LLM, Yet Another Solution to RecSys?

Dr. Sun Aixin 孙爱欣
NTU Singapore

DataFunSummit # 2023

NANYANG TECHNOLOGICAL UNIVERSITY

DataFun.

What is RecSys?

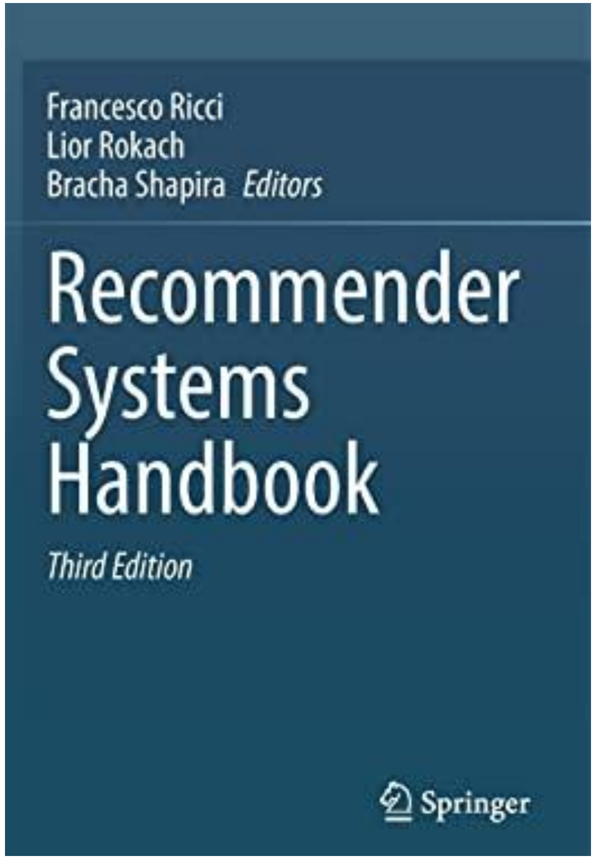
Data	Static/offline	Stream/online
Evaluation	Train/test split	A/B testing
Metric	HitRate, NDCG…	CTR, CVR, GMV…
Model	A single model	Mixture of models?

LLM?

- RecSys evaluation is challenging

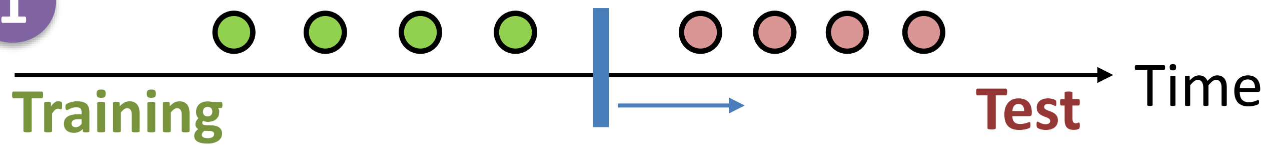
- “The goal of the offline experiments is to filter out inappropriate approaches, leaving a relatively small set of candidate algorithms to be tested” online

- “it is necessary **to simulate the online process** where the system makes predictions or recommendations”



