Online Learning to Rank: An Overview

Shuai Li

http://shuaili8.github.io/

July 15, 2021

DRL4IR Workshop @ SIGIR 2021
Motivation - Learning to Rank

- Amazon, YouTube, Facebook, Netflix, Taobao
Multi-armed Bandits
Bandits

- Five rounds to go. Which arm would you choose next?

<table>
<thead>
<tr>
<th>Time</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Left arm</strong></td>
<td>$1</td>
<td>$0</td>
<td>$1</td>
<td>$1</td>
<td>$0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Right arm</strong></td>
<td>$1</td>
<td>$0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Multi-armed Bandit Problem

• A special case of reinforcement learning

• There are $L$ arms
  • Each arm $a$ has an unknown reward distribution on $[0, 1]$ with unknown mean $\alpha(a)$
  • The best arm is $a^* = \text{argmax } \alpha(a)$
MAB Setting

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

• Objective:
  • Maximize the expected cumulative reward in $T$ rounds $\mathbb{E}[\sum_{t=1}^{T} \alpha(a_t)]$
  • Minimize the regret in $T$ rounds
    \[
    R(T) = T \cdot \alpha(a^*) - \mathbb{E}\left[\sum_{t=1}^{T} \alpha(a_t)\right]
    \]

• Balance the trade-off between exploration and exploitation
  • Exploitation: Select arms that yield good results so far
  • Exploration: Select arms that have not been tried much before
UCB – Upper Confidence Bound

• Principle: optimism in face of uncertainty
• UCB policy: Select

\[ a_t = \arg\max_a \hat{\alpha}_a + \sqrt{\frac{2 \log t}{T_a(t)}} \]

• Gap-dependent regret bound \( O(\frac{L}{\Delta} \log T) \) where \( \Delta = \min_{\alpha_a < \alpha^*} (\alpha^* - \alpha_a) \) is the minimal gap

• Gap-free bound \( O(\sqrt{LT \log T}) \) tight up to a factor of \( \sqrt{\log T} \)

Online Learning to Rank
Setting: Online Learning to Rank

- There are $L$ items
  - Each item $a$ with an unknown attractiveness $\alpha(a)$
- There are $K$ positions
- At each time $t$
  - The learning agent recommends a list of items $A_t = (a_1^t, a_2^t, ..., a_K^t)$
  - Receives the binary click feedback vector $C_t \in \{0,1\}^K$

- Objective: minimize the regret over $T$ rounds
  \[
  R(T) = T \cdot r(A^*) - \mathbb{E} \left[ \sum_{t=1}^{T} r(A_t) \right]
  \]
  where
  - $r(A)$ is the reward of list $A$
  - $A^* = (1,2, ..., K)$ by assuming arms are ordered by $\alpha(1) \geq \alpha(2) \geq \ldots \geq \alpha(L)$
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
  • There is at most 1 click
  • \( r(A) = 1 - \prod_{k=1}^{K} (1 - \alpha(a_k)) \)
  • The meaning of received feedback (0,0,1,0,0)

Key Point for Analysis

• The regret is defined on the whole list

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

• But the received feedback is partial and random

• A key lemma

\[ r(A^*; w_t) - r(A_t; w_t) \leq \sum_{k=1}^{K} \prod_{i=1}^{k-1} (1 - w_t(a_{t,i})) [w_t(a^*_{t,k}) - w_t(a_{t,k})] \]
Regret Bound

• For the cascade click model

\[ R(T) = O \left( \frac{L}{\Delta} \log T \right) \]

• Contextual cascading bandits
  • The click rate of each item is a linear form

\[ R(T) = O \left( d \sqrt{KT} \log T + e^K \right) \]
  • \( d \) is the feature dimension

Click Models – Dependent Click Model (DCM)

• Allow multiple clicks
• Assume there is a probability of satisfaction after each click
• \( r(A) = 1 - \prod_{k=1}^{K} (1 - \alpha(a_k) \gamma_k) \)
  • \( \gamma_k \): satisfaction probability after click on position \( k \)
• The meaning of received feedback \((0,1,0,1,0)\)
Click Models – 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 \( \beta_k \cdot \alpha(\alpha) \)
  - \( \beta_k \) is position bias. Usually \( \beta_1 \geq \beta_2 \geq \cdots \geq \beta_K \)

\[
r(A) = \sum_{k=1}^{K} \beta_k \cdot \alpha(a_k)
\]
- The meaning of received feedback (0,1,0,1,0)
## Bandit Works for Specific Click Models

