UCB —TopRank



Summary on Bandits with Click Models

	Context	Click Model	Regret
[KSWA, 2015]	-	CM	$O(\frac{L}{\Delta}\log(T))$
[LWZC, ICML'2016]	Linear	CM	$O(\frac{d}{p^*}\sqrt{TK}\log(T))$
[LZ, AAAI'2018]	GL	CM	$O(d\sqrt{TK}\log(T))$
[KKSW, 2016]	-	DCM	$O(\frac{L}{\Delta}\log(T))$
[LLZ, COCOON'2018]	GL	DCM	$O(dK\sqrt{TK}\log(T))$
[LVC, 2016]	-	PBM with β	$O(\frac{L}{\Delta}\log(T))$
[ZTGKSW, 2017]	-	General	$O(\frac{K^3 L}{\Delta}\log(T))$
[LKLS, NIPS'2018]	-	General	$O\left(\frac{KL}{\Delta}\log(T)\right)$ $O\left(\sqrt{K^3 LT\log(T)}\right)$ $\Omega\left(\sqrt{KLT}\right)$
[LLS, ICML'2019]	Linear	General	$O(K\sqrt{dT\log(LT)})$

Outline

Offline Evaluations

- Motivation
  - Can we estimate the expected number of clicks of new policies without directly employing it?

Offline Evaluations

- Motivation
  - Can we estimate the expected number of clicks of new policies without directly employing it?
- Offline Evaluation!

Offline Evaluations

- Motivation
  - Can we estimate the expected number of clicks of new policies without directly employing it?
- Offline Evaluation!
- Objective:
  - To design statistically efficient estimators based on logged dataset for any ranking policy

Offline Evaluations

- Motivation
  - Can we estimate the expected number of clicks of new policies without directly employing it?
- Offline Evaluation!
- Objective:
  - To design statistically efficient estimators based on logged dataset for any ranking policy
- Challenge:
  - The number of different lists is exponential in K

Offline Evaluation of Ranking Policies with Click Models [LAKMVW, KDD'2018]– Results

- We design estimators for different click models
  - Item-Position, Random, Rank-Based, Position-Based, Document-Based

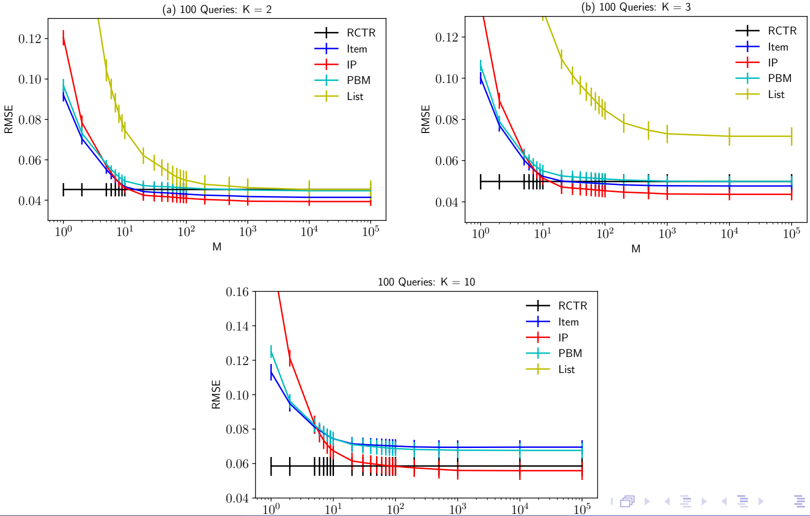
Offline Evaluation of Ranking Policies with Click Models [LAKMVW, KDD'2018]– Results

- We design estimators for different click models
  - Item-Position, Random, Rank-Based, Position-Based, Document-Based
- We prove that our estimators
  - are unbiased in a larger class of policies
  - have lower bias
  - the best policy have better theoretical guarantees

  than the existing unstructured estimators under the corresponding click model assumptions

Offline Evaluation of Ranking Policies with Click Models [LAKMVW, KDD'2018] – Experiments

Experiments – 100 most frequent queries in Yandex dataset



Outline

Conclusions

- Context + Cascade model (CM) / Dependent click model (DCM)
- Online clustering of bandits + Cascade model (CM)
- Improved algorithm on clustering of bandits
- Context + General click model
- Offline evaluation of ranking policies with click models

