TopRank: A Practical Algorithm for Online Stochastic Ranking

Tor Lattimore, Branislav Kveton, Shuai Li, Csaba Szepesvári

Motivation

- Online learning to rank
  - A sequential decision-making problem
  - Recommends a list of items to user
  - Receives click feedback from user
- Common click models: Cascade, Position-Based Model, Document-Based, etc
- Existing works
  1. Either focus on a specific model and might perform poorly in different models
  2. Or assume a general model, but propose unnatural algorithms that discard lots of data

Setting

- L items, $K \leq L$ positions
- Action set $\mathcal{A} = \Pi([L])$
  - For each $a \in \mathcal{A}$, $a(k)$ is the item placed at the kth position
- In each round t
  - The learner chooses an action $A_t \in \mathcal{A}$
  - The learner observes click feedback $C_{t1}, \ldots, C_{tL}$
- Assume click probability on item $i = a(k)$ is given by

$$\mathbb{P}(C_{ti} = 1 \mid A_t = a) = v(a, k)$$

- The goal of the learner is to minimize the expected cumulative regret

$$R_n = \max_{a \in \mathcal{A}} \mathbb{E}\left[\sum_{t=1}^{n} \sum_{k=1}^{K} (v(a,k) - v(A_t,k))\right] .$$

Assumptions

Assumption 1. *$v(a,k) = 0$ for all $k > K$.*

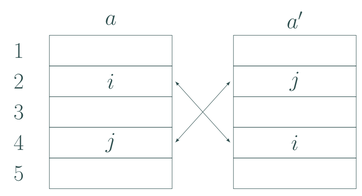
- There exists an unknown attractiveness function $\alpha : [L] \to [0,1]$
- An action a optimal if $\alpha(a(k)) = \max_{k' \geq k} \alpha(a(k'))$ for all $k \in [K]$

Assumption 2. *Let $a^* \in \mathcal{A}$ be an optimal action. Then $\max_{a \in \mathcal{A}} \sum_{k=1}^{K} v(a,k) = \sum_{k=1}^{K} v(a^*,k)$.*

Assumption 3. *Suppose $\alpha(i) \geq \alpha(j)$ and $\sigma : \mathcal{A} \to \mathcal{A}$ only exchanges i and j. Then $\forall a \in \mathcal{A}$,*

$$v(a, a^{-1}(i)) \geq \frac{\alpha(i)}{\alpha(j)} v(\sigma \circ a, a^{-1}(i)) .$$

Illustration Suppose $\alpha(i) \geq \alpha(j)$. Then $v(a,2) \geq v(a',2)$ and $v(a,4) \leq v(a',4)$.



Assumption 4. *For any action a and optimal action a^* with $\alpha(a(k)) = \alpha(a^*(k))$ it holds that $v(a,k) \geq v(a^*,k)$.*

Cascade Model Assume the user checks the items from 1st position and clicks and stops at first satisfying item:

$$v(a,k) = \left(\prod_{\ell=1}^{k-1} (1 - \alpha(a(\ell)))\right) \alpha(a(k))$$

Position-Based Model Suppose β_k's are (unknown) position examination probability.

$$v(a,k) = \beta_k \cdot \alpha(a(k))$$

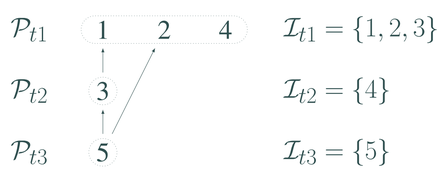
Algorithm

- Given relation $G \subseteq [L]^2$ and $X \subseteq [L]$, $\min_G(X) = \{i \in X : (i,j) \notin G \text{ for all } j \in X\}$.
- Let $\mathcal{A}(\mathcal{P}_1, \ldots, \mathcal{P}_d)$ be the set of actions a where the items in $\mathcal{P}_1$ are placed at the first $|\mathcal{P}_1|$ positions, the items in $\mathcal{P}_2$ are placed at the next $|\mathcal{P}_2|$ positions, and so on.

TopRank

1. $G_1 \leftarrow \emptyset$ and $c \leftarrow \frac{4\sqrt{2/\pi}}{\mathrm{erf}(\sqrt{2})}$
2. **for** $t = 1, \ldots, n$ **do**
3. $\quad d \leftarrow 0$
4. $\quad$ **while** $[L] \setminus \bigcup_{c=1}^{d} \mathcal{P}_{tc} \neq \emptyset$ **do**
5. $\quad\quad d \leftarrow d + 1$
6. $\quad\quad \mathcal{P}_{td} \leftarrow \min_{G_t}\left([L] \setminus \bigcup_{c=1}^{d-1} \mathcal{P}_{tc}\right)$
7. $\quad$ Choose A_t uniformly at random from $\mathcal{A}(\mathcal{P}_{t1}, \ldots, \mathcal{P}_{td})$
8. $\quad$ Observe click indicators $C_{ti} \in \{0,1\}$ for all $i \in [L]$
9. $\quad$ **for all** $(i,j) \in [L]^2$ **do**
10. $\quad\quad U_{tij} \leftarrow \begin{cases} C_{ti} - C_{tj} & \text{if } i,j \in \mathcal{P}_{td} \text{ for some } d \\ 0 & \text{otherwise} \end{cases}$
11. $\quad\quad S_{tij} \leftarrow \sum_{s=1}^{t} U_{sij}$ and $N_{tij} \leftarrow \sum_{s=1}^{t} |U_{sij}|$
12. $G_{t+1} \leftarrow G_t \cup \left\{(j,i) : S_{tij} \geq \sqrt{2N_{tij} \log\left(\frac{c}{\delta}\sqrt{N_{tij}}\right)} \text{ and } N_{tij} > 0\right\}$

Illustration Suppose $L = 5$ and $K = 4$ and the relation is $G_t = \{(3,1),(5,2),(5,3)\}$. In round t the first three positions in the ranking will contain items from $\mathcal{P}_{t1} = \{1,2,4\}$, but with random order. The fourth position will be item 3 and item 5 is not shown to the user.