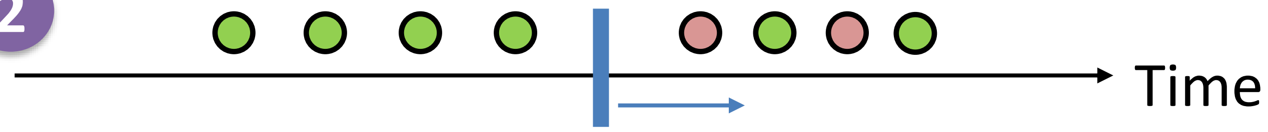
The 5 settings in offline evaluation

1



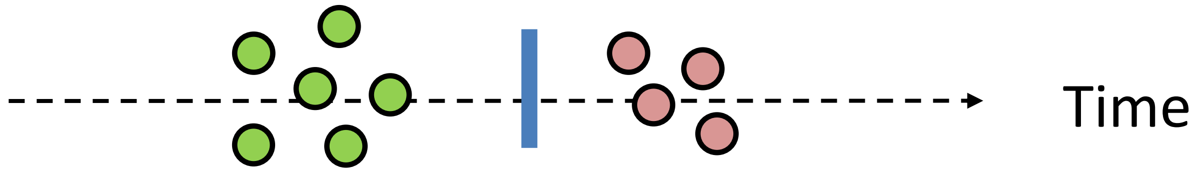
Sliding

2



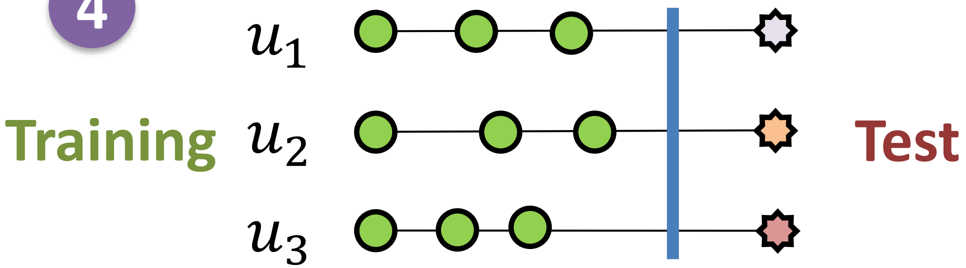
Sliding

3



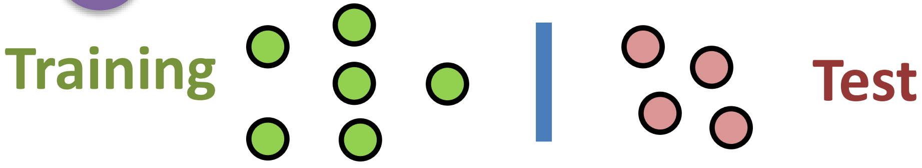
A sampled timestamp

4



Leave-one-out

5



Random split

Case study: what train/split?

- Collection: 88 papers in RecSys conferences (2020 – 2022)

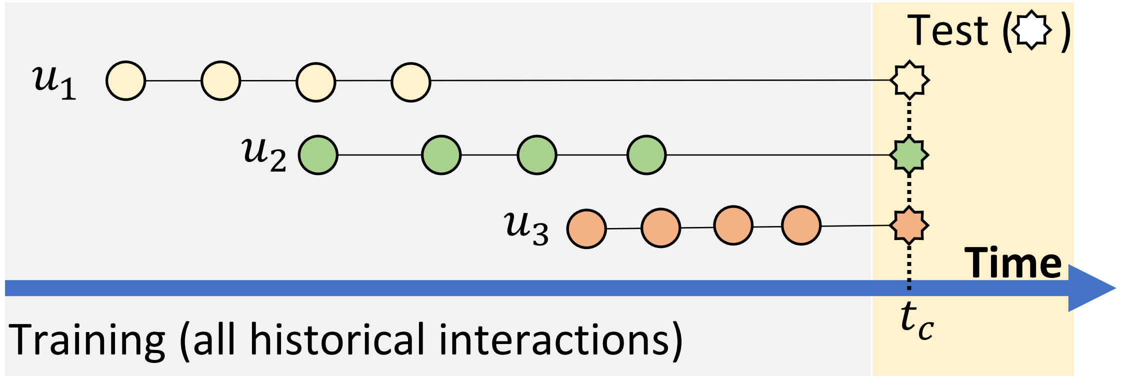
No. papers	Percentage	Train/test split	Global timeline?
30	**34%**	Random split	No
22	**25%**	Leave-one-out	No
17	19.5%	Single time point	Partially
15	17%	Simulation-based online	Yes
4	4.5%	Sliding window	Yes

Bandits and reinforcement learning for recommendation.
Incremental learning or session-based learning.

