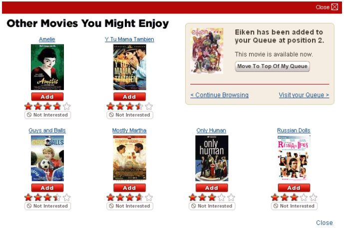
Offline Evaluation of Ranking Policies with Click Models

Shuai Li

The Chinese University of Hong Kong

Joint work with
Yasin Abbasi-Yadkori (Adobe Research)
Branislav Kveton (Google Research, was in Adobe Research)
S. Muthukrishnan (Rutgers University)
Vishwa Vinay (Adobe Research)
Zheng Wen (Adobe Research)

Motivation



Amazon, Facebook, Netflix

Motivation

Production Policy π

Number of clicks: $V(\pi) = 1$

Hypothetical Policy h

Number of clicks: $V(h) = 2$

Search “London” @ Adobe Stock

Motivation

Production Policy π

Number of clicks: $V(\pi) = 1$

Hypothetical Policy h

Number of clicks: $V(h) = 2$

- How can we know $V(h) = 2$?

Search “London” @ Adobe Stock

Directly Implementing New Policy h

Risks for directly implementing new policy h

- Expensive
- Uses a portion of live users and poor policy might harm user experience
- Not replicable

Directly Implementing New Policy h

Risks for directly implementing new policy h

- Expensive
- Uses a portion of live users and poor policy might harm user experience
- Not replicable

Can we know $V(h) = 2$ without directly implementing it?

Directly Implementing New Policy h

Risks for directly implementing new policy h

- Expensive
- Uses a portion of live users and poor policy might harm user experience
- Not replicable

Can we know $V(h) = 2$ without directly implementing it?

Offline Evaluation!

Setting

- Lists: $A = (a_1, \ldots, a_K)$

$\ldots$

Setting

- Lists: $A = (a_1, \ldots, a_K)$

$\ldots$

- The value of list A with the click realization w:

$$V(A; w) = \sum_{k=1}^{K} w(a_k, k)$$

Setting

- Lists: $A = (a_1, \dots, a_K)$

$\dots$

- The value of list A with the click realization w:

$$V(A; w) = \sum_{k=1}^{K} w(a_k, k)$$

- The value of a policy h:

$$V(h) = \mathbb{E}_{x, w, A \sim h(\cdot|x)} \left[V(A; w) \right]$$

Setting

- Suppose the clicks depend only on (item, position) pairs

Setting

- Suppose the clicks depend only on (item, position) pairs
- The CTR of putting item a at k-th position under context x is

$$\bar{w}(a, k \mid x)$$

Setting

- Suppose the clicks depend only on (item, position) pairs
- The CTR of putting item a at k-th position under context x is

$$\bar{w}(a, k \mid x)$$

- The expected value of A

$$V(A) = \sum_{k=1}^{K} \bar{w}(a_k, k \mid x)$$

Setting (Problem Definition)

Logged dataset $S = \{(x_t, A_t, w_t)\}_{t=1}^{n}$

- At each time t
  - The environment draws context x_t and click realizations w_t
  - The learner observes x_t and selects A_t according to policy π
  - The environment reveals $\{w_t(a_k^t, k)\}_{k=1}^{K}$

Setting (Problem Definition)

Logged dataset $S = \{(x_t, A_t, w_t)\}_{t=1}^{n}$

- At each time t
  - The environment draws context x_t and click realizations w_t
  - The learner observes x_t and selects A_t according to policy π
  - The environment reveals $\{w_t(a_k^t, k)\}_{k=1}^{K}$

Objective

- To design statistically efficient estimators based on logged dataset for any ranking policy

Setting (Problem Definition)

Logged dataset $S = \{(x_t, A_t, w_t)\}_{t=1}^{n}$

- At each time t
  - The environment draws context x_t and click realizations w_t
  - The learner observes x_t and selects A_t according to policy π
  - The environment reveals $\{w_t(a_k^t, k)\}_{k=1}^{K}$

Objective

- To design statistically efficient estimators based on logged dataset for any ranking policy

