A Deep Recurrent Survival Model for Unbiased Ranking

Jiarui Jin[1], Yuchen Fang[1], Weinan Zhang[1], Kan Ren[2], Guorui Zhou[3], Jian Xu[3], Yong Yu[1], Jun Wang[4], Xiaoqiang Zhu[3], Kun Gai[3]

[1]Shanghai Jiao Tong University, [2]Microsoft Research, [3]Alibaba Group, [4]University College London

Slides: https://jiaruijin.com/publication/files/sigir20_oral.pdf

Content

- Problem Background
  - Unbiased Ranking
  - Current Challenges
- Architecture
  - Deep Recurrent Survival Model
  - Point-wise Loss Function
  - Pair-wise Loss Function
- Experiment
- Conclusion

Content

- **Problem Background**
  - Unbiased Ranking
  - Current Challenges
- Architecture
  - Deep Recurrent Survival Model
  - Point-wise Loss Function
  - Pair-wise Loss Function
- Experiment
- Conclusion

Explicit & Implicit Feedback

- Explicit Feedback
  - score (relevance)
  - predict relevance probability $P(r)$ relevance

query q

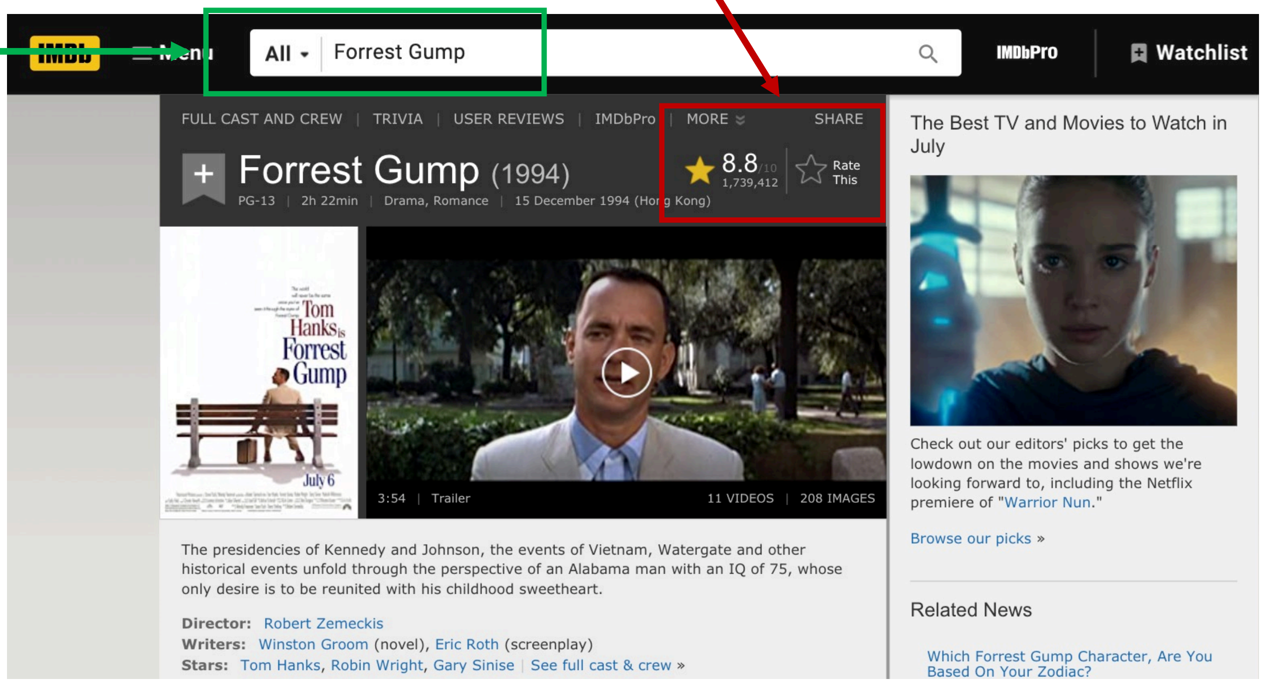


Figure: IMDB Movie

Explicit & Implicit Feedback

- Implicit Feedback
  - click probability, observe probability
  - predict ?