<table>
<thead>
<tr>
<th>Context</th>
<th>Click Model</th>
<th>Regret</th>
</tr>
</thead>
<tbody>
<tr>
<td>[KSWA, 2015]</td>
<td>CM</td>
<td>$\Theta\left(\frac{L}{\Delta}\log T\right)$</td>
</tr>
<tr>
<td>[LWZC, ICML’2016][ZNSKWK, UAI’2016][LZ, AAAI’2018]</td>
<td>GL, CM</td>
<td>$O(d\sqrt{TK} \log T)$</td>
</tr>
<tr>
<td>[KKSW, 2016]</td>
<td>-</td>
<td>DCM</td>
</tr>
<tr>
<td>[LLZ, COCOON’2018]</td>
<td>GL</td>
<td>DCM</td>
</tr>
<tr>
<td>[LVC, 2016]</td>
<td>-</td>
<td>PBM with known $\beta$</td>
</tr>
</tbody>
</table>

General Click Models
Modeling Bias

- **CascadeKLUCB** is the best algorithm under Cascade Model, but suffers linear regret in the environment of Position-based Model.

- **TopRank** and **BatchRank** are two algorithms designed for general click model.

![Regret Graphs](image)

General Click Models

• Common observations for click models
  • The click-through-rate (CTR) of list $A$ on position $k$ can be factored as
    \[ \nu(A, k) = \text{CTR}(A, k) = \chi(A, k)\alpha(a_k) \]
    where $\chi(A, k)$ is the examination probability of list $A$ on position $k$.
  • $\chi(A, k) = \prod_{i=1}^{k-1} (1 - \alpha(a_i))$ in Cascade Model.
  • $\chi(A, k) = \beta_k$ in Position-based Model.

• Difficulties on General Click Models
  • $\chi$ depends on both click models and lists.
Assumptions

1. $v(A, k) = 0$ for all $k > K$

2. $A^* = (1, 2, ..., K)$ has the highest value $\sum_{k=1}^{K} v(A, k)$, where $\alpha(1) \geq \alpha(2) \geq \cdots \geq \alpha(L)$

3. Suppose $\alpha(i) \geq \alpha(j)$ and $\sigma: [L] \rightarrow [L]$ only exchanges $i$ and $j$. Then for any list $A$

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

- Illustration: $\alpha(i) \geq \alpha(j)$. Then $\chi(A, 2) \geq \chi(A', 2)$ and $\chi(A, 4) \leq \chi(A', 4)$

4. $\chi(A, k) \geq \chi(A^*, k)$

It can be checked that CM and PBM both satisfy these assumptions
TopRank [LKLS, NeurIPS’18]

- TopRank: Topological Ranking
- It maintains a set of order relationships between pairs of items: item $b$ is worse than item $a$

$$\mathcal{P}_1 \rightarrow 1 \rightarrow 2 \rightarrow 4 \quad \mathcal{K}_1 = \{1, 2, 3\}$$

$$\mathcal{P}_2 \rightarrow 3 \rightarrow 6 \quad \mathcal{K}_2 = \{4, 5\}$$

$$\mathcal{P}_3 \rightarrow 5 \quad \mathcal{K}_3 = \{6\}$$

e.g. $(2,1,4,3,6,5), (4,1,2,6,3,5)$
TopRank [LKLS, NeurIPS’18] 2

• TopRank ranks items randomly in each partition

• Based on the received click-or-not feedback, it is equivalent to draw a click difference $X_{ab}$ on $\{-1, 0, 1\}$ for each pair of items $(a, b)$ in the same partition
  • $X_{ab} = 1$ if $a$ is clicked but $b$ is not clicked
  • $\mathbb{E}[X_{ab} | X_{ab} \neq 0] \geq \frac{\alpha(a) - \alpha(b)}{\alpha(a) + \alpha(b)}$

• $b$ is worse than $a$ if $S_{ab} \geq \sqrt{2N_{ab} \log \left( \frac{c}{\delta} \sqrt{N_{ab}} \right)}$ and $N_{ab} > 0$
  • $S_{ab} = \sum_t X_{ab,t}$ is the sum of click difference in the same partition
  • $N_{ab} = \sum_t |X_{ab,t}|$ is the sum of times there is a click difference in same partition
  • This concentration bound is better to use $N_{ab}$ instead of the number of samples
Online Learning to Rank with Features (LLS, ICML’19)

