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 ≤ L positions
- Action set A = {1, ..., K}
- For each a ∈ A, o(a) is the item placed at the ith position
- In each round t:
  - The learner chooses an action A_t ∈ A
  - The learner observes click feedback C_{t1}, ..., C_{tK}
- Assume click probability on item i = o(a) is given by
  \[ P(C_{ti} = 1 | A_t = a) = v_{ti} \]
- The goal of the learner is to minimize the expected cumulative regret
  \[ R_T = \max_{A ∈ \calA} \sum_{t=1}^T \sum_{k=1}^K (v_{tk} - v_{tk}) \]

Assumptions
Assumption 1. \( v(a, k) = 0 \) for all \( k > K \).
- There exists an unknown attractiveness function \( \alpha : [L] \rightarrow [0, 1] \).
- An action \( a \) is optimal if \( o(a)(k) = \max_{a'} o(a')(k) \) for all \( k \in [K] \).
Assumption 2. Let \( a^* \in A \) be an optimal action. Then \( \max_{k \in [K]} \sum_{a' \in A} v_{k} = \sum_{a' \in A} v(a', a^*) \).
Assumption 3. Suppose \( v(a, k) ≥ o(a) \) and \( \alpha \rightarrow A \) only exchanges i and j. Then \( \forall a \in A \),
\[ v(a, k) ≥ \sum_{i=1}^T (1 - o(\ell)) \alpha(i) \]

Illustration
Suppose \( a(k) ≥ o(k) \). Then \( v(a, 1) ≥ v(a, 2) \) and \( v(a, 0) ≤ v(a', k) \).

Algorithm
- Given relation \( G \subseteq [K]^2 \) and \( X \subseteq [L], \text{min}(X) = (i \in X, (i, j) \notin G \text{ for all } j \in X) \).
- Let \( \mathcal{A} = \{P_1, ..., P_{16}\} \) be the set of actions \( a \) where the items in \( P_i \) are placed at the first \( |P_i| \) positions, the items in \( P_2 \) are placed at the next \( |P_2| \) positions, and so on.

TopRank
1. \( G_t = \emptyset \) and \( v + \frac{\sqrt{\pi}}{2} \).
2. For \( t = 1, ..., \infty \) do
3. \( i = 0 \).
4. While \( [L] \setminus \bigcup_{t=1}^T P_t \neq \emptyset \) do
5. \( i = i + 1 \).
6. \( P_i \leftarrow \min_{C_{i}} \{G_t \setminus \bigcup_{t=1}^T P_t\} \).
7. Choose \( A_t \) uniformly at random from \( \mathcal{A} \).
8. Observe click indicators \( C_{t1}, ..., C_{tK} \).
9. For all \( (i, j) \notin G_t \) do
10. \( U_{ij} \leftarrow \sum_{j \in X} S_{ij} \) if \( j \in X \) for some \( d \) otherwise
11. \( S_{ij} = \sum_{j \in X} S_{ij} \) and \( N_{ij} = \sum_{j \in X} U_{ij} \).
12. \( \alpha_{ij} = \frac{G_{ij}}{\sqrt{\gamma N_{ij}}} \) and \( N_{ij} > 0 \).

Illustration
Suppose \( L = 5 \) and \( K = 4 \) and the relation \( G_t = \{1, 1, 5, 2, 3\} \).
To find the first three positions in the ranking will contain items from \( P_1 = \{1, 2, 4\} \), but with random order. The fourth position will be item 3 and item 5 is not shown to the user.

References