Challenge

- The number of different lists is exponential in K

Existing Method - Direct Method

- Direct Method

$$\hat{V}(h) = \frac{1}{n} \sum_{t=1}^{n} \sum_{a} \sum_{k=1}^{K} h(a, k \mid x_t) \, \hat{w}(a, k \mid x_t)$$

Existing Method - Direct Method

- Direct Method

$$\hat{V}(h) = \frac{1}{n} \sum_{t=1}^{n} \sum_{a} \sum_{k=1}^{K} h(a, k \mid x_t) \, \hat{w}(a, k \mid x_t)$$

- Can be used to evaluate any policy
- Unstable when the number of observations for some item is small
- No theoretical guarantee for known computationally efficient method for some click models

Existing Method - Unstructured List Estimator [Strehl et al. 2010]

- Importance sampling (for list level)

$$
\begin{aligned}
V(h) &= \mathbb{E}_{A \sim h}[V(A)] \\
&= \mathbb{E}_{A \sim h}\left[V(A) \cdot \frac{\pi(A)}{\pi(A)}\right] \\
&= \mathbb{E}_{A \sim \pi}\left[V(A) \cdot \frac{h(A)}{\pi(A)}\right]
\end{aligned}
$$

Existing Method - Unstructured List Estimator [Strehl et al. 2010]

- Importance sampling (for list level)

$$\begin{aligned} V(h) &= \mathbb{E}_{A \sim h}[V(A)] \\ &= \mathbb{E}_{A \sim h}\left[V(A) \cdot \frac{\pi(A)}{\pi(A)}\right] \\ &= \mathbb{E}_{A \sim \pi}\left[V(A) \cdot \frac{h(A)}{\pi(A)}\right] \end{aligned}$$

- List estimator

$$\hat{V}_{\mathrm{L}}(h) = \frac{1}{|S|} \sum_{(x,A,w) \in S} V(A; w) \ \min\left\{\frac{h(A \mid x)}{\hat{\pi}(A \mid x)},\ M\right\}$$

Existing Method - Unstructured List Estimator [Strehl et al. 2010]

- Importance sampling (for list level)

$$\begin{aligned} V(h) &= \mathbb{E}_{A\sim h}[V(A)] \\ &= \mathbb{E}_{A\sim h}\left[V(A)\cdot\frac{\pi(A)}{\pi(A)}\right] \\ &= \mathbb{E}_{A\sim \pi}\left[V(A)\cdot\frac{h(A)}{\pi(A)}\right] \end{aligned}$$

Trade-off between bias and variance

- List estimator

$$\hat{V}_{\mathrm{L}}(h) = \frac{1}{|S|}\sum_{(x,A,w)\in S} V(A;w)\ \min\left\{\frac{h(A\mid x)}{\hat{\pi}(A\mid x)},\ M\right\}$$

Empirical distribution over logged data

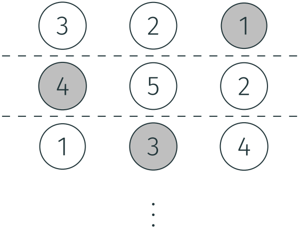
List Estimator - Disadvantages

- Disadvantages
  - Have to match the exact lists
  - The number of lists is exponential in K, thus $\hat{\pi}(A \mid x)$ is very small

List Estimator - Disadvantages

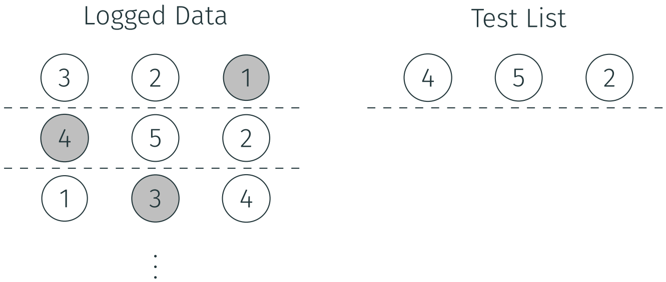
- Disadvantages
  - Have to match the exact lists
  - The number of lists is exponential in K, thus $\hat{\pi}(A \mid x)$ is very small