• Each item $a$ is represented by a feature vector $x_a \in \mathbb{R}^d$
• The attractiveness of item $a$ is $\alpha(a) = \theta^T x_a$
• The concentration bound
  \[ \mathbb{E}[X_{ab} | X_{ab} \neq 0] \geq \frac{\alpha(a) - \alpha(b)}{\alpha(a) + \alpha(b)} \]
  can’t be transferred to a concentration bound for $\theta$
• RecurRank (Recursive Ranking)

• Li, S., Lattimore, T., & Szepesvari, C. Online Learning to Rank with Features. ICML, 2019.
## Bandit Works for OLTR with Click Models

<table>
<thead>
<tr>
<th>Context</th>
<th>Click Model</th>
<th>Regret</th>
</tr>
</thead>
<tbody>
<tr>
<td>[KSWA, ICML’2015]</td>
<td>CM</td>
<td>$\Theta\left(\frac{L}{\Delta} \log T\right)$</td>
</tr>
<tr>
<td>[LWZC, ICML’2016] [ZNSKWK, UAI’2016] [LZ, AAAI’2018]</td>
<td>GL CM</td>
<td>$O\left(d\sqrt{TK} \log T\right)$</td>
</tr>
<tr>
<td>[KKSW, ICML’2016]</td>
<td>DCM</td>
<td>$O\left(\frac{L}{\Delta} \log T\right)$</td>
</tr>
<tr>
<td>[LLZ, COCOON’2018]</td>
<td>GL DCM</td>
<td>$O\left(dK\sqrt{TK} \log T\right)$</td>
</tr>
<tr>
<td>[LVC, NeurIPS’2016]</td>
<td>PBM with known $\beta$</td>
<td>$O\left(\frac{L}{\Delta} \log T\right)$</td>
</tr>
<tr>
<td>[ZTGKSW, ICML’2017] [LKL, NeurIPS’2018]</td>
<td>General</td>
<td>$O\left(\frac{LK}{\Delta} \log T\right)$, $O\left(\sqrt{K^3LT \log T}\right)$, $\Omega\left(\sqrt{LKT}\right)$</td>
</tr>
<tr>
<td>[LLS, ICML’2019]</td>
<td>Linear General</td>
<td>$O\left(\sqrt{K^3dT \log(nT)}\right)$</td>
</tr>
</tbody>
</table>
Best-of-both-worlds
Adversarial MAB

• There are \( n \) arms
  • An adversary secretly preselects all loss vectors \( \{l_{t,a}\}_{t,a} \) from \([0,1]\)
  • The best arm is \( a^* = \text{argmin} \sum_{t=1}^{T} l_{t,a} \)
Setting of Adversarial MAB

• At each time $t$
  • The learning agent selects one arm $a_t$
  • Observe the loss $l_{t,a_t}$

• Objective:
  • Minimize the expected cumulative loss in $T$ rounds $\mathbb{E}\left[\sum_{t=1}^{T} l_{t,a_t}\right]$
  • Minimize the regret in $T$ rounds
    \[
    R(T) = \mathbb{E}\left[\sum_{t=1}^{T} l_{t,a_t}\right] - \min_a \sum_{t=1}^{T} l_{t,a}
    \]

• Balance the trade-off between exploration and exploitation
  • Exploitation: Select arms that yield good results so far
  • Exploration: Select arms that have not been tried much before
Exp3: Exponential Weight Algorithm for Exploration and Exploitation

• Importance-weight estimator

\[ \hat{\ell}_{t,i} = \frac{\mathbb{I}\{a_t = i\} \cdot l_{t,a_t}}{\mathbb{P}(a_t = i)} \]

• For each time \( t \)
  • Calculate the sampling distribution

\[ \mathbb{P}(a_t = i) = \frac{\exp(-\eta \hat{L}_{t-1,i})}{\sum_{j=1}^{n} \exp(-\eta \hat{L}_{t-1,j})} \]

  • Sample \( a_t \sim \mathbb{P}(a_t = i) \) and observe \( l_{t,a_t} \)
  • Calculate \( \hat{L}_{t,i} = \sum_{s=1}^{t} \hat{\ell}_{t,i} \)