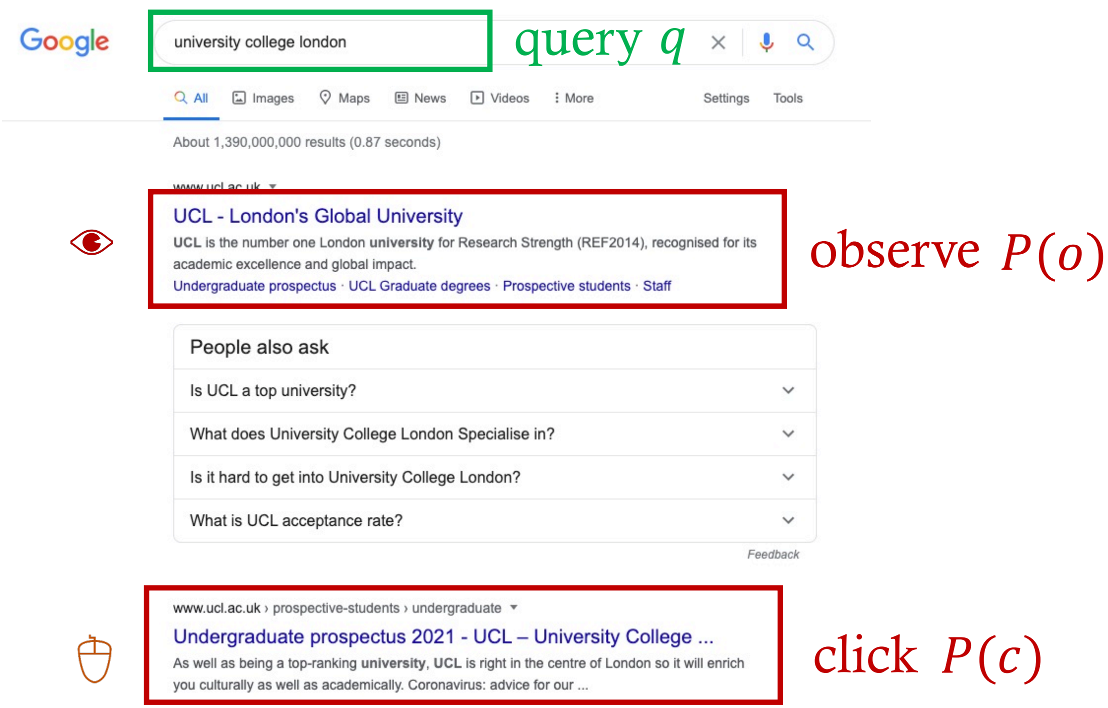


Figure: Google

Position Bias

- Traditional Ranking
  - information we have: click, observe
  - predict click and rank click probability $P(c)$

Search Tea

Figure: Illustration of position bias

Position Bias

- Traditional Ranking
  - information we have: click, observe
  - predict click and rank click probability $P(c)$
- Position Bias
  - item at low position is not likely to be observed and, of course, not be clicked
- Unbiased Ranking
  - information we have: click, observe
  - information we need: relevance probability $P(r)$

Search Tea

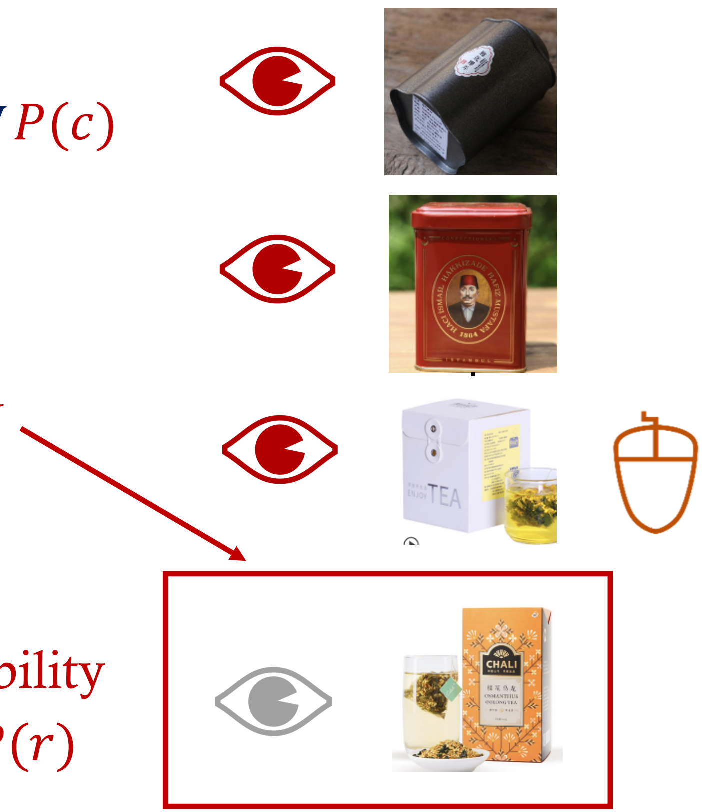


Figure: Illustration of position bias

Unbiased Ranking

- User Browsing Behavior Model
  - provides several basic assumptions about user browsing behavior.
  - [Thorsten Joachims, et al.] only model user behavior patterns without simultaneously sufficient optimization for ranking algorithm.
- Counterfual Learning Framework
  - treats the position bias as the counterfactual factor.
  - debiases the user feedbacks through inverse propensity weighting.
  - ignores the contextual information of the given ranking list, e.g., the content of the previous items may influence the observation of the next item.

Ref: Thorsten Joachims, et al. Unbiased learning-to-rank with biased feedback. WSDM, 2017.

Current Challenges

(C1) the user behaviors contain various and highly correlated patterns based on the contextual information.