Logged Data



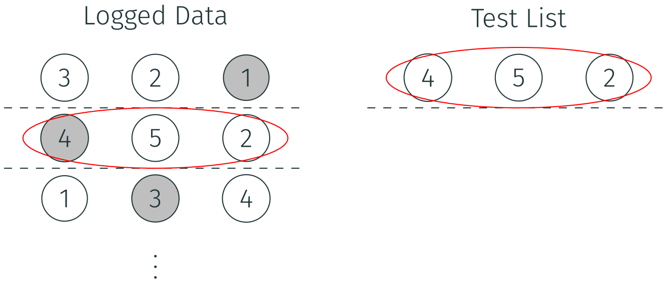
List Estimator - Disadvantages

- Disadvantages
  - Have to match the exact lists
  - The number of lists is exponential in K, thus $\hat{\pi}(A \mid x)$ is very small



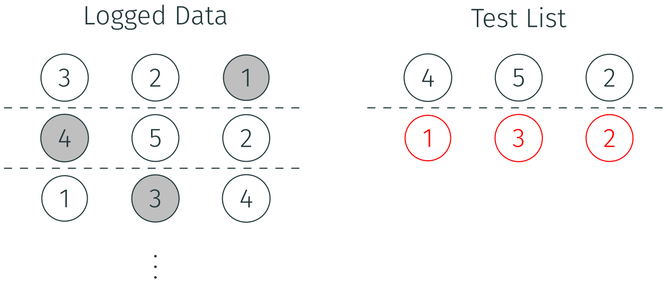
List Estimator - Disadvantages

- Disadvantages
  - Have to match the exact lists
  - The number of lists is exponential in K, thus $\hat{\pi}(A \mid x)$ is very small



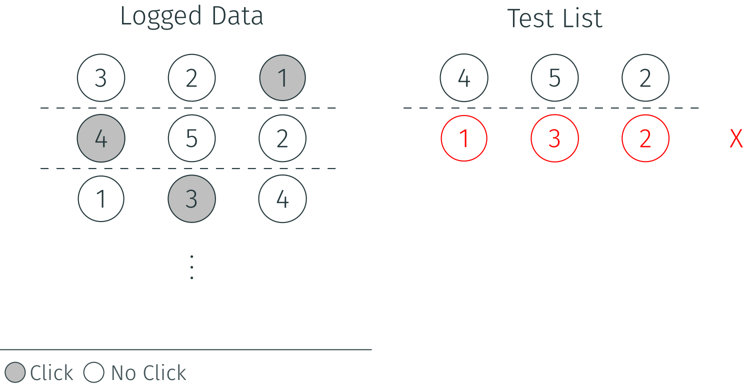
List Estimator - Disadvantages

- Disadvantages
  - Have to match the exact lists
  - The number of lists is exponential in K, thus $\hat{\pi}(A \mid x)$ is very small



List Estimator - Disadvantages

- Disadvantages
  - Have to match the exact lists
  - The number of lists is exponential in K, thus $\hat{\pi}(A \mid x)$ is very small

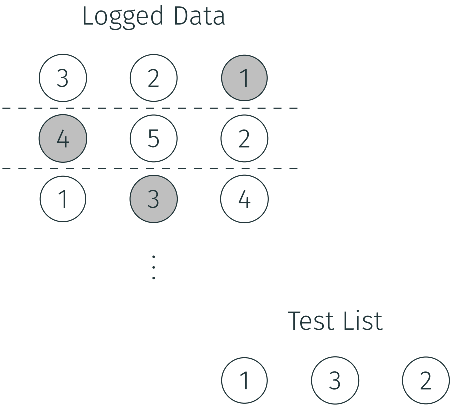


Document-Based Click Model (DCTR)