Analysis

Theorem 1 (Upper Bound)**.** *The n-step regret of* `TopRank` *is bounded from above as*

$$R_n \leq O\left(\frac{KL}{\Delta}\log(n)\right) , \qquad R_n \leq O\left(\sqrt{K^3 L n \log(n)}\right) .$$

Theorem 2 (Lower Bound)**.** *Suppose that $L = NK$ with $n \geq K$ and $n \geq N \geq 8$. For any algorithm, there exists a ranking problem such that $\mathbb{E}[R_n] \geq \Omega\left(\sqrt{KLn}\right)$.*

Experiments

- Yandex dataset
  - 167 million search queries
  - In each query, the user is shown 10 documents and the search engine records the user clicks.
- We select 60 frequent search queries, and learn their CMs and PBMs using PyClick[1]
- **Goal:** Rerank $L = 10$ most attractive items to maximize CTR at the first $K = 5$ positions

- Figure 1: General trend on specific queries
  - `TopRank` typically outperforms `BatchRank`
  - Cascade Model
    - `CascadeKL-UCB` outperforms `TopRank`
  - Position-Based Model
    - `CascadeKL-UCB` learns very good policies in about two thirds of queries, but suffers linear regret for the rest
    - In many queries, `TopRank` outperforms `CascadeKL-UCB` in one million steps

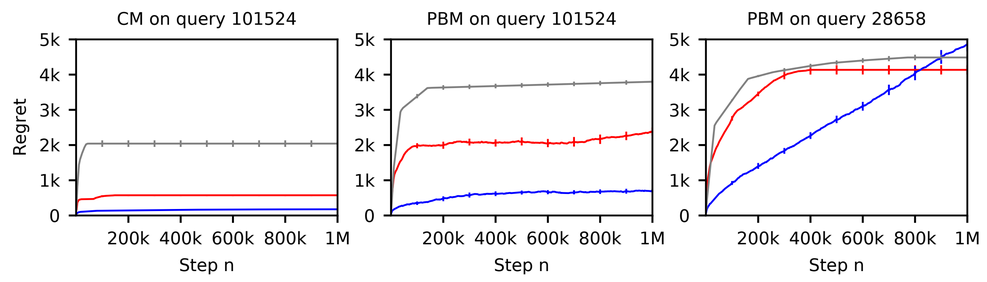


Figure 1: The n-step regret of `TopRank` (red), `CascadeKL-UCB` (blue), and `BatchRank` (gray) in three problems. The results are averaged over 10 runs. The error bars are the standard errors of our regret estimates.

- Figure 2: Average regret over all queries
  - Cascade Model
    - The regret of `CascadeKL-UCB` is about three times lower than that of `TopRank`
    - The regret of `TopRank` is about three times lower than that of `BatchRank`
  - Position-Based Model
    - The regret of `CascadeKL-UCB` is higher than that of `TopRank` after 4 million steps
    - The regret of `TopRank` is about 30% lower than that of `BatchRank`
  - `TopRank` improves over `BatchRank` in both the cascade and position-based models
  - The worse performance of `TopRank` relative to `CascadeKL-UCB` in the cascade model is offset by its robustness to multiple click models

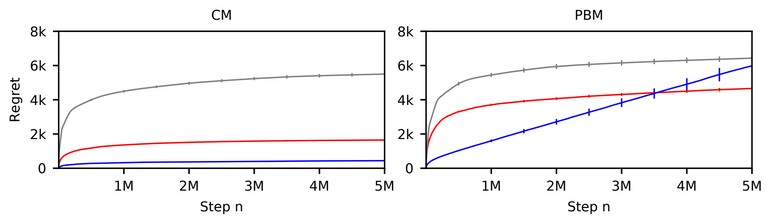


Figure 2: The n-step regret of `TopRank` (red), `CascadeKL-UCB` (blue), and `BatchRank` (gray) in two click models. The results are averaged over 60 queries and 10 runs per query. The error bars are the standard errors of our regret estimates.

Conclusions

- Introduced a new click model for online ranking that subsumes common click models
  - With stronger regret guarantees
  - With an easier and more insightful proof
  - With improved empirical performance
- Improved on lower bound for combinatorial semi-bandits with m-sets
- Future work: Consider the large-scale contextual setting where items have feature vectors

References

[1] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. Click models for web search. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 7(3):1–115, 2015.

[2] Branislav Kveton, Csaba Szepesvari, Zheng Wen, and Azin Ashkan. Cascading bandits: Learning to rank in the cascade model. In *International Conference on Machine Learning*, pages 767–776, 2015.

[3] Masrour Zoghi, Tomas Tunys, Mohammad Ghavamzadeh, Branislav Kveton, Csaba Szepesvari, and Zheng Wen. Online learning to rank in stochastic click models. In *International Conference on Machine Learning*, pages 4199–4208, 2017.