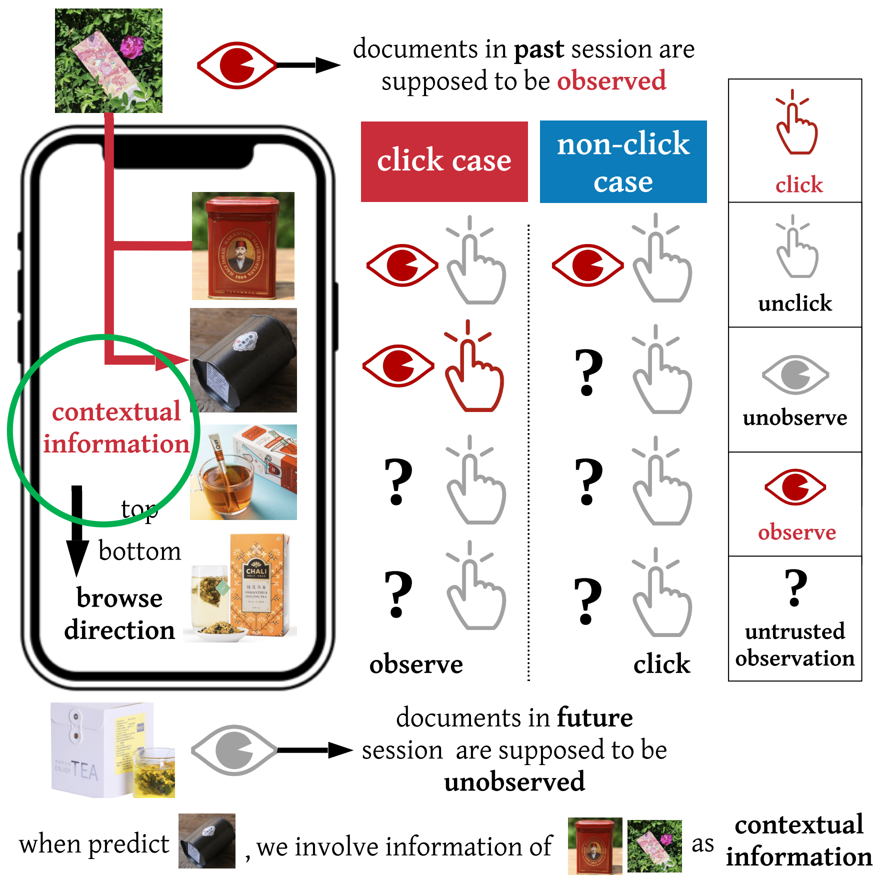
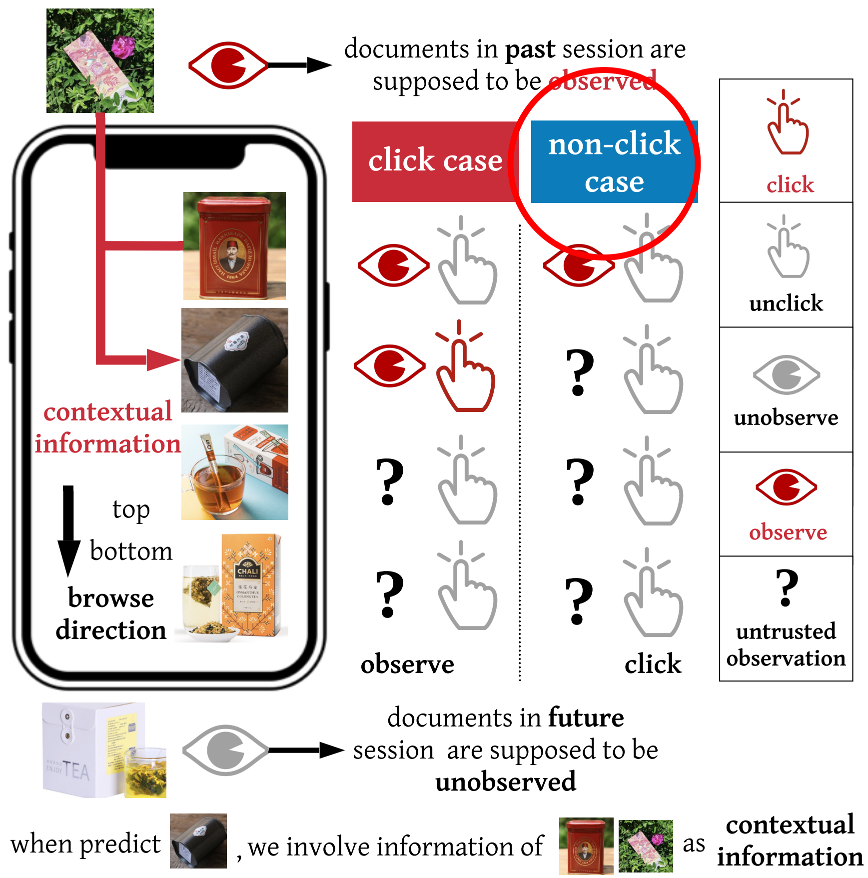


Figure: Illustration of user various behaviors (i.e., click and non-click case) when browsing document list as shown in the left side. Notations are provided in the right side.