- $\bar{w}(a, k \mid x)$ only depends on item a, for any context x

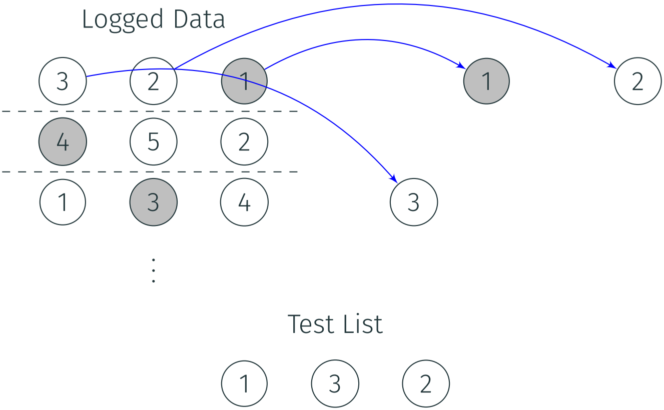
Document-Based Click Model (DCTR)

- $\bar{w}(a, k \mid x)$ only depends on item a, for any context x



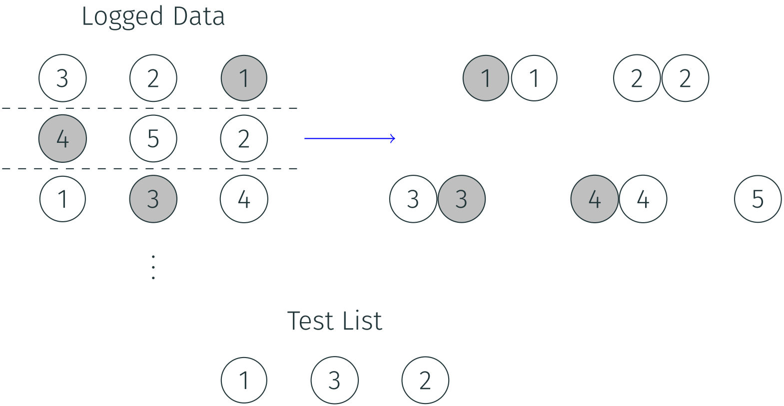
Document-Based Click Model (DCTR)

- $\bar{w}(a, k \mid x)$ only depends on item a, for any context x



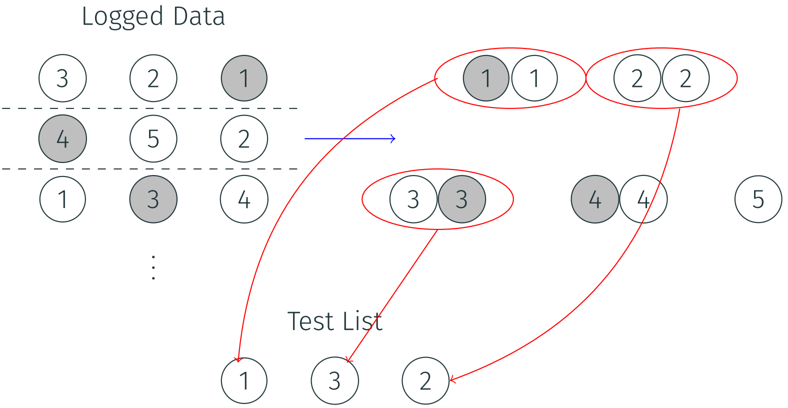
Document-Based Click Model (DCTR)

- $\bar{w}(a, k \mid x)$ only depends on item a, for any context x



Document-Based Click Model (DCTR)

- $\bar{w}(a, k \mid x)$ only depends on item a, for any context x



Estimators for DCTR

- DCTR: $\bar{w}(a, k \mid x)$ only depends on item a for any context x

$$\hat{V}_{\mathrm{I}}(h) = \frac{1}{|S|} \sum_{(x,A,w) \in S} \sum_{k=1}^{K} w(a_k, k) \ \min \left\{ \frac{h(a_k \mid x)}{\hat{\pi}(a_k \mid x)}, \ M \right\}$$

Estimators for DCTR

- DCTR: $\bar{w}(a, k \mid x)$ only depends on item a for any context x

$$\hat{V}_{\mathrm{I}}(h) = \frac{1}{|S|} \sum_{(x,A,w)\in S} \sum_{k=1}^{K} w(a_k, k) \ \min\left\{ \frac{h(a_k \mid x)}{\hat{\pi}(a_k \mid x)},\ M \right\}$$

