sigir21

Causal Intervention for Leveraging Popularity Bias in Recommendation

Yang Zhang[1], Fuli Feng[2], Xiangnan He[1], Tianxin Wei[1],
Chonggang Song[3], Guohui Ling[3], and Yongdong Zhang[1]

[1] University of Science and Technology of China
[2] National University of Singapore
[3] WeChat, Tencent

Tencent 腾讯

Outline

sigir21

- Introduction
- Method
- Experiments
- Conclusion

Popularity Bias In RecSys

- Definitions [1]:
  - Popularity bias refers to the problem where the recommendation algorithm **favors a few popular items** while not giving deserved attention to the majority of other items.
  - Popularity bias is a well-known phenomenon in recommender systems where popular items are recommended even more frequently than their popularity would warrant, **amplifying** long-tail effects already present in many recommendation domains.

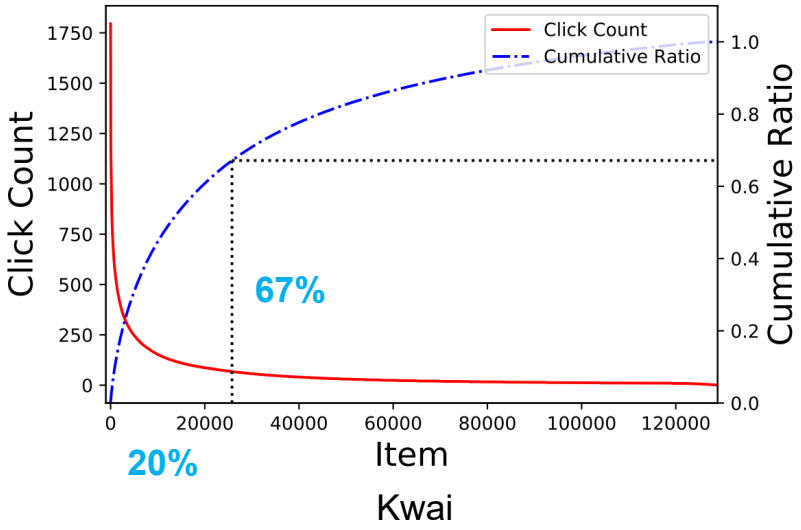
[1] Abdollahpouri, Himan. *Popularity Bias in Recommendation: A Multi-stakeholder Perspective*. Diss. University of Colorado, 2020.

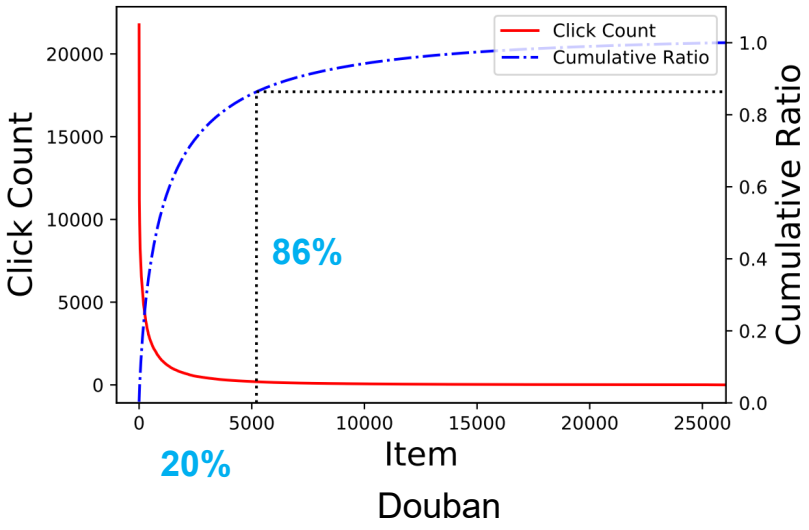
Popularity Bias In RecSys

sigir21

- From data perspective

  The long-tail shape of interactions:





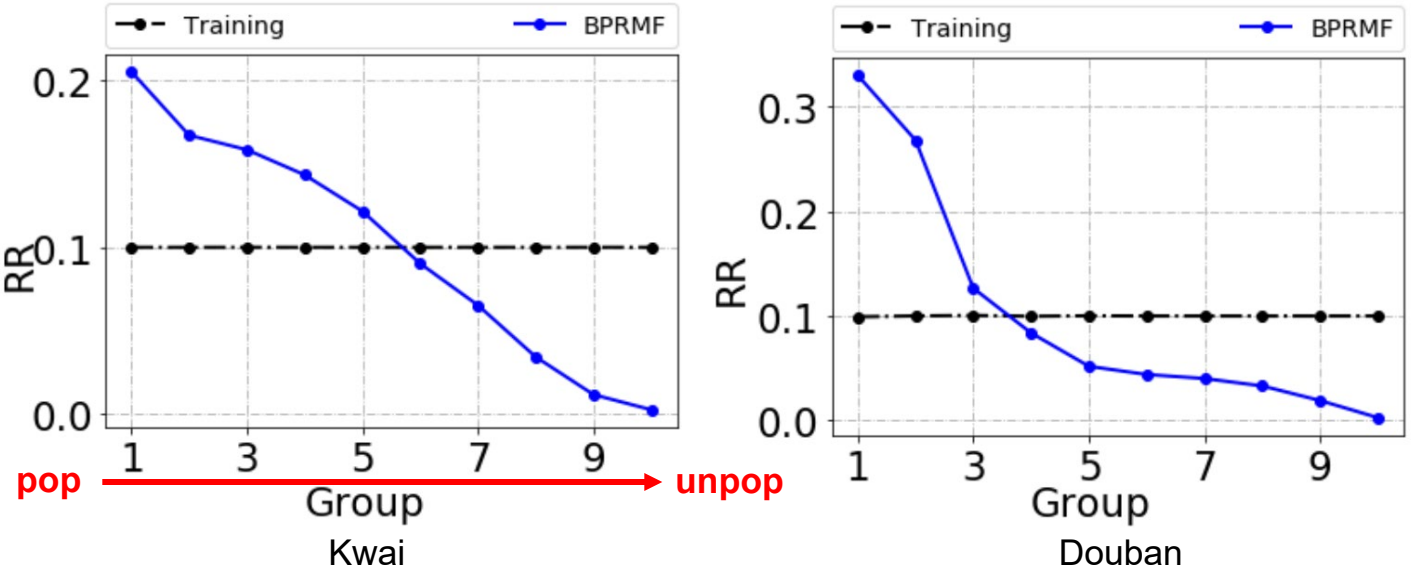
Few popular items: take up the majority of rating interactions
The majority of the items: receive small attention from the users

Popularity Bias In RecSys

sigir21

- From model perspective

Recommendation Ratio(RR) On Kwai and Douban



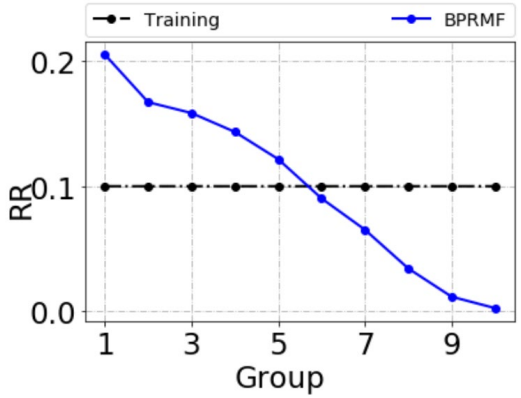
Note: each group has almost the same amount of interactions

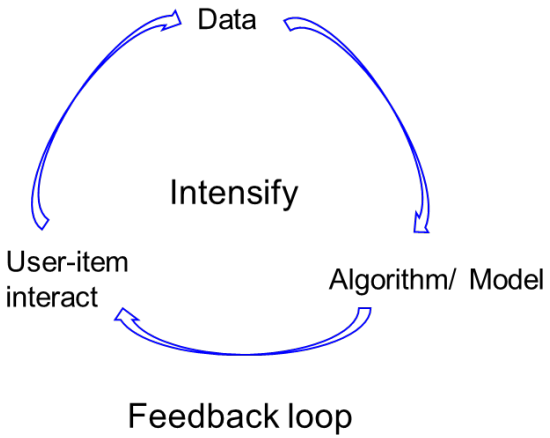
Not only inherit bias from data, but also **amplify** the bias.