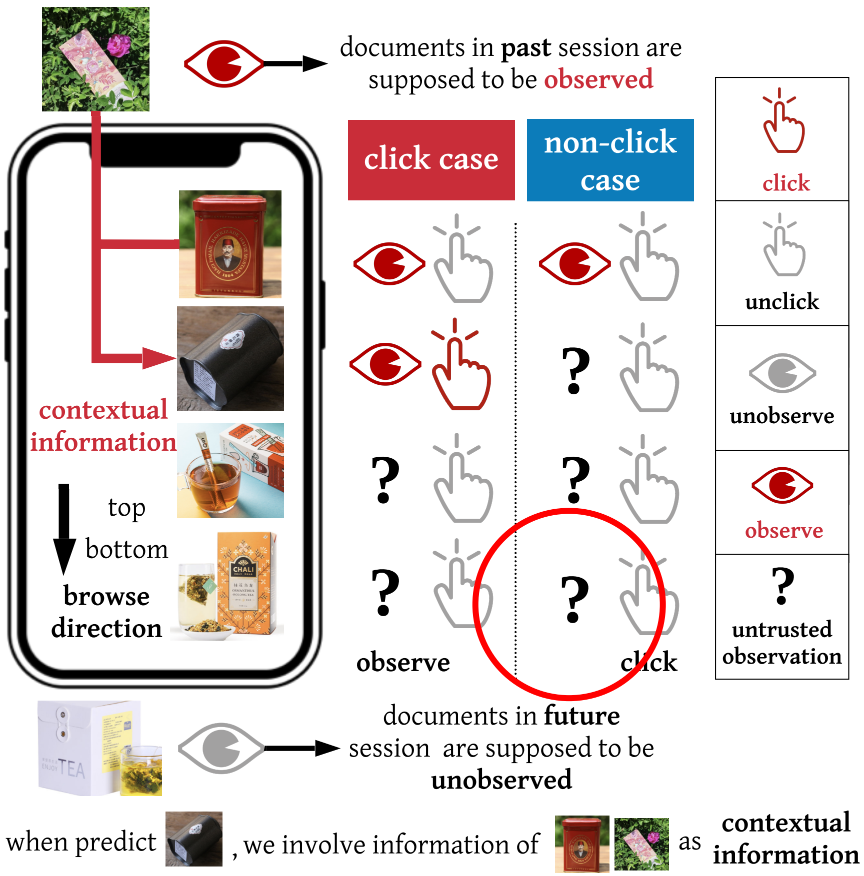
Current Challenges



(C2) There are large scale of latent observe patterns hidden in the non-click queries. i.e., when the user starts a search with a query, she stop browsing by interruption or end the search session due to lack of interest, which leaves many queries without any user click behavior, often referred as non-click queries

Figure: Illustration of user various behaviors (i.e., click and non-click case) when browsing document list as shown in the left side. Notations are provided in the right side.

Current Challenges



(C3) When the user scrolls down the page and observes the presented items, the tracking logs record user observations until the last position. However, this may not be true since the user may stop and lost her attention before that. We called these noisy logs untrusted observations.

Figure: Illustration of user various behaviors (i.e., click and non-click case) when browsing document list as shown in the left side. Notations are provided in the right side.

Content

- Problem Background
  - Unbiased Ranking
  - Current Challenges
- **Architecture**
  - Deep Recurrent Survival Model
  - Point-wise Loss Function
  - Pair-wise Loss Function
- Experiment
- Conclusion