- List estimator

$$\hat{V}_{\mathrm{L}}(h) = \frac{1}{|S|} \sum_{(x,A,w)\in S} \sum_{k=1}^{K} w(a_k, k) \ \min\left\{ \frac{h(A \mid x)}{\hat{\pi}(A \mid x)},\ M \right\}$$

Estimators for Click Models

Click Model	Assumption	Estimator
Random	$\bar{w}(a, k \mid \cdot)$ constant	$\hat{V}_{\mathrm{R}}$
Rank-Based	$\bar{w}(a, k \mid \cdot)$ only depends on position k	$\hat{V}_{\mathrm{R}}$
Document-Based	$\bar{w}(a, k \mid \cdot)$ only depends on item a	$\hat{V}_{\mathrm{I}}$
Position-Based	$\bar{w}(a, k \mid \cdot) = \mu(a \mid \cdot)\, p(k \mid \cdot)$	$\hat{V}_{\mathrm{PBM}}$
Item-Position	$\bar{w}(a, k \mid \cdot)$	$\hat{V}_{\mathrm{IP}}$

Analysis

Proposition (Unbiased in a larger class of policies)

The structured estimators are unbiased in a larger class of policies than list estimator.

Proposition (Lower bias in estimating policy)

The structured estimators have lower bias than list estimator.

Proposition (Better guarantee for policy optimization)

The best policy found by structured estimators have better theoretical guarantees than that by list estimator.

Experiments

Personalized Web Search Challenge

Re-rank web documents using personal preferences

$9,000 · 194 teams · 5 years ago

- Recorded over 27 days
- Each record contains
  - A query ID
  - The day when the query occurs
  - 10 displayed item as a response to the query
  - The corresponding click indicators of each displayed items

Experiments

Yandex

Personalized Web Search Challenge

Re-rank web documents using personal preferences

$9,000 · 194 teams · 5 years ago

- Recorded over 27 days
- Each record contains
  - A query ID
  - The day when the query occurs
  - 10 displayed item as a response to the query
  - The corresponding click indicators of each displayed items

- Logged dataset S
  - Any record except day d
  - $\hat{\pi}$ is the empirical distribution over S

Experiments

Personalized Web Search Challenge

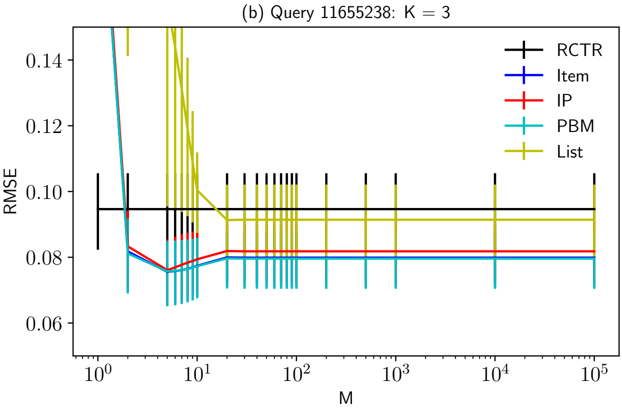
Re-rank web documents using personal preferences

$9,000 · 194 teams · 5 years ago

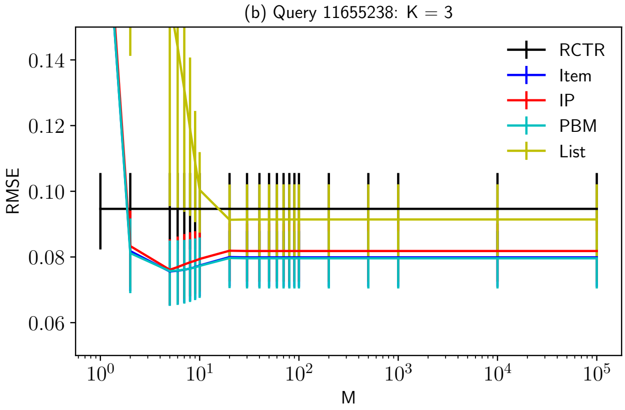
- Recorded over 27 days
- Each record contains
  - A query ID
  - The day when the query occurs
  - 10 displayed item as a response to the query
  - The corresponding click indicators of each displayed items