• Regret bound \( O(\sqrt{LT \log L}) \)

Comparison between Stochastic and Adversarial Environments

- **Stochastic**
  - Reward fixed distribution on $[0,1]$ with fixed mean
  - Best arm $a^* = \arg\max \alpha(a)$
  - Regret bound $O(\log T)$
  - Runs in adversarial setting
    - Regret may not even converge

- **Adversarial**
  - Loss arbitrary on $[0,1]$
  - Best arm $a^* = \arg\min \sum_{t=1}^{T} l_{t,a}$
  - Regret bound $O(\sqrt{T})$
  - Runs in stochastic setting
    - Regret bound $O(\sqrt{T})$

Can we design algorithms that achieve $O(\log T)$ regret if run in stochastic setting and $O(\sqrt{T})$ if run in adversarial setting?

**Best of Both Worlds**
Best-of-both-worlds in OLTR
Adversarial setting under PBM

• The adversary secretly preselects the loss vectors $l_{t,i,j} \in \{0,1\}$ for any round $t$, item $i$ at position $j$

• The action set can be rewritten as

$$\chi = \left\{ X \in \{0,1\}^{L \times K} : \sum_{i=1}^{L} X_{i,j} = 1, \forall j \in [K]; \sum_{j=1}^{K} X_{i,j} \leq 1, \forall i \in [L] \right\}$$

• Objective: Minimize the regret over $T$ rounds

$$R(T) = \mathbb{E} \left[ \sum_{t=1}^{T} \langle X_t, l_t \rangle - \min_{X \in \chi} \sum_{t=1}^{T} \langle X, l_t \rangle \right]$$
Algorithm: Follow the Regularized Leader (FTRL)

- Input $\chi$
- $\hat{L}_0 = 0_{L\times K}, \eta_t = 1/(2\sqrt{t})$
- For $t = 1,2, ...$
  - Compute $x_t = \arg \min_{x \in \text{Conv}(\chi)} \langle x, \hat{L}_{t-1} \rangle + \eta_t^{-1}\Psi(x)$
  - Sample $X_t \sim P(x_t)$
  - Compute the loss estimator $\hat{l}_{t,i,j} = \frac{\mathbb{I}\{x_{t,i,j} = 1\} \cdot l_{t,i,j}}{x_{t,i,j}}$
  - Compute $\hat{L}_t = \hat{L}_{t-1} + \hat{l}_t$
Proof idea in the adversarial setting

- $\Psi(x) = \sum_i -\sqrt{x_i}$ for $x \in [0,1]^L$ $\frac{1}{2}$-Tsallis entropy
- Let $\Phi_t(\cdot) = \max_{x \in \text{Conv}(\chi)} \langle x, \cdot \rangle - \eta_t^{-1} \Psi(x)$ Fenchel conjugate

$$R(T) = \mathbb{E} \left[ \sum_{t=1}^{T} \langle X_t, l_t \rangle + \Phi_t(-\hat{L}_t) - \Phi_t(-\hat{L}_{t-1}) \right]$$

- Stability Term
- Regularization Penalty Term
Proof idea in the adversarial setting 2

• $R_{stab} \leq \sum_{t=4}^{T} \left[ 2\eta_t \sum_{j=1}^{K} \sum_{i \neq l_j^*} \left( \sqrt{\mathbb{E}[x_{t,i,j}]} + \mathbb{E}[x_{t,i,j}] \right) \right] + O(K \log T)$

• $R_{pen} \leq \sum_{t=1}^{T} \sum_{j=1}^{K} \sum_{i \neq l_j^*} \frac{1}{\sqrt{t}} \left( 2 \sqrt{\mathbb{E}[x_{t,i,j}]} - \mathbb{E}[x_{t,i,j}] \right)$

• By Cauchy-Schwartz Theorem
  
  \[ R(T) = O(K\sqrt{LT}) \]
Proof idea in the stochastic setting

• In the stochastic case, $l_{t,i,j} \sim \text{Ber}(1 - \alpha_i \beta_j)$

• Define gap for PBM as

$$\Delta_{i,j} = \begin{cases} 
(\beta_j - \beta_{j+1})(\alpha_j - \alpha_i), & j < i \\
0, & j = i \\
(\beta_{j-1} - \beta_j)(\alpha_i - \alpha_j), & j > i 
\end{cases}$$

the minimal regret incurred as putting item $i$ at position $j$

• $R(T) \geq \sum_{t=1}^{T} \sum_{i=1}^{L} \sum_{j=1}^{K} \frac{1}{2} \Delta_{i,j} \mathbb{E}[x_{t,i,j}]$ self-bounding constraint