Motivation

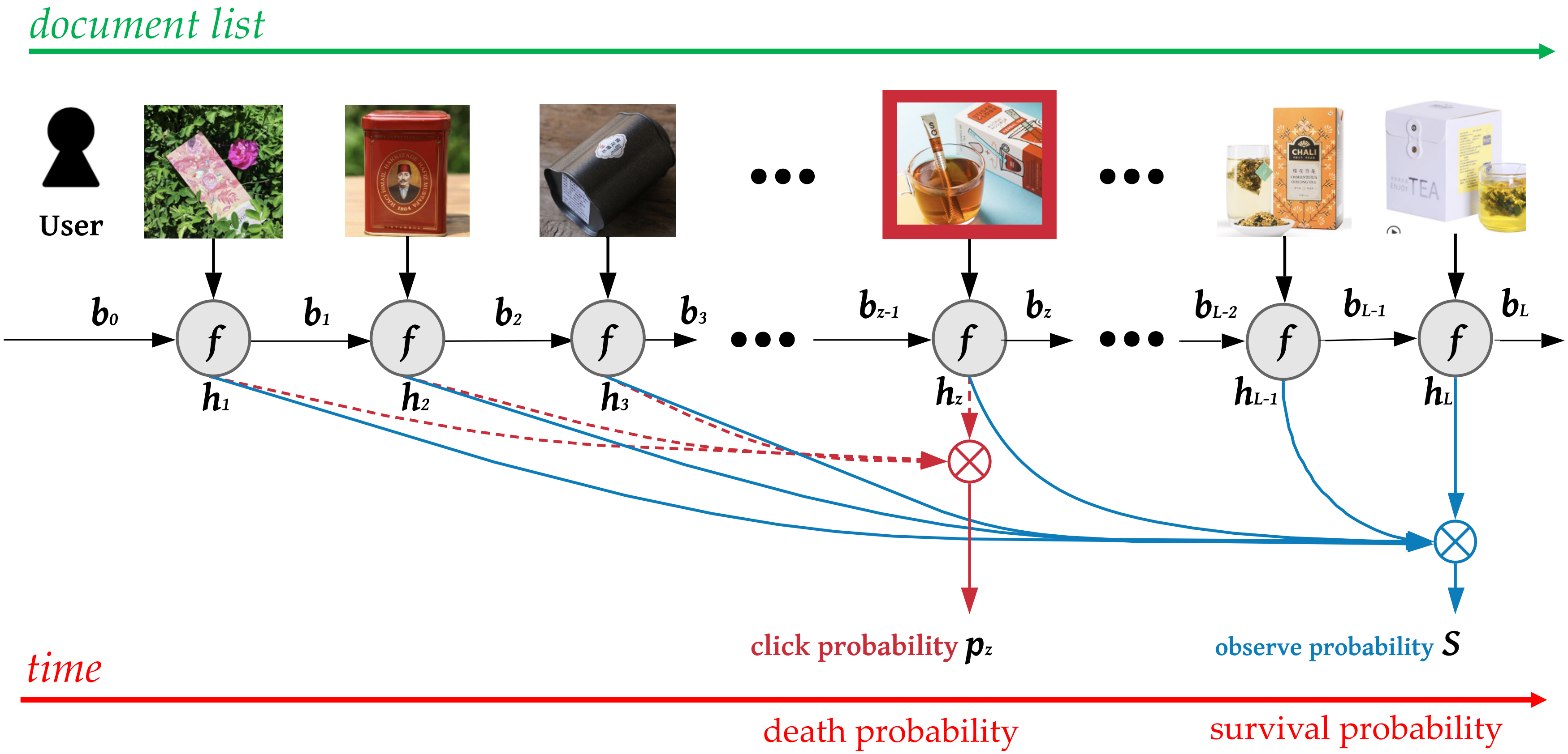


Figure: Illustration of Deep Recurrent Survival Ranking Model. Note that we mine click patterns in click case and observe patterns in both click and non-click cases.

Survival Model

click event z

- click probability (death probability), the probability density function (P.D.F.) of click occurring at i-th document

$$p_i = P(c_i = 1) = P(z = i)$$

- observe probability (survival probability), the cumulative distribution function (C.D.F.), since user will keep browsing until she finds and clicks a favored one

$$S(i) = P(o_i = 1) = \sum_{\tau \geq i} P(z = \tau)$$

- relevance probability, the conditional click probability, the click probability at document given that the document is observed

$$h_i = P(r_i = 1) = \frac{P(c_i = 1)}{P(o_i = 1)} = \frac{P(z = i)}{P(z \geq i)} = \frac{p_i}{S_i}$$

Deep Recurrent Survival Model

- relevance probability

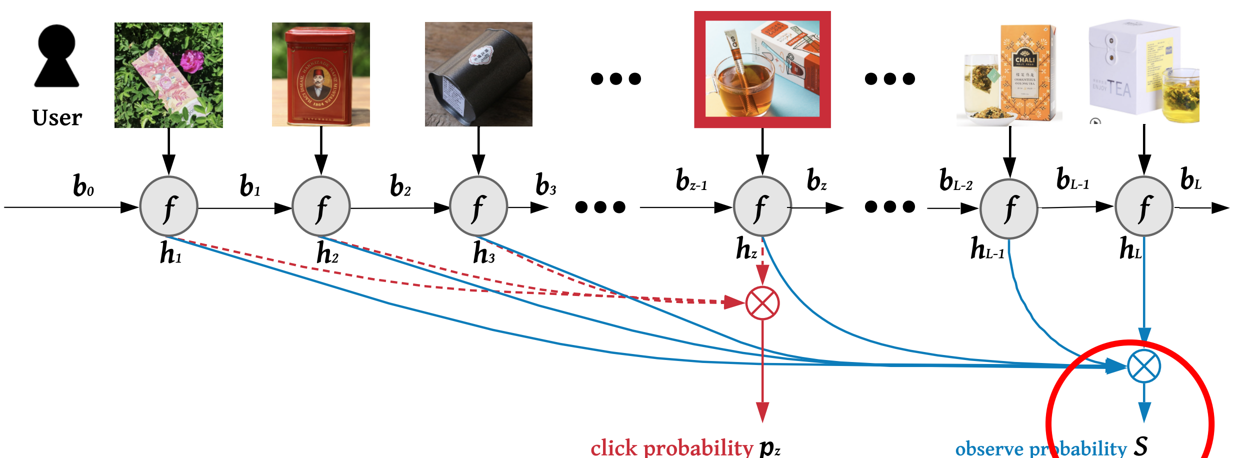
$$h_i = P(z = i | z \geq i, x; \theta) = \underline{f_\theta\,(x_i\,|b_{i-1})} \quad \text{RNN function}$$

- observe probability (survival probability)

$$S(i|x;\theta) = P(i \leq z) = P(z \neq 1, z \neq 2, z \neq 3, \ldots, z \neq i-1 | x; \theta)$$

$$= P(z \neq 1 | x_1; \theta) \cdot P(z \neq 2 | z \neq 1, x_2; \theta) \cdots P(z \neq i-1 | z \neq 1, \cdots z \neq i-2 | x_{i-1}; \theta)$$

$$= \prod_{\tau:\tau<i} [1 - P(z = \tau | z \geq \tau, x_\tau; \theta)] = \prod_{\tau:\tau<i} (1 - h_\tau)$$