Publications

First-author papers in thesis – in the order of thesis

1. Shuai Li, Baoxiang Wang, Shengyu Zhang, Wei Chen, *Contextual Combinatorial Cascading Bandits*, ICML, 2016
2. Shuai Li, Shengyu Zhang, *Online Clustering of Contextual Cascading Bandits*, AAAI, 2018
3. Shuai Li, Wei Chen, S Li, Kwong-Sak Leung, *Improved Algorithm on Clustering of Bandits*, IJCAI 2019
4. Shuai Li, Tor Lattimore, Csaba Szepesvari, *Online Learning to Rank with Features*, ICML, 2019
5. Shuai Li, Yasin Abbasi-Yadkori, Branislav Kveton, S. Muthukrishnan, Vishwa Vinay and Zheng Wen, *Offline Evaluation of Ranking Policies with Click Models*, KDD, 2018

Publications

Mentioned co-authored papers

6. Weiwen Liu, Shuai Li, Shengyu Zhang, *Contextual Dependent Click Bandit Algorithm for Web Recommendation*, COCOON, 2018
7. Tor Lattimore, Branislav Kveton, Shuai Li, Csaba Szepesvari, TopRank: A Practical Algorithm for Online Stochastic Ranking, NeurIPS, 2018

Other co-authored papers

8. Pengfei Liu, Hongjian Li, Shuai Li, Kwong-Sak Leung, *Improving Prediction of Phenotypic Drug Response on Cancer Cell Lines Using Deep Convolutional Network*, BMC Bioinformatics, 2019
9. Ran Wang, Shuai Li, Man-Hon Wong, and Kwong-Sak Leung, *Drug-Protein-Disease Association Prediction and Drug Repositioning Based on Tensor Decomposition*, BIBM, 2018
10. Pengfei Liu, Shuai Li, Weiying Yi, Kwong-Sak Leung, *A Hybrid Distributed Framework for SNP Selections*, PDPTA, 2016

Publications

In submission

11. Shuai Li, Wei Chen, Zheng Wen, Kwong-Sak Leung, *Stochastic Online Learning with Probabilistic Feedback Graph*
12. Shuai Li, Kwong-Sak Leung, *Generalized Clustering Bandits*
13. Shuai Li, Tong Yu, Ole Mengshoel, Kwong-Sak Leung, *Online Semi-Supervised Learning with Large Margin Separation*
14. Xiaojin Zhang, Shuai Li, Shengyu Zhang, *Contextual Combinatorial Conservative Bandits*
15. Pengfei Liu, Shuai Li, Kwong-Sak Leung, *The Recovery of Stochastic Differential Equations with Genetic Programming and Kullback-Leibler Divergence*

Thank you!

&

Questions?

References I

P. Auer, N. Cesa-Bianchi, and P. Fischer.
Finite-time analysis of the multiarmed bandit problem.
Machine learning, 47(2-3):235–256, 2002.

S. Katariya, B. Kveton, C. Szepesvari, and Z. Wen.
Dcm bandits: Learning to rank with multiple clicks.
In *International Conference on Machine Learning*, pages 1215–1224, 2016.

B. Kveton, C. Szepesvari, Z. Wen, and A. Ashkan.
Cascading bandits: Learning to rank in the cascade model.
In *International Conference on Machine Learning*, pages 767–776, 2015.

References II

P. Lagrée, C. Vernade, and O. Cappe.
Multiple-play bandits in the position-based model.
In *Advances in Neural Information Processing Systems*, pages 1597–1605, 2016.

T. Lattimore, B. Kveton, Li, Shuai, and C. Szepesvari.
Toprank: A practical algorithm for online stochastic ranking.
In *The Conference on Neural Information Processing Systems*, 2018.

W. Liu, Li, Shuai, and S. Zhang.
Contextual dependent click bandit algorithm for web recommendation.
In *International Computing and Combinatorics Conference*, pages 39–50. Springer, 2018.

References III

Li, Shuai, Y. Abbasi-Yadkori, B. Kveton, S. Muthukrishnan, V. Vinay, and Z. Wen.
Offline evaluation of ranking policies with click models.
In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2018.

Li, Shuai, W. Chen, S. Li, and K.-S. Leung.
Improved algorithm on online clustering of bandits.
In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

Li, Shuai, T. Lattimore, and C. Szepesvári.
Online learning to rank with features.
In *International Conference on Machine Learning (ICML)*, 2019.

References IV

Li, Shuai, B. Wang, S. Zhang, and W. Chen.
Contextual combinatorial cascading bandits.
In *International Conference on Machine Learning*, pages 1245–1253, 2016.

Li, Shuai and S. Zhang.
Online clustering of contextual cascading bandits.
In *The AAAI Conference on Artificial Intelligence*, 2018.