• $R(T) = O \left( \sum_{i \neq j} \frac{\log T}{\Delta_{i,j}} \right)$

• Also provide a lower bound $R(T) = \Omega(K\sqrt{LT})$
<table>
<thead>
<tr>
<th>Contributions to Existing Works</th>
<th>Context</th>
<th>Click Model</th>
<th>Regret</th>
<th>Adversarial</th>
</tr>
</thead>
<tbody>
<tr>
<td>[KSWA, ICML’2015]</td>
<td>-</td>
<td>CM</td>
<td>$\Theta \left( \frac{L}{\Delta} \log T \right)$</td>
<td></td>
</tr>
<tr>
<td>[LWZC, ICML’2016] [ZNSKWK, UAI’2016] [LZ, AAAI’2018]</td>
<td>GL</td>
<td>CM</td>
<td>$O\left( d\sqrt{TK \log T} \right)$</td>
<td></td>
</tr>
<tr>
<td>[KKSW, ICML’2016]</td>
<td>-</td>
<td>DCM</td>
<td>$O\left( \frac{L}{\Delta} \log T \right)$</td>
<td></td>
</tr>
<tr>
<td>[LLZ, COCOON’2018]</td>
<td>GL</td>
<td>DCM</td>
<td>$O\left( dK\sqrt{TK \log T} \right)$</td>
<td></td>
</tr>
<tr>
<td>[LVC, NeurIPS’2016]</td>
<td>-</td>
<td>PBM with known $\beta$</td>
<td>$O\left( \frac{L}{\Delta} \log T \right)$</td>
<td></td>
</tr>
<tr>
<td>[ZTGKSW, ICML’2017] [LKLS, NeurIPS’2018]</td>
<td>General</td>
<td></td>
<td>$O\left( \frac{LK}{\Delta} \log T \right)$</td>
<td>$O\left( \sqrt{K^3LT \log T} \right)$</td>
</tr>
<tr>
<td>[LLS, ICML’2019]</td>
<td>General</td>
<td></td>
<td>$O\left( K\sqrt{dT \log nT} \right)$</td>
<td></td>
</tr>
<tr>
<td>In submission</td>
<td>PBM</td>
<td></td>
<td>$O\left( \frac{KL}{\delta \beta \Delta} \log T \right)$</td>
<td>$O(K\sqrt{LT})$</td>
</tr>
</tbody>
</table>
Other Related Works
Thompson Sampling Algorithms

• Cascade Model:

• Position-based Model:

• Dependent Click Model:
  • In submission
Other metrics

• Safety
  • Li, C., Kveton, B., Lattimore, T., Markov, I., de Rijke, M., Szepesvári, C., & Zoghi, M. BubbleRank: Safe online learning to re-rank via implicit click feedback. UAI, 2020.

• Diversity

• Differential privacy
Click Models

• Imitation learning

• Graph NN
Summary: OLTR

- Stochastic
  - UCB: CM, DCM, PBM, General (matched upper/lower bounds)
  - TS: CM, PBM

- Adversarial
  - PBM

- Best-of-both-worlds
  - PBM (not tight in the stochastic case)
Future Directions

• Stochastic: Thompson sampling algorithm for generalized click model
• Adversarial: Cascade click model
• Adversarial: General click model
• Best-of-both-worlds: General click model
• Corruption & attack

You are welcome to contact me if you are interested in any of these topics.
Other Research Projects

• Online influence maximization
  • Analysis for Thompson sampling

• Online matching markets
  • Many-to-one matching
  • Thompson sampling algorithms

• Best-of-both-worlds
  • Online learning with graph feedback
  • Multi-agent with communication graph

• Online clustering of bandits

• Conversation aided recommendations
  • The application of bandit/RL algorithms
Thanks!

&

Questions?

Shuai Li

• Assistant Professor at John Hopcroft Center Shanghai Jiao Tong University
• Research interests: Bandit/RL algorithms
• Personal website: http://shuaili8.github.io/