- Logged dataset S
  - Any record except day d
  - $\hat{\pi}$ is the empirical distribution over S
- Evaluation policy h
  - Records of day d
  - h is the empirical distribution over these records
  - $V(h)$ is the average CTR for these records

Experiments - Example Query with $K = 3$

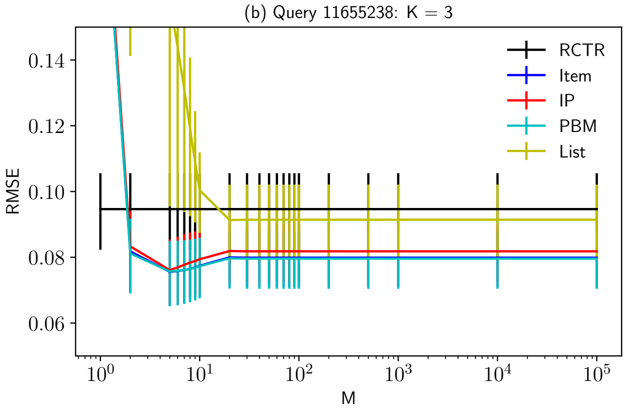


Experiments - Example Query with $K = 3$



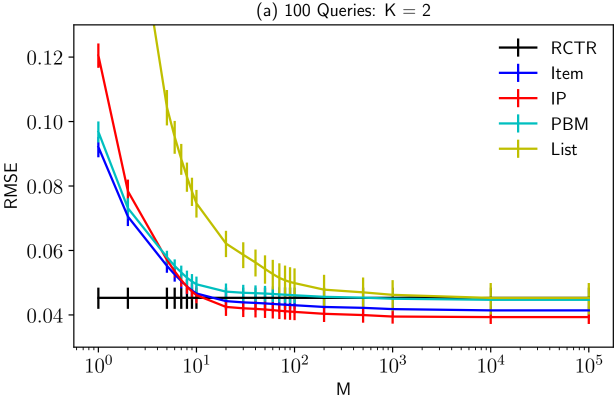
- Structured estimators better

Experiments - Example Query with $K = 3$

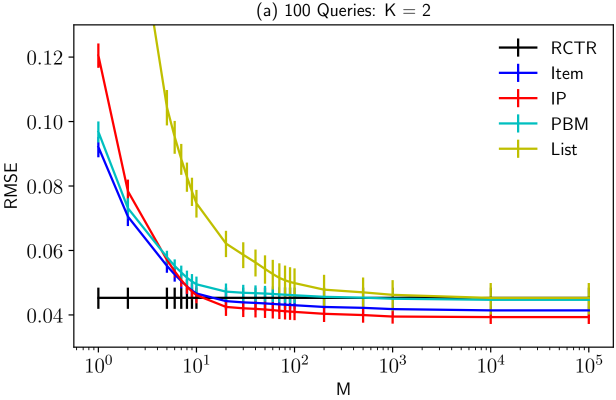


- Structured estimators better
- Tuning of M matters

Experiments - 100 Most Frequent Queries with $K = 2$

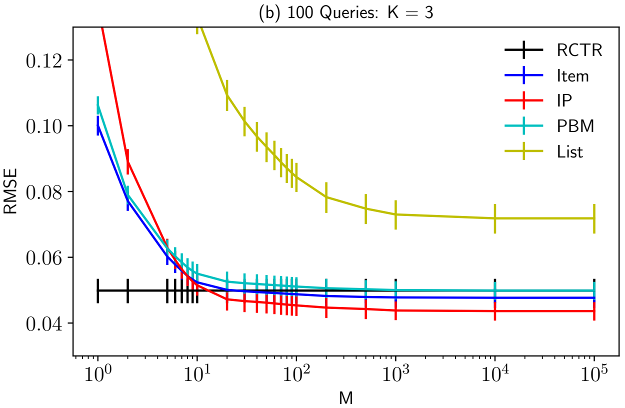


Experiments - 100 Most Frequent Queries with $K = 2$

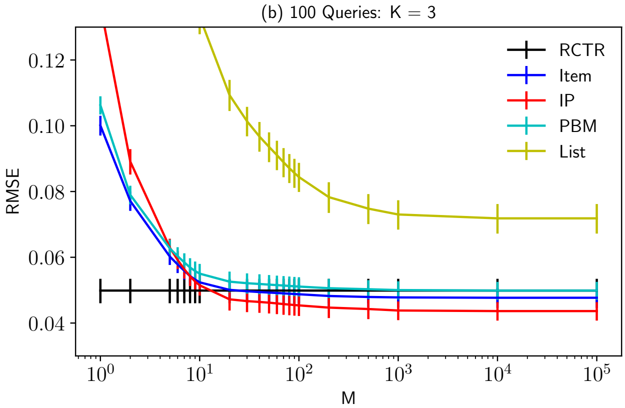


- IP estimator improves 18% over list estimator
- IP estimator improves 13% over RCTR estimator

Experiments - 100 Most Frequent Queries with $K = 3$

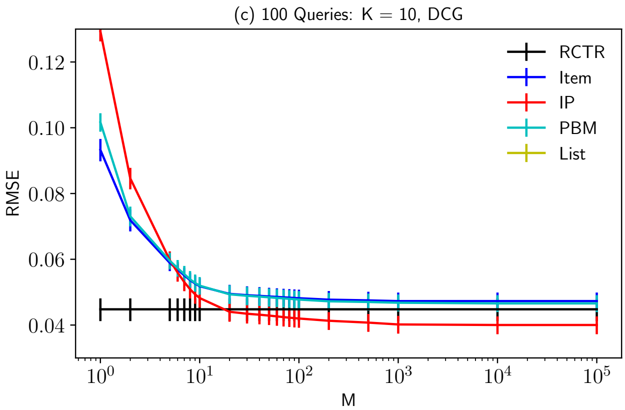


Experiments - 100 Most Frequent Queries with $K = 3$

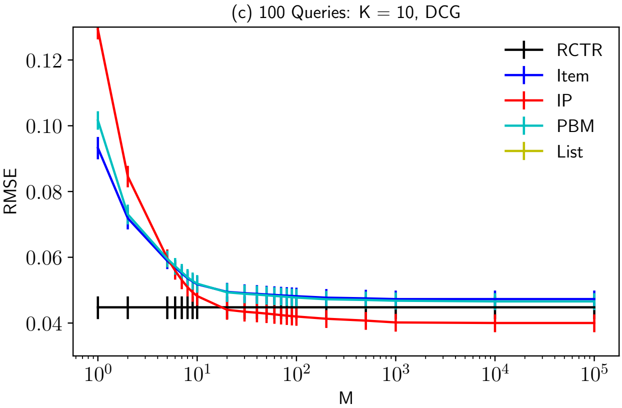


- IP estimator improves 46% over list estimator
- IP estimator improves 13% over RCTR estimator

Experiments - 100 Most Frequent Queries with $K = 10$, DCG



Experiments - 100 Most Frequent Queries with $K = 10$, DCG



- IP estimator improves 82% over list estimator
- IP estimator improves 11% over RCTR estimator

Conclusions

- We propose various estimators for the expected number of clicks on lists generated by ranking policies that leverage the structure of click models
- We prove that our estimators are better than the unstructured list estimators
    - Less biased
    - Better guarantees for policy optimization
- Our estimators consistently outperform the list estimator in experiments

T. Joachims, A. Swaminathan, and T. Schnabel.
Unbiased learning-to-rank with biased feedback.
In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 781–789. ACM, 2017.

A. Strehl, J. Langford, L. Li, and S. M. Kakade.
Learning from logged implicit exploration data.
In *Advances in Neural Information Processing Systems*, pages 2217–2225, 2010.

A. Swaminathan, A. Krishnamurthy, A. Agarwal, M. Dudik, J. Langford, D. Jose, and I. Zitouni.
Off-policy evaluation for slate recommendation.
In *Advances in Neural Information Processing Systems*, pages 3632–3642, 2017.