Deep Recurrent Survival Model

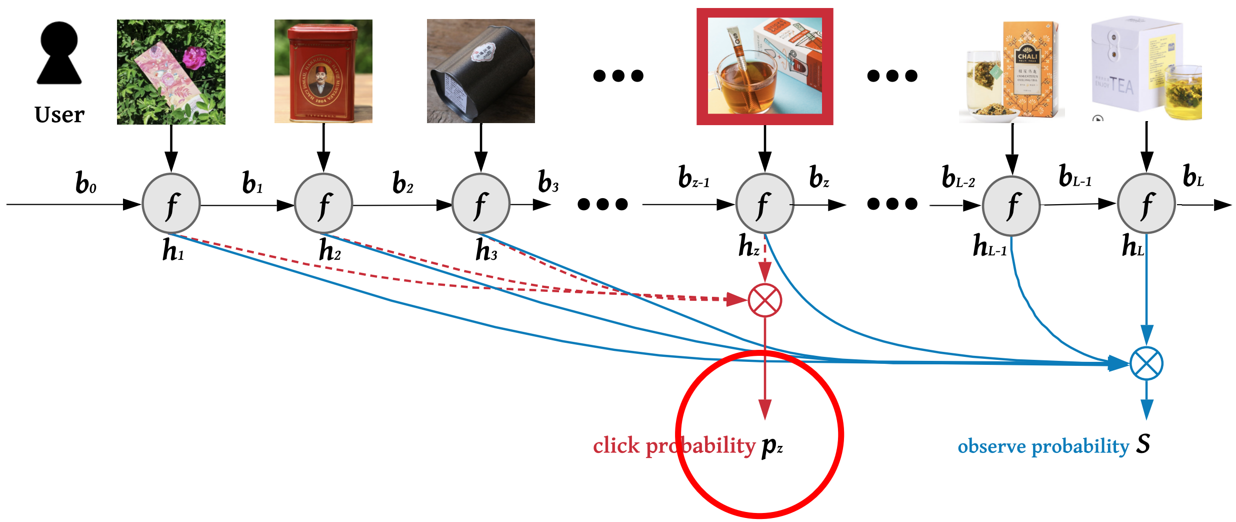
- relevance probability

$$h_i = P(z = i | z \geq i, x; \theta) = \underline{f_\theta (x_i | b_{i-1})} \quad \text{RNN function}$$

- click probability (death probability)

$$p_i = [1 - S(i+1)] - [1 - S(i)] = S(i) - S(i+1)$$

$$p_i = P(z = i | x; \theta) = h_i \prod_{\tau:\tau<i} (1 - h_\tau)$$



Alibaba Group

Apex Data & Knowledge Management Lab

Point-wise Loss Function

- click probability (P.D.F.)
  - aims to minimize negative log-likelihood of the click document d_j over the clicked logs as

$$L_{\text{point}(z)} = -\log \prod_{(x,z)\in\mathcal{D}_{\text{click}}} P(z=j|x;\theta) = -\log \prod_{(x,z)\in\mathcal{D}_{\text{click}}} p_j$$

$$= -\log \prod_{(x,z)\in\mathcal{D}_{\text{click}}} \left[h_j \prod_{\tau:\tau<i} (1-h_\tau)\right]$$

$$= -\sum_{(x,z)\in\mathcal{D}_{\text{click}}} [\log h_j + \sum_{\tau:\tau<i} \log(1-h_\tau)]$$

Point-wise Loss Function

- observe probability (C.D.F.)
  - push up the observe probability of document whose position in range of $[0, l]$, and pull down in range of $[l, \infty]$.

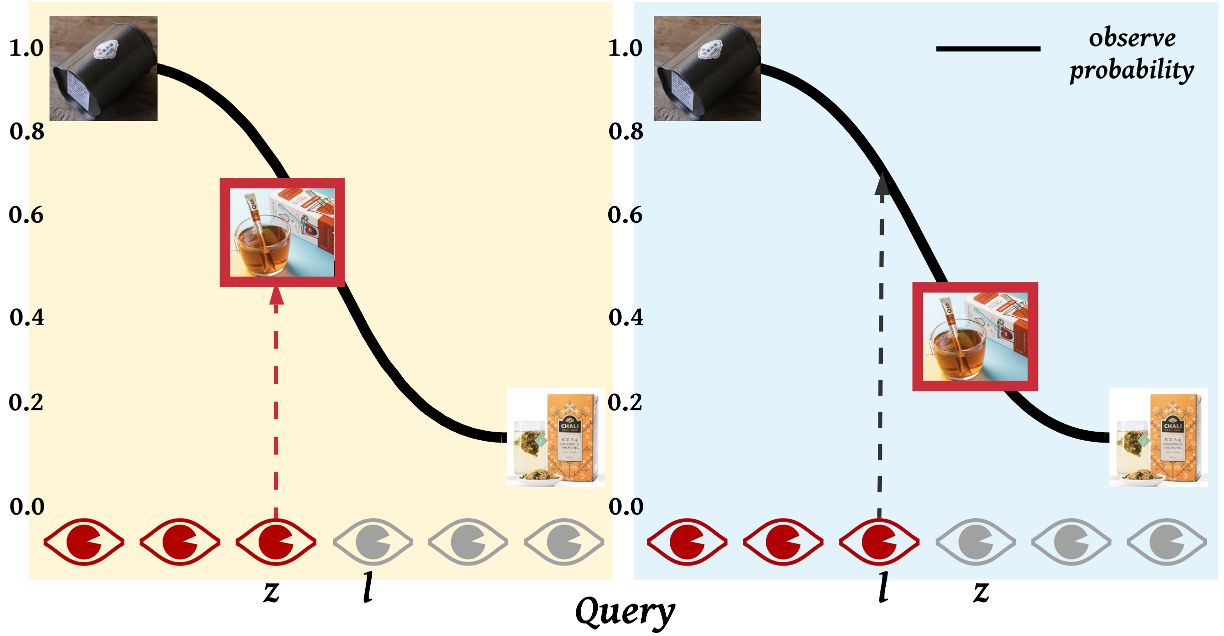


Figure: Iintuition behind C.D.F. losses. The left and right sub-figures denote click and non-click cases respectively.