M. Zoghi, T. Tunys, M. Ghavamzadeh, B. Kveton, C. Szepesvari, and Z. Wen.
Online learning to rank in stochastic click models.
In *International Conference on Machine Learning*, pages 4199–4208, 2017.

References V

S. Zong, H. Ni, K. Sung, N. R. Ke, Z. Wen, and B. Kveton.
Cascading bandits for large-scale recommendation problems.
In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pages 835–844. AUAI Press, 2016.

A Key Part Proof for CLUB-cascade (Improving C^3-UCB)

$$
\begin{aligned}
&\mathbb{E}_t[R(\boldsymbol{A}_t, \boldsymbol{y}_t)] \\
=&\mathbb{E}_t\left[\left(1-\prod_{k=1}^{K}(1-\boldsymbol{y}_t(x^*_{t,k}))\right)-\left(1-\prod_{k=1}^{K}(1-\boldsymbol{y}_t(\boldsymbol{x}_{t,k}))\right)\right] \\
=&\mathbb{E}_t\left[\prod_{k=1}^{K}(1-\boldsymbol{y}_t(\boldsymbol{x}_{t,k}))-\prod_{k=1}^{K}(1-\boldsymbol{y}_t(x^*_{t,k}))\right] \\
=&\mathbb{E}_t\left[\sum_{k=1}^{K}\left(\prod_{\ell=1}^{k-1}(1-\boldsymbol{y}_t(\boldsymbol{x}_{t,\ell}))\right)\left[(1-\boldsymbol{y}_t(\boldsymbol{x}_{t,k}))-(1-\boldsymbol{y}_t(x^*_{t,k}))\right]\left(\prod_{\ell=k+1}^{K}(1-\boldsymbol{y}_t(x^*_{t,\ell}))\right)\right] \\
\leq&\mathbb{E}_t\left[\sum_{k=1}^{K}\left(\prod_{\ell=1}^{k-1}(1-\boldsymbol{y}_t(\boldsymbol{x}_{t,\ell}))\right)[\boldsymbol{y}_t(x^*_{t,k})-\boldsymbol{y}_t(\boldsymbol{x}_{t,k})]\right] \\
=&\mathbb{E}_t\left[\sum_{k=1}^{\boldsymbol{K}_t}[\boldsymbol{y}_t(x^*_{t,k})-\boldsymbol{y}_t(\boldsymbol{x}_{t,k})]\right]
\end{aligned}
$$

Proof Sketch for RecurRank

- Use (ℓ, i) to represent the i-th call of RecurRank with $\ell, \mathcal{A}_{\ell i}, \mathcal{K}_{\ell i}$

Proof Sketch for RecurRank

- Use (ℓ, i) to represent the i-th call of RecurRank with $\ell, \mathcal{A}_{\ell i}, \mathcal{K}_{\ell i}$
- Prove with high probability for any (ℓ, i)
  - $a_k^* \in \mathcal{A}_{\ell i}$ if $k \in \mathcal{K}_{\ell i}$
  - $|\hat{\theta}_{\ell i}^\top x_a - \chi_{\ell i} \theta_*^\top x_a| \leq \Delta_\ell$, where $\chi_{\ell i}$ is the examination probability of the optimal list on the first position in $\mathcal{K}_{\ell i}$

Proof Sketch for RecurRank

- Use (ℓ, i) to represent the i-th call of RecurRank with $\ell, \mathcal{A}_{\ell i}, \mathcal{K}_{\ell i}$
- Prove with high probability for any (ℓ, i)
  - $a_k^* \in \mathcal{A}_{\ell i}$ if $k \in \mathcal{K}_{\ell i}$
  - $|\hat{\theta}_{\ell i}^\top x_a - \chi_{\ell i}\theta_*^\top x_a| \leq \Delta_\ell$, where $\chi_{\ell i}$ is the examination probability of the optimal list on the first position in $\mathcal{K}_{\ell i}$
- In (ℓ, i)th call, item a is put at position k, then
  - $\chi_{\ell i}\left(\alpha(a_k^*) - \alpha(a)\right) \leq 8|\mathcal{K}_{\ell i}|\Delta_\ell$ if k is the first position in $\mathcal{K}_{\ell i}$
  - $\chi_{\ell i}\left(\alpha(a_k^*) - \alpha(a)\right) \leq 4\Delta_\ell$ if k is the remaining position
  - thus $O(|\mathcal{K}_{\ell i}|\Delta_\ell)$ regret for this part