Online Learning to Rank with Features

Authors: Shuai Li, Tor Lattimore, Csaba Szepesvári

The Chinese University of Hong Kong
DeepMind
University of Alberta
Amazon, YouTube, Facebook, Netflix, Taobao
There are \( L \) items and \( K \leq L \) positions

At each time \( t = 1, 2, \ldots, \)

- Choose an ordered list \( A_t = (a^t_1, \ldots, a^t_K) \)
- Show the user the list
- Receive click feedback \( C_{t1}, \ldots, C_{tK} \in \{0, 1\} \), per position

Objective: Maximize the expected number of clicks

\[
\mathbb{E} \left[ \sum_{t=1}^{T} \sum_{k=1}^{K} C_{tk} \right]
\]
Click Models

• Click models describe how users interact with item lists

• Cascade Model (CM)
  • Assumes the user checks the list from position 1 to position \( K \), clicks at the first satisfying item and stops

• Dependent Click Model (DCM)
  • Further assumes there is a satisfaction probability after click

• Position-Based Model (PBM)
  • Assumes the user click probability on an item \( a \) of position \( k \) can be factored into item attractiveness and position bias

• Generic model
  • Make as few assumptions as possible about the click model
Each item $a$ is represented by a feature vector $x_a \in \mathbb{R}^d$

The attractiveness of item $a$ is $\alpha(a) = \theta^T x_a$

Click probability factors: $P_t (C_{ti} = 1) = \alpha(a^t_i) \chi(A_t, i)$ where $\chi$ is the examination probability, which satisfies reasonable assumptions

RecurRank (Recursive Ranking)

For each phase $\ell$

- Use first position for exploration
- Use remaining positions for exploitation, rank best items first

Split items and positions when the phase ends

Recursively call the algorithm with increased phase
Example

Instance 1

\[ \ell = 1 \]

1

1

8

\[ A \]

\[ a_1 \]

\[ a_8 \]

\[ \ldots \]

\[ a_{50} \]

Instance 1
Example

Instance 1

$\ell = 1$

$A$

$\begin{array}{l}
1 \\
\vdots \\
8 \\
\vdots \\
a_{50}
\end{array}$

Instance 2

$\ell = 2$

$A$

$\begin{array}{l}
1 \\
\vdots \\
8 \\
\vdots \\
a_{25}
\end{array}$

Instance 3

$\ell = 2$

$A$

$\begin{array}{l}
1 \\
\vdots \\
8 \\
\vdots \\
a_{25}
\end{array}$

Instance 4

$\ell = 3$

Instance 5

Instance 6
Example

Instance 1

$\ell = 1$

$A$

$1$

$a_1$

$\ldots$

$a_8$

$\ldots$

$a_{50}$

Instance 2

$\ell = 2$

$A$

$1$

$a_1$

$2$

$a_2$

$3$

$a_3$

Instance 3

$\ell = 2$

$A$

$4$

$a_4$

$8$

$a_8$

$\ldots$

$a_{25}$

Instance 4

$\ell = 3$

$A$

$1$

$a_1$

$2$

$a_2$

$3$

$a_3$

Instance 5

$\ell = 3$

$A$

$1$

$a_1$

$2$

$a_2$

$3$

$a_3$

$4$

$a_4$

$5$

$a_5$

$6$

$a_6$

$7$

$a_7$

$8$

$a_8$

$\ldots$

$a_{12}$

Instance 6

$\ell = 3$

$A$

$1$

$a_1$

$2$

$a_2$

$3$

$a_3$

$4$

$a_4$

$5$

$a_5$

$6$

$a_6$

$7$

$a_7$

$8$

$a_8$

$\ldots$

$a_{12}$
Example
Example
Results

- Regret bound

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

- Improves over existing bound \( O\left(\sqrt{K^3LT \log(T)}\right) \)
Results

- Regret bound

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

- Improves over existing bound \( O(\sqrt{K^3LT \log(T)}) \)
Thank you!
Sumeet Katariya, Branislav Kveton, Csaba Szepesvari, and Zheng Wen.

**Dcm bandits: Learning to rank with multiple clicks.**

Branislav Kveton, Csaba Szepesvari, Zheng Wen, and Azin Ashkan.

**Cascading bandits: Learning to rank in the cascade model.**

Paul Lagrée, Claire Vernade, and Olivier Cappe.

**Multiple-play bandits in the position-based model.**


Shi Zong, Hao Ni, Kenny Sung, Nan Rosemary Ke, Zheng Wen, and Branislav Kveton.

**Cascading bandits for large-scale recommendation problems.**