Point-wise Loss Function

- observe probability (C.D.F.)
  - over the click cases

$$L_{\text{click}} = -\log \prod_{(x,l)\in\mathcal{D}_{\text{click}}} P(l \geq z|x;\theta) \approx -\log \sum_{(x,l)\in\mathcal{D}_{\text{click}}} [1 - S(l|x;\theta)]$$

$$= -\sum_{(x,l)\in\mathcal{D}_{\text{click}}} \log\,[1 - \prod_{\tau:\tau<l}(1-h_\tau)]$$

  - over the non-click cases

$$L_{\text{non-click}} = -\log \prod_{(x,l)\in\mathcal{D}_{\text{non-click}}} P(z > l|x;\theta) \approx -\log \sum_{(x,l)\in\mathcal{D}_{\text{non-click}}} S(l|x;\theta)$$

$$= -\sum_{(x,l)\in\mathcal{D}_{\text{non-click}}} \sum_{\tau:\tau<l} \log(1-h_\tau)$$

Pair-wise Loss Function

Figure: Illustration of permutation document modeling, where documents here are sampled from different subsets in the right side. Double sided arrow denotes exchange order operation.

Pair-wise Loss Function

- over the o0 query

$$L_{\text{pair}(o0)} = -\log \prod_{(d_i,d_j)\in I_q} P(z=j|z\geq i,x;\theta) = -\log \prod_{(d_i,d_j)\in I_q} \frac{P(z=j|x;\theta)}{P(z\geq i|x;\theta)}$$

$$= - \sum_{(d_i,d_j)\in I_q} \{\left[\log h_j \sum_{\tau:\tau<j} \log(1-h_\tau)\right] - \sum_{\tau:\tau<i} \log(1-h_\tau)\}$$

- over the r1 query

$$L_{\text{pair(r1)}} = \log \prod_{(d_i,d_j)\in I_q} P(z=i|z\geq j,x;\theta) = \log \prod_{(d_i,d_j)\in I_q} \frac{P(z=i|x;\theta)}{P(z\geq j|x;\theta)}$$

$$= \sum_{(d_i,d_j)\in I_q} \{\left[\log h_i \sum_{\tau:\tau<l} \log(1-h_\tau)\right] - \sum_{\tau:\tau<j} \log(1-h_\tau)\}$$

Pair-wise Loss Function

- over the r2 query

$$L_{\text{pair}(o0)} = -\log \prod_{(d_i,d_k)\in I_q} P(z \geq k | z \geq i, x; \theta) = -\log \prod_{(d_i,d_j)\in I_q} \frac{P(z \geq k|x;\theta)}{P(z \geq i|x;\theta)}$$

$$= -\sum_{(d_i,d_k)\in I_q} \{\sum_{\tau:\tau<k} \log(1-h_\tau) - \sum_{\tau:\tau<i} \log(1-h_\tau)\}$$

Content

- Problem Background
    - Unbiased Ranking
    - Current Challenges
- Architecture
    - Deep Recurrent Survival Model
    - Point-wise Loss Function
    - Pair-wise Loss Function
- **Experiment**
- Conclusion

Experiment

Table 2: Comparison of different unbiased learning-to-rank methods under Yahoo Search Engine and Alibaba Recommender System. CCM is utilized as click generation model. * indicates p-value < 0.001 in significance test vs the best baseline.

Ranker	Debiasing Method	Yahoo Search Engine (CCM)				Alibaba Recommender System (CCM)			
		MAP	NDCG@1	NDCG@3	NDCG@5	MAP	NDCG@1	NDCG@3	NDCG@5
DRSR (Ours)	Labeled Data	0.861	0.747	0.759	0.771	0.850	0.737	0.741	0.755
	Pairwise Debiasing	**0.842***	**0.719***	**0.721***	**0.737***	**0.831***	**0.684***	**0.685***	**0.707***
	Pointwise Debiasing	**0.839***	**0.713***	**0.717***	**0.730***	**0.830***	**0.682***	**0.684***	**0.706***
	Regression-EM [43]	0.829	0.679	0.685	0.701	0.820	0.657	0.668	0.673
	Click Data	0.817	0.636	0.652	0.667	0.810	0.613	0.627	0.658
LambdaMART	Labeled Data	0.854	0.745	0.745	0.757	0.847	0.729	0.732	0.743
	Ratio Debiasing [17]	0.830	0.688	0.685	0.699	0.821	0.661	0.669	0.674
	Regression-EM [43]	0.826	0.669	0.676	0.691	0.818	0.636	0.651	0.667
	Click Data	0.813	0.628	0.646	0.673	0.804	0.603	0.618	0.646
DNN	Labeled Data	0.831	0.677	0.685	0.705	0.824	0.674	0.679	0.693
	Dual Learning Algorithm [2]	0.825	0.672	0.678	0.691	0.814	0.629	0.647	0.674
	Regression-EM [43]	0.823	0.665	0.669	0.687	0.813	0.628	0.645	0.672
	Click Data	0.809	0.611	0.619	0.648	0.801	0.600	0.612	0.641

[Regression-EM] [Xuanhui Wang, et al.] a regression-based EM method where position bias is estimated from regular production clicks.

[Dual Learning Algorithm] [Qingyao Ai, et al.] a dual learning framework jointly learning a ranker and debiasing click data.

[Ratio Debiasing] [Ziniu Hu, et al.] an unbiased pair-wise learning-to-rank based on inverse propensity weight.

[Point-wise Debiasing] our model in point-wise setting.

[Pair-wise Debiasing] our model in pair-wise setting.