Popularity Bias In RecSys

sigir21

- What are the influences?





Matthew effect:
—— the rich get richer and the poor get more poor

Unfairness[1]:
—— Both for user and the provider of items

High-level problem:
—— filter bubble, echo chamber

Must Consider popularity bias !

[1]. Abdollahpouri, Himan, et al. "The Connection Between Popularity Bias, Calibration, and Fairness in Recommendation". RecSys 2020.

[2] Chen et al. arxiv 2021. Bias and Debias in Recommender System: A Survey and Future Directions

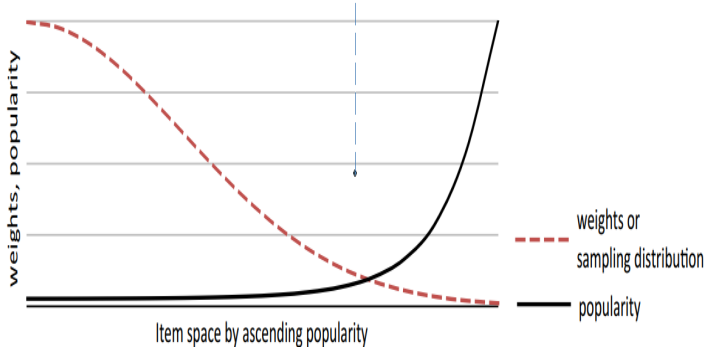
Existing Methods

sigir21

➢ Inverse Propensity Scoring(IPS)

- Basic idea: intervene data distribution by sample reweighting:

$$L_{ips} = \frac{1}{U \cdot I} \sum_{(u,i) \in_{DT}} \frac{1}{ps(u,i)} \delta(y_{ui}, \hat{y}_{ui})$$



- Properly defining propensity scores can lead to unbiased estimator

$$L_{ideal} = \frac{1}{U \cdot I} \sum_{u \in U, i \in I} \delta(y_{ui}, \hat{y}_{ui})$$

$$\neq E(L_{naive}) = E\left(\frac{1}{|\{(u,i): O_{ui} = 1\}|} \sum_{u \in U, i \in I} \delta(y_{ui}, \hat{y}_{ui})\right)$$

$$= E(L_{ips}) = \frac{1}{U \cdot I} \sum_{u \in U, i \in I} E_{O_{ui}} \frac{O_{ui}}{ps(u,i)} \delta(u,i) = L_{ideal}$$

But, finding good propensity is not easy.

Subjected to high variance.

Schnabel, Tobias, et al. "Recommendations as treatments: Debiasing learning and evaluation." ICML, 2016.

Existing Methods

sigir21

➢ Ranking Adjustment

- Basic idea: Intentionally increase the scores of less popular items by reranking or regularization
- Re-ranking

  -- modify the ranking score to adjust the ranking list

$$argmax_i \boxed{\hat{R}_{int}(u,i)} + \lambda \boxed{\hat{R}_{pop}(u,i)}$$

Adjusting score

Example -- popularity compensation[1]

$$\hat{R}_{pop}(u,i) = \alpha \frac{1}{pop_i}\left(\beta \hat{R}_{ui} + 1 - \beta\right), \quad \alpha = \frac{n_u}{m_u}$$ is used to rescale

Heuristically designed, lack theoretical foundations

[1] Ziwei zhu et.al. “Popularity-Opportunity Bias in Collaborative Filtering.” In WSDM 2021.

Existing Methods

sigir21

➢ Causal Embedding

- Utilizing cause-specific data (e.g., uniform data) to guide model learning.

  E.g., Joint training (CausE [1]):

On uniform data On biased data

$$\min_{\mathcal{W}_c,\mathcal{W}_t} \frac{1}{|S_c|}\sum_{(i,j)\in S_c}\ell\left(y_{ij},\hat{y}^c_{ij}\right) + \frac{1}{|S_t|}\sum_{(i,j)\in S_t}\ell\left(y_{ij},\hat{y}^t_{ij}\right) + \lambda_c R\left(\mathcal{W}_c\right) + \lambda_t R\left(\mathcal{W}_t\right) + \lambda^{CausE}_{tc}\left\|\mathcal{W}_t - \mathcal{W}_c\right\|^2_F,$$

Guiding term

Other ways: knowledge distillation [2]

But, obtaining uniform data(bias-free) is not easy.