Recommendation in practice

- Users get recommendations when visiting a site or app, at current time t_c

- All historical interactions before t_c can be used as training data



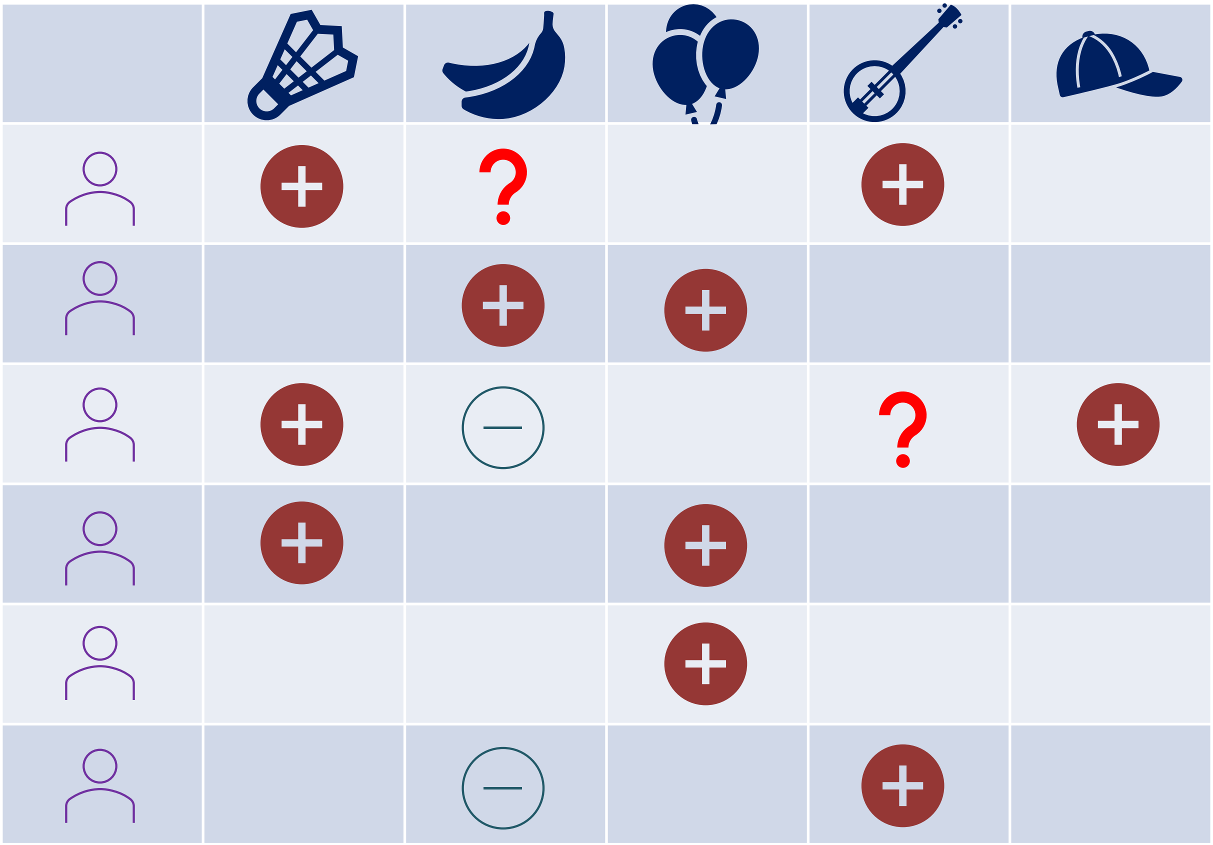
- Learning from ***past interactions***

- To ***predict*** users' preferred items ***in (near) future***

RecSys in academic research: problem abstraction

One problem definition for many RecSys tasks

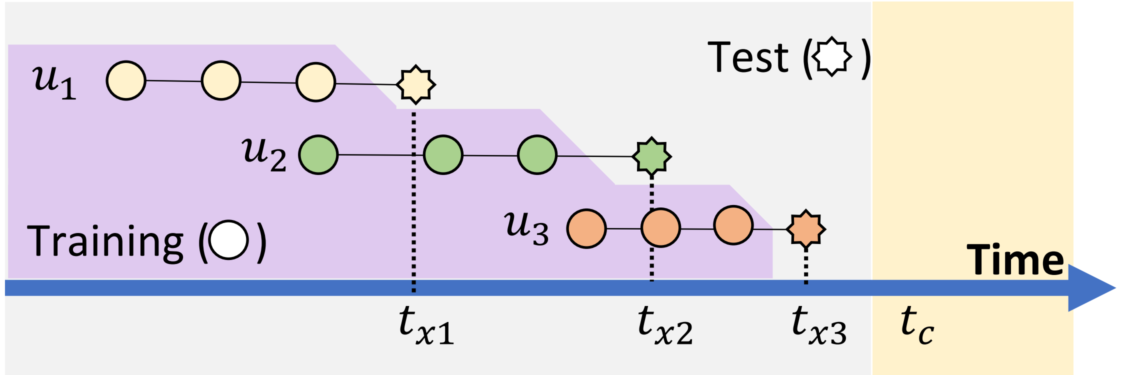
Global timeline not observed



Ignorance of global timeline: data leakage

- Recommenders access user-item interactions that "would happen" after the test time point
- Recommenders may recommend "future items"
- Recommendation accuracies may not mean much

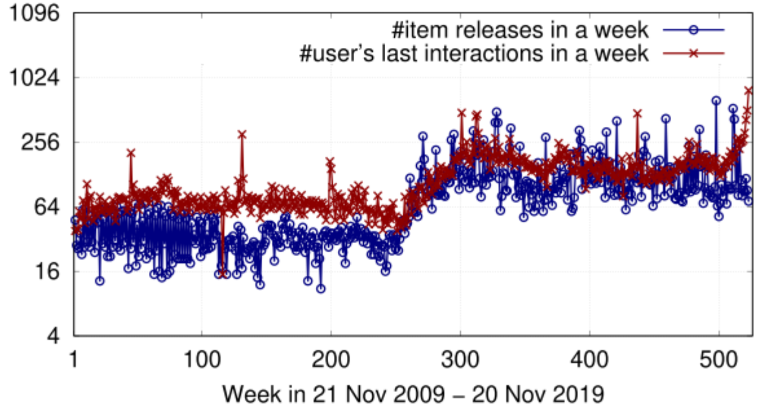
An illustration: Leave-one-out



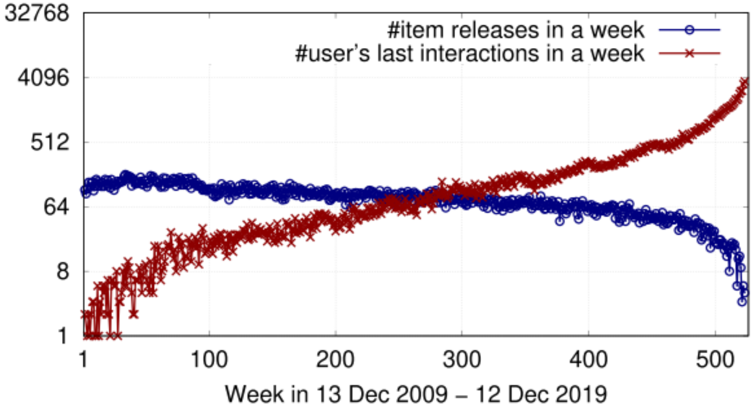
Applicable to Popularity and ML/DL-based models

Global timeline vs local timeline

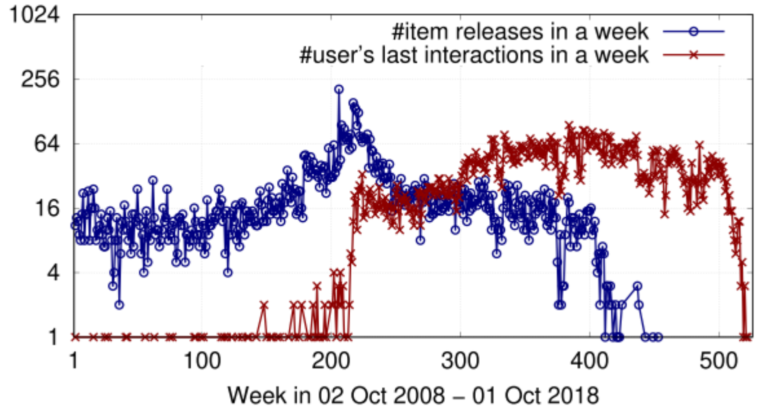
- Number of item first interactions in each week
- Number of user last interactions in each week
- On 4 datasets for 10 years duration



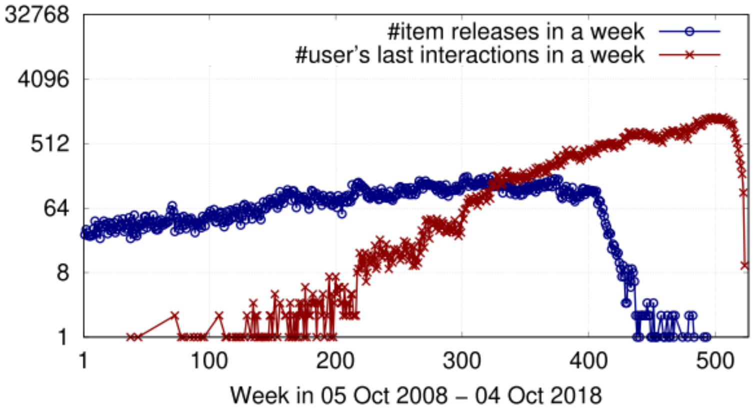
(a) MovieLens-25M



(b) Yelp

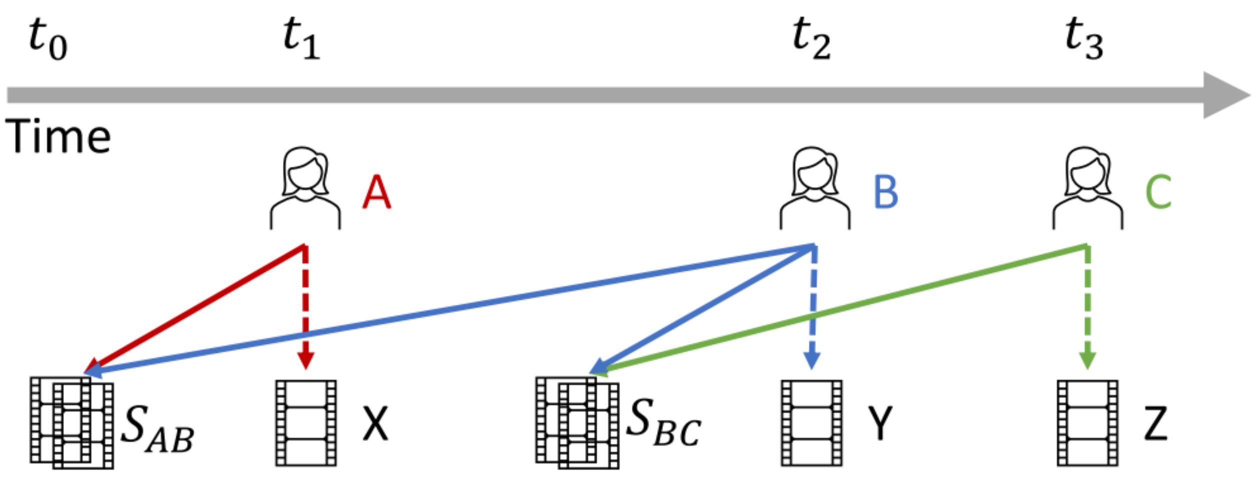


(c) Amazon-music

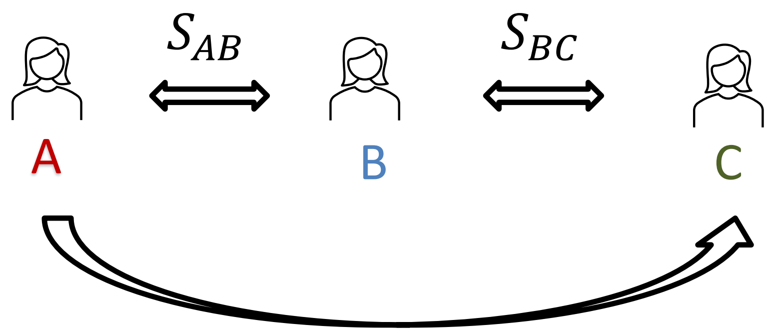


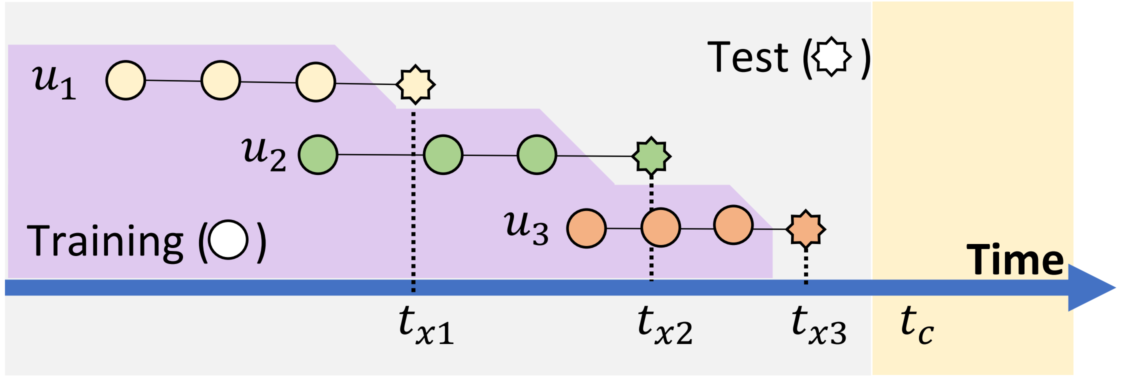
(d) Amazon-electronic

Data leakage in offline evaluation of recommender system



(a) User-item interaction along global timeline.





S_{AB}: items rated by both users A and B
S_{BC}: items rated by both users B and C

X: test instance of user A
Y: test instance of user B
Z: test instance of user C

All interactions by user *C* happened after the test instance of *A*

Experiments: the impact of data leakage

Dataset	Time span selected	Data Filtering	#User	#Item	#Rating	Sparsity
MovieLens-25M	21 Nov 2009 to 20 Nov 2019	No filtering	62,202	56,774	9,808,925	$2.78e-3$
Yelp	13 Dec 2009 to 12 Dec 2019	10-core	116,655	61,027	3,127,215	$4.39e-4$
Amazon-music	02 Oct 2008 to 01 Oct 2018	5-core	15,839	11,071	162,880	$9.29e-4$
Amazon-electronic	05 Oct 2008 to 04 Oct 2018	10-core	141,633	49,325	2,365,483	$3.38e-4$

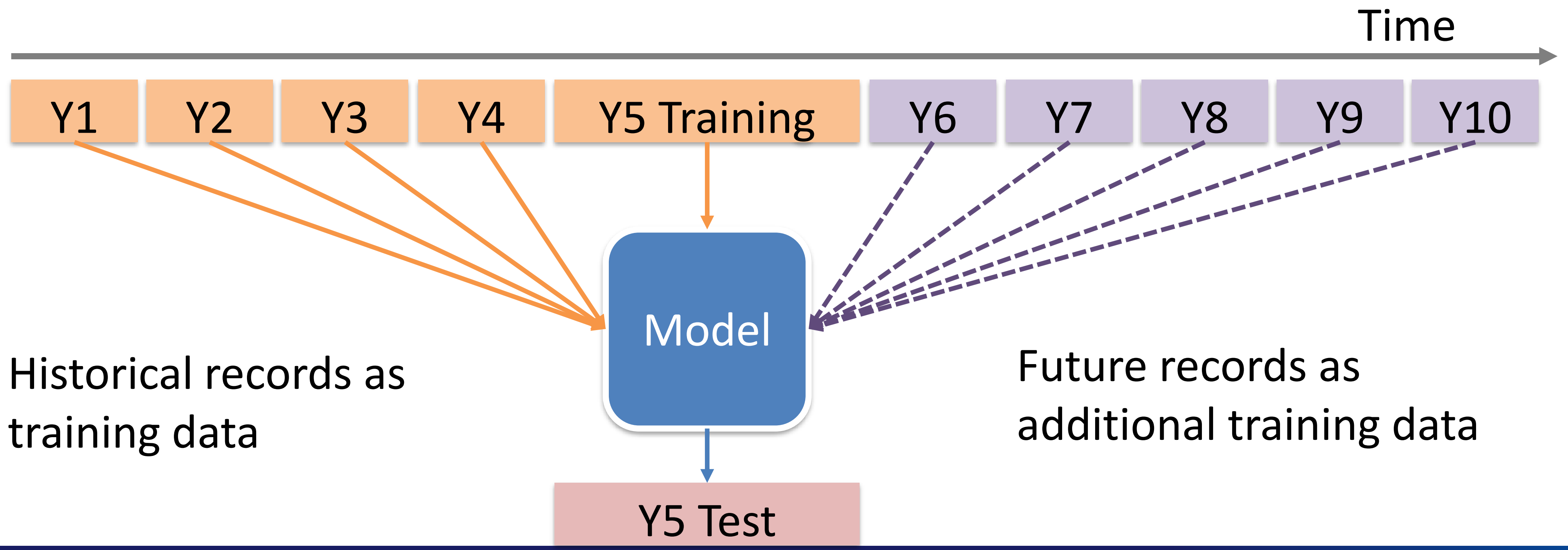
- Data partition: Leave-one-out splitting
- Baselines: BPR, NeuMF, LightGCN, SASRec
- Evaluation metrics: HR@20, NDCG@20

Recommendation List

Recommendation Accuracy

Experiment: to simulate different severity of data leakage

- Test set: test instances that happened in Year 5 (example test year)
- Training set: (Instances before Y5) + (training instances in Y5) + (x year of future instances), $x \in [0,5]$



Impact of data leakage on recommendation list

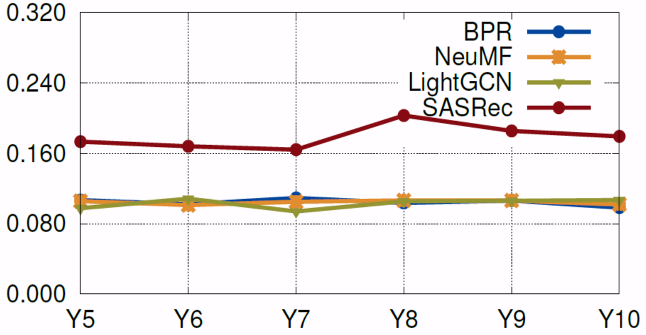
- **Future items**: the items are exclusively available only after the specific time point of a given test instance.
- All models recommend "future items" → **invalid recommendation**

Model	Dataset / Test year	MovieLens-25M Y5	MovieLens-25M Y7	Yelp Y5	Yelp Y7	Amazon-music Y5	Amazon-music Y7	Amazon-electronic Y5	Amazon-electronic Y7
BPR	Y5	0	–	0	–	0	–	0	–
	Y6	0	–	421	–	615	–	79	–
	Y7	22	0	829	0	970	0	363	0
	Y8	7	11	2,365	504	1,101	651	263	200
	Y9	6	88	5,048	287	1,304	1,103	499	1,224
	Y10	4	81	1,851	1,598	1,197	1,155	200	583
NeuMF	Y5	0	–	0	–	0	–	0	–
	Y6	3	–	602	–	910	–	28	–
	Y7	7	0	1,631	0	1,501	0	1,303	0
	Y8	27	31	3,260	130	1,733	878	549	0
	Y9	22	6	3,542	1,177	1,491	1,276	729	216
	Y10	15	1	5,205	1,791	1,577	1,573	2,655	326
LightGCN	Y5	0	–	0	–	0	–	0	–
	Y6	11	–	369	–	626	–	37	–
	Y7	32	0	739	0	1,050	0	148	0
	Y8	116	189	1,070	569	998	632	367	220
	Y9	22	26	1,257	979	1,036	893	262	430
	Y10	15	58	1,103	1,360	1,152	1,029	260	470
SASRec	Y5	0	–	0	–	0	–	0	–
	Y6	315	–	967	–	906	–	216	–
	Y7	442	0	3,074	0	1,548	0	625	0
	Y8	144	489	2,228	2,666	1,814	1,341	487	1388
	Y9	342	403	3,162	2,893	1,982	1,376	20	3,209
	Y10	993	386	1,741	3,014	1,980	1,662	12	2,479

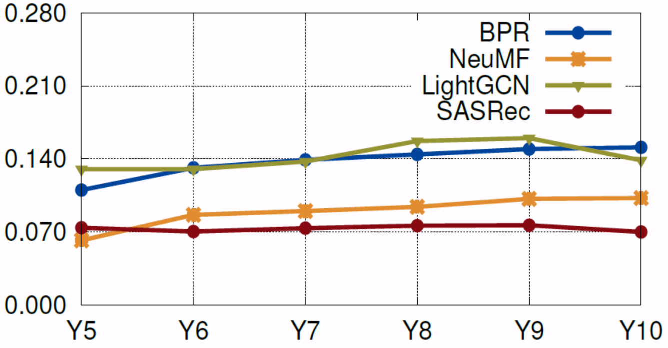
Impact of data leakage on recommendation accuracy

- The impact on recommendation accuracy can vary, and it is not predictable.

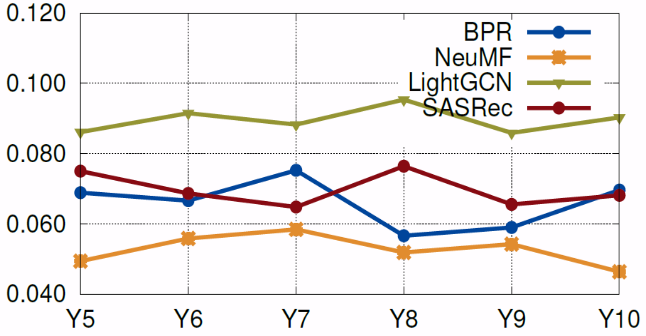
- The **relative performance ordering** of the evaluated models does not exhibit consistent patterns.



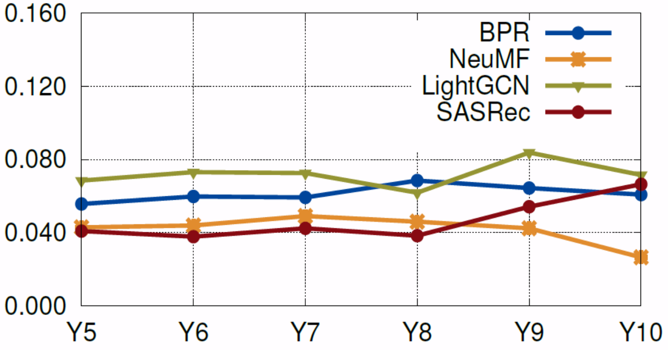
(A) HR@20
MovieLens-25M



(E) HR@20
Amazon-music



(C) HR@20
Yelp



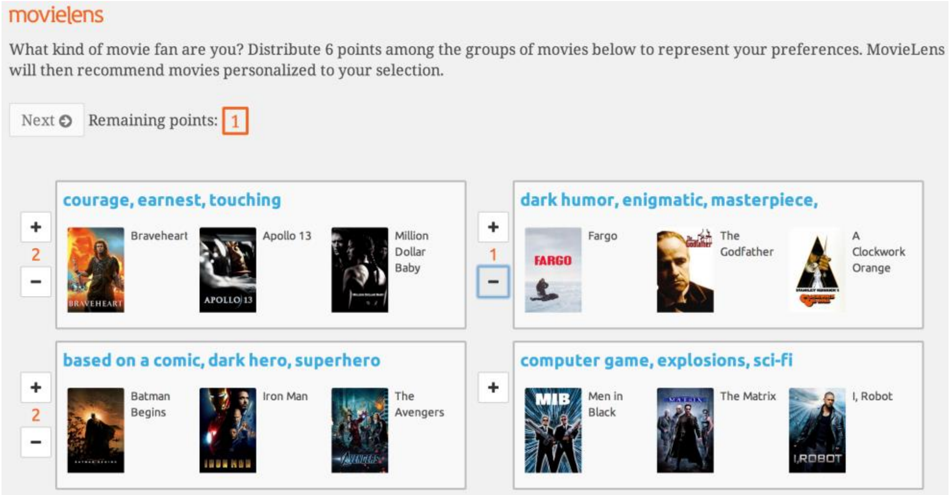
(G) HR@20
Amazon-electronic

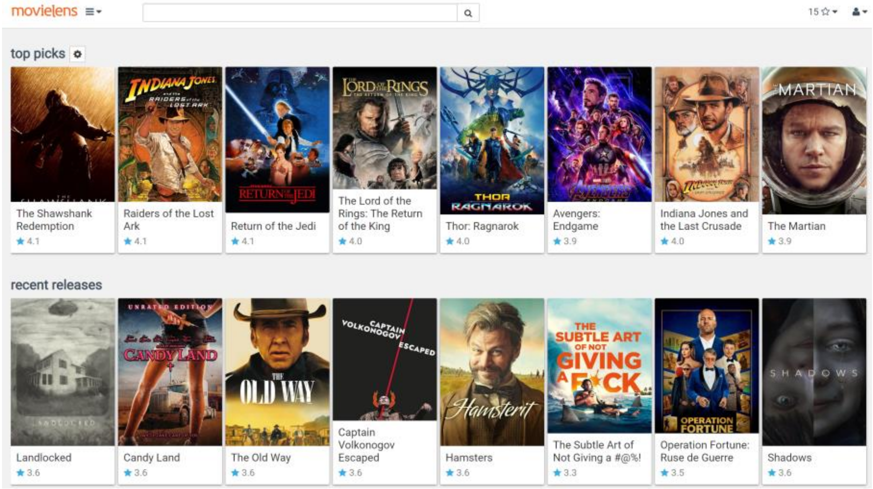
Data