[Click Data] the raw click data without debiasing to train the ranker.

[Labeled Data] the human annotated relevance labels without any bias.

Ref: Xuanhui Wang, et al. Position bias estimation for unbiased learning to rank in personal search. WSDM, 2018.
Ref: Qjngyao Ai, et al. Unbiased Learning to Rank with Unbiased Propensity Estimation. SIGIR, 2018.
Ref: Ziniu Hu, et al. Unbiased LambdaMART: An Unbiased Pairwise Learning-to-Rank Algorithm. WWW, 2019.

Experiment

Table 3: Comparison with PBM as click generation model. Notations are same with Table 2.

Yahoo Search Engine (PBM)				
Ranker	MAP	NDCG@1	NDCG@3	NDCG@5
DRSR (Ours)	0.861	0.747	0.759	0.771
	0.848*	**0.726***	**0.737***	**0.745***
	0.843*	**0.723***	**0.731***	**0.740***
	0.834	0.698	0.705	0.712
	0.825	0.671	0.679	0.693
LambdaMART	0.854	0.745	0.745	0.757
	0.836	0.717	0.716	0.728
	0.830	0.685	0.684	0.700
	0.820	0.658	0.669	0.672
DNN	0.831	0.677	0.685	0.705
	0.828	0.674	0.683	0.697
	0.829	0.676	0.684	0.699
	0.819	0.637	0.651	0.667

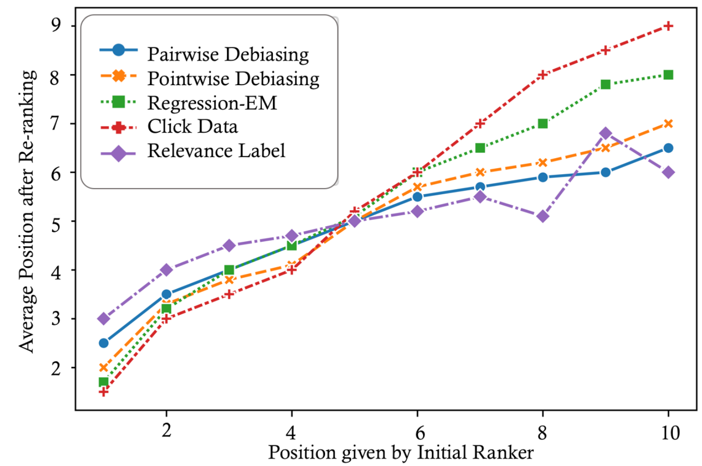


Figure: Average positions after re-ranking of documents at each original position by different debiasing methods combined with DRSR.

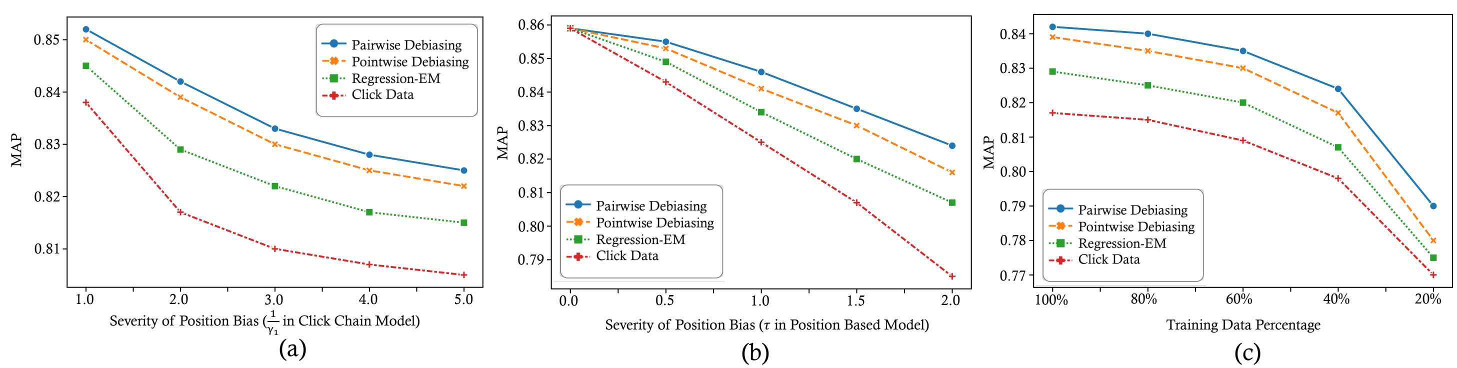


Figure: (a) & (b) Performance of DRSR against other debiasing methods with different degrees of position bias. (c) with different sizes of training data.

Content

- Problem Background
  - Unbiased Ranking
  - Current Challenges
- Architecture
  - Deep Recurrent Survival Model
  - Point-wise Loss Function
  - Pair-wise Loss Function
- Experiment
- **Conclusion**

Conclusion

- We adopt survival analysis techniques accompanied with probability chain rule to derive the joint probability of user various behaviors.
- This framework enables unbiased model to leverage the contextual information in the ranking list.
- Also, we incorporate with survival analysis, and thus can model the non-click queries as the censored click logs.
- DRSR can be easily adopted in the pair-wise loss setting to capture relative relevance between trusted function and untrusted observation.

Thanks for Your Listening

Slides Link

Jerry's Wechat

Full Paper Link

Email: jinjiarui97@gmail.com

Website: https://jiaruijin.com/