Such data is much smaller.

[1] Bonner, Stephen et.al. "Causal embeddings for recommendation." In RecSys 2018.
[2] Liu, Dugang, et al. "A general knowledge distillation framework for counterfactual recommendation via uniform data." In SIGIR 2020.

Motivation

- Previous works aim at eliminating the effect of popularity bias. (some take even state as goal)
- But, no all pop biases in the data are bad!!
  - E.g., some items have higher popularity because of better quality.
  - Such patterns are beneficial for better predicting future interactions.
- How to REMOVE bad effect of pop bias & INJECT desired bias in model serving?
  - What's the bad effect?

    Causality tell us
  - How ?

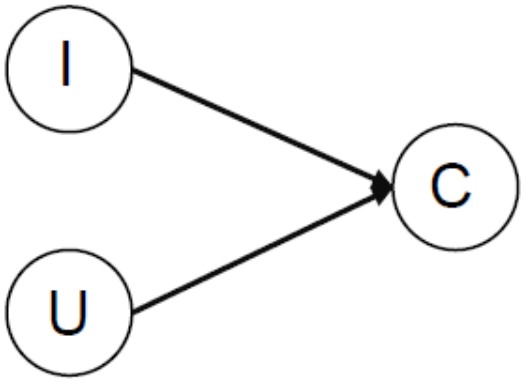
    Tools for Causality

Outline

- Introduction
- **Method**
- Experiments
- Conclusion

Causal Story

➢ Traditional Causal graph for recommendation



Traditional graph

U: user I: item C: interaction (click)

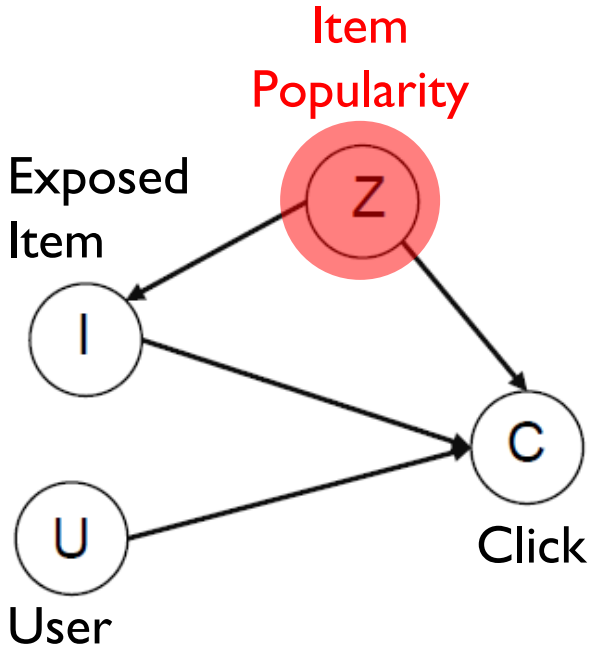
- Story: User-item matching predicts/generates affinity score
- Represent the model structure
- Represent the assumption of data generation process
- Problem:

  popularity affect the process, but isn't considered!

Causal Story

sigir21

➢ Our causal graph for recommendation



Our assumption

we assume users can only interact with exposed item

- (U, I, Z) → C :

  C is determined by the factors U,I, and Z.

  - **(U, I) → C**: same to traditional graph, user-item matching
  - Z → C: many users have the herd mentality (follow the majority to consume pop items)
- Z → I:

  item popularity affects the exposure of items

Z is a common cause of I and C

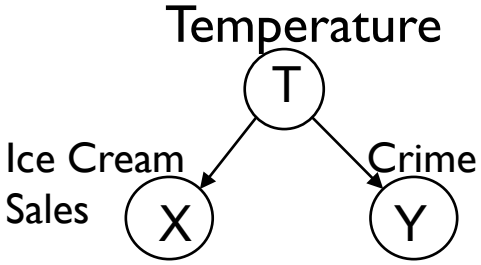
⇩

Z is a confounder between I and C

Popularity De-confounding(PD)

sigir21

- Z is a confounder, bringing spurious (bad effect) correlation between I and C.

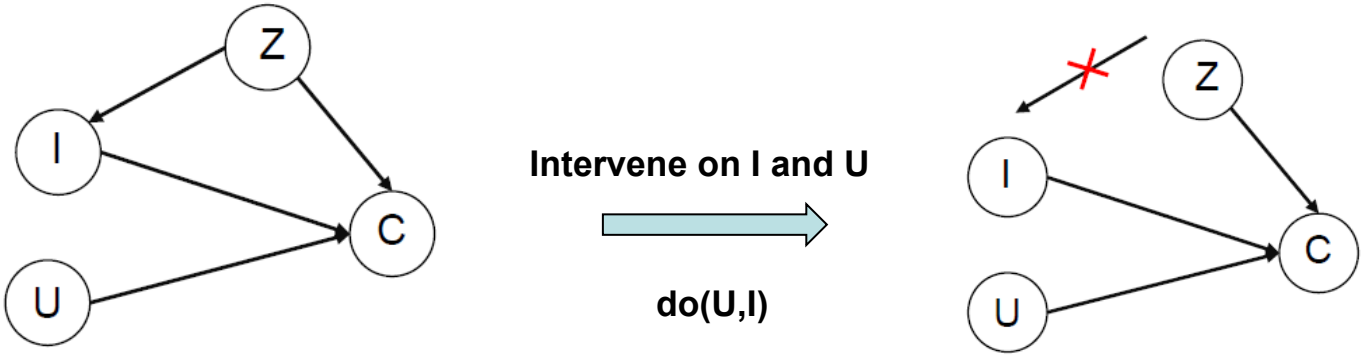


There is no causal relation between X and Y

But there is still a correlation between X and Y

This is a spurious correlation

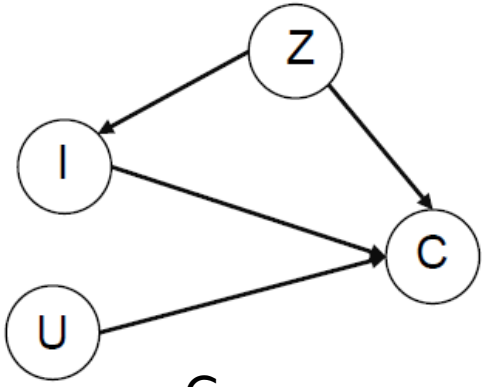
- P(C|U,I) cannot reflect the true interest because of the confounding effect!
- Take P(C|do(U,I)) instead of P(C|U,I) to estimate interest!



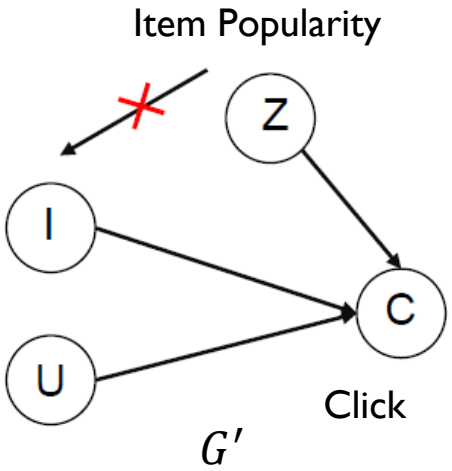
Popularity De-confounding(PD)

sigir21

- P(C|do(U,I)) Vs P(C|U,I)



G



G'

Traditional methods estimate P(C|U,I)

- Contains spurious correlation due to confounder

$$P(C|U,I) \overset{(1)}{=} \sum_Z P(C,Z|U,I)$$

$$\overset{(2)}{=} \sum_Z P(C|U,I,Z)P(Z|U,I)$$

$$\overset{(3)}{=} \sum_Z P(C|U,I,Z)P(Z|I)$$

Cause Bad Effect

$$\overset{(4)}{\propto} \sum_Z P(C|U,I,Z)P(I|Z)P(Z)$$

We estimate P(C|do(U,I))

- Removes the hidden confounder

$$P(C|do(U,I)) \overset{(1)}{=} P_{G'}(C|U,I)$$

$$\overset{(2)}{=} \sum_Z P_{G'}(C|U,I,Z)P_{G'}(Z|U,I)$$

$$\overset{(3)}{=} \sum_Z P_{G'}(C|U,I,Z)P_{G'}(Z)$$

$$\overset{(4)}{=} \sum_Z P(C|U,I,Z)P(Z),$$

Popularity De-confounding(PD)

sigir21

- Estimate P(C|do(U,I))

$$P(C|do(U,I)) = \sum_z P(C|U,I,z)P(z)$$

Step 1. estimate P(C|U,I,Z)

Model:

$$P(c|U=u, I=i, Z=m_i^t) := P_\Theta(c|u,i,m_i^t) := ELU'(f_\Theta(u,i)) \times (m_i^t)^\gamma$$

m_i^t: popularity of item i at timestamp t

γ: smooth popularity and control the strength of conformity effect

activation

$f \times m$: disentangle and simplify

user-item matching, we take MF

learning:

$$\text{minimize}_\Theta \sum_{(u,i,m_i^t)\in D} l_0(P_\Theta(c|u,i,m_i^t))$$

l_0 can be BPR or log loss.

Popularity De-confounding(PD)

sigir21

- Estimate P(C|do(U,I))

$$P(C|do(U,I)) = \sum_z P(C|U,I,z)P(z)$$

Step 2. compute $P(C|do(u,i))$

$$\begin{aligned}
&\sum_z P_\Theta(c|u,i,z)P(z) \\
&= \sum_z ELU'\big(f_\Theta(u,i)\big) \times z^\gamma P(z) \\
&= ELU'\big(f_\Theta(u,i)\big) \times \sum_z z^\gamma P(z) \\
&= ELU'\big(f_\Theta(u,i)\big) E(Z^\gamma)
\end{aligned}$$

$E(Z^\gamma)$ is a constant, which will not change ranking

$ELU'\big(f_\Theta(u,i)\big)$ is an approximation.

So, ranking with $ELU'\big(f_\Theta(u,i)\big)$

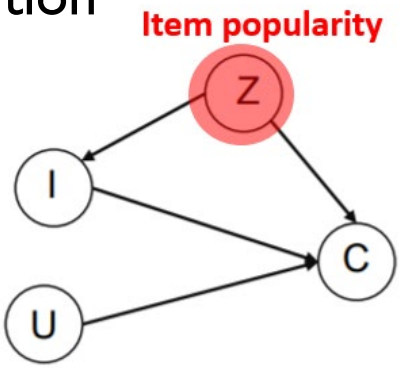
Popularity De-confounding & Adjusting

- We have estimated $P(C|do(U,I))$, which does not chase even state but the real interests.
- Is It enough?
  - No, there is another demand --- inject some desired popularity bias into recommendation.
  - Such as, recommend more items that have potential to be popular in the future if we can know this knowledge.

- Inference: Popularity Adjusting (inject desired popularity bias)
  - Inject the desired pop bias $\tilde{Z}$ by causal intervention

$$P(C|do(U,I),do(Z=\tilde{z}))$$

$$\Rightarrow \quad f_\Theta(u,i) \times (\tilde{m}_i)^\gamma$$



$\tilde{m}_i$: pop of item i under the desired bias

Desired popularity bias: predict by trends of popularity

Outline

- Introduction
- Method
- **Experiments**
- Conclusion

Experiments

➢ Experimental Setting

☐ Datasets:

Dataset	#User	#Item	#Interaction	#Sparsity	#type
Kwai	37,663	128,879	7,658,510	0.158%	Click
Douban	47,890	26,047	7,174,218	0.575%	Review
Tencent	80,339	27,070	1,816,046	0.084%	Like

☐ Data Splitting:

Split each dataset into 10 time stages regarding timestamp.

0-8th stages: training

9th stage: validation & testing.

☐ Evaluation Setting:

PD: directly test

PDA: Most recent stages can be utilized to predict future popularity.

Experiments

sigir21

➢ Baselines

- For PD (Not inject desired popularity bias):
  - MostPop, BPRMF
  - xQuad (2019FLAIRS) – ranking adjustment
  - BPR-PC (2021WSDM) – ranking adjustment
  - DICE (2021WWW) – causal embedding
  - PD: based on MF.
- For PDA (inject desired popularity bias):
  - MostRecent (2020SIGIR) -- local popularity
  - BPR(t)-pop (2017 Rectemp@RecSys) – Dynamic model
  - BPRMF-A – BPRMF + adjusting $(\times pop^{\gamma})$
  - DICE-A – DICE + adjusting $(\times pop^{\gamma})$
  - PDA : based on MF

Experiments for PD

sigir21

➢ Results for PD

Data / Method	Kwai		Douban		Tencent	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
MostPop	0.0014	0.0030	0.0218	0.0349	0.0145	0.0093
BPRMF	0.0054	0.0067	0.0274	0.0405	0.0553	0.0328
xQuad	0.0054	0.0068	0.0274	0.0391	0.0552	0.0326
BPR-PC	**0.0070**	**0.0072**	**0.0282**	0.0381	**0.0556**	**0.0331**
DICE	0.0053	0.0067	0.0273	**0.0421**	0.0516	0.0312
PD(ours)	0.0143	0.0177	0.0453	0.0607	0.0715	0.0429

- PD has better performance, because of the better estimation of interest with removing the spurious correlation.
- DICE& BPR-PC can not bring great improvement in such testing setting.
- Different improvements, because of different characteristics of datasets

Experiments for PD

sigir21

➢ PD —— Recommendation Analysis.

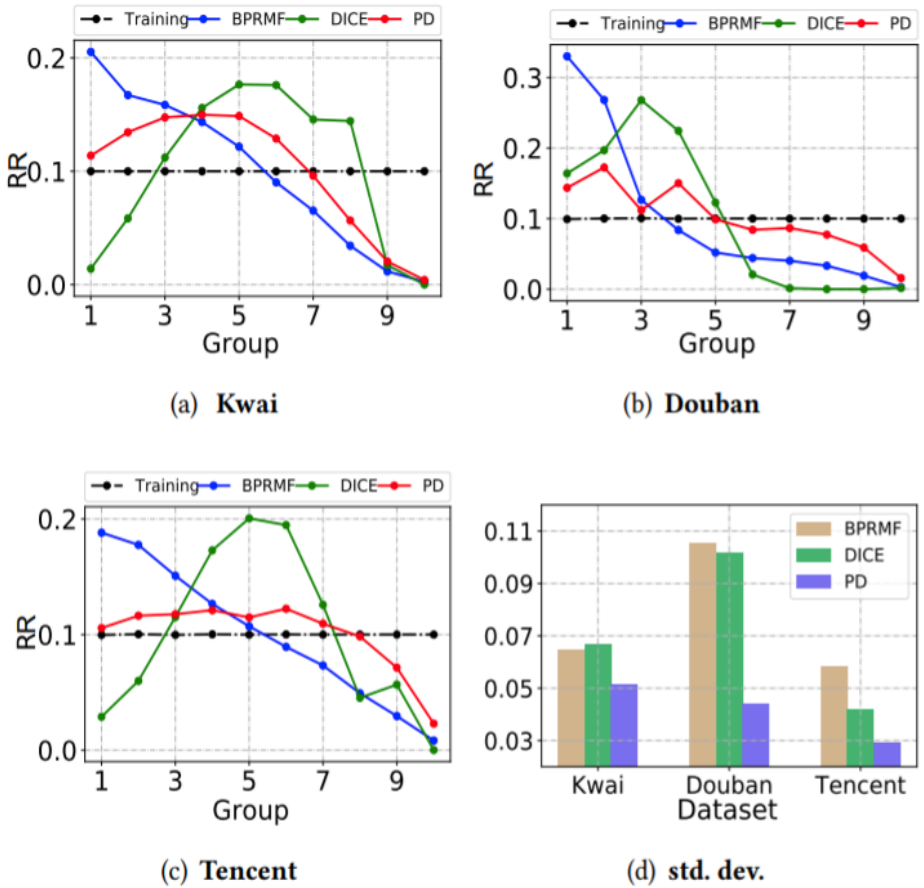


Figure 4: Recommendation rate(RR) over item groups.

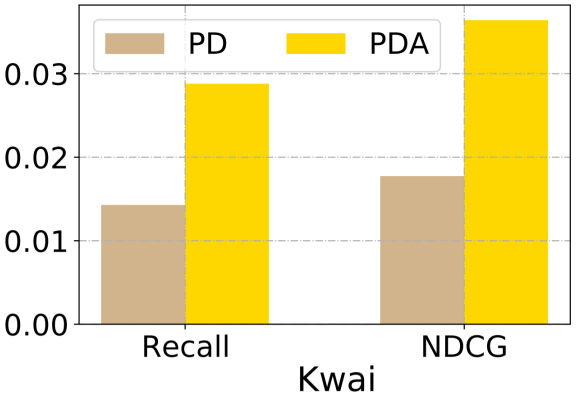
- Less amplification for most popular groups compared with BPRMF
- Do not **over-suppress** the most popular groups compared with DICE
- More flat lines and standard deviations over different groups
  --- relative fair recommendation opportunities for different group (refer to training set)
- Meanwhile, better performance
  --- only remove bad effect to improve model performance

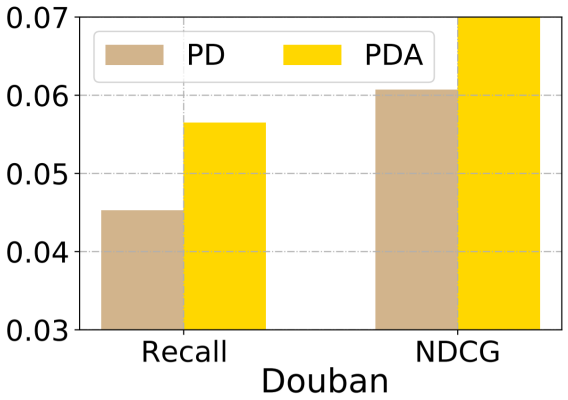
Experiments for PDA

sigir21

➢ Results for PDA

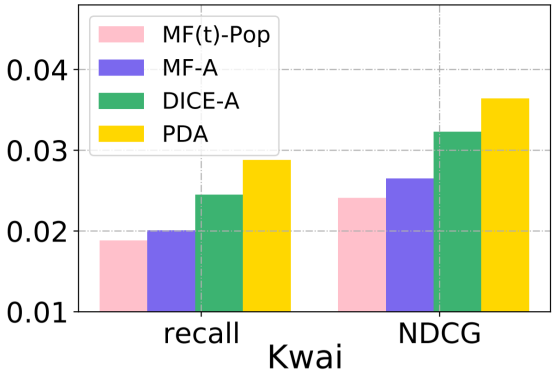
- Compared with model that not inject the bias

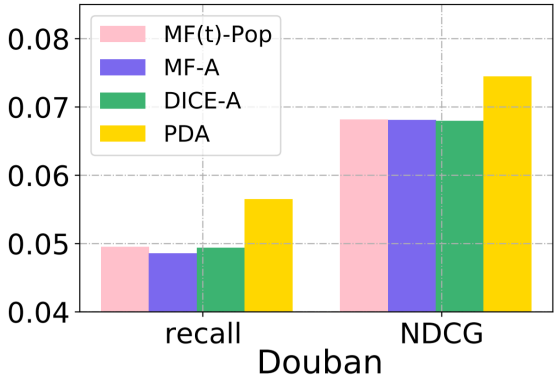




Injecting the desired bias is very valuable!

- Compared with baselines that also inject the bias





PDA injects the desired bias better via intervention.

Conclusion

- Conclusion
  - Consider popularity influence in Causal graph.

    -- treat it as confounder
  - Estimate P(C|do(U,I)) instead of P(C|U,I))
  - Leveraging popularity bias instead of blindly eliminating it.
- Future works
  - Better methods for estimating causal-effect.

    -- unbalance of different Z.

    existing: representations balance, and separate models
  - Consider bias problems with content features.

Thanks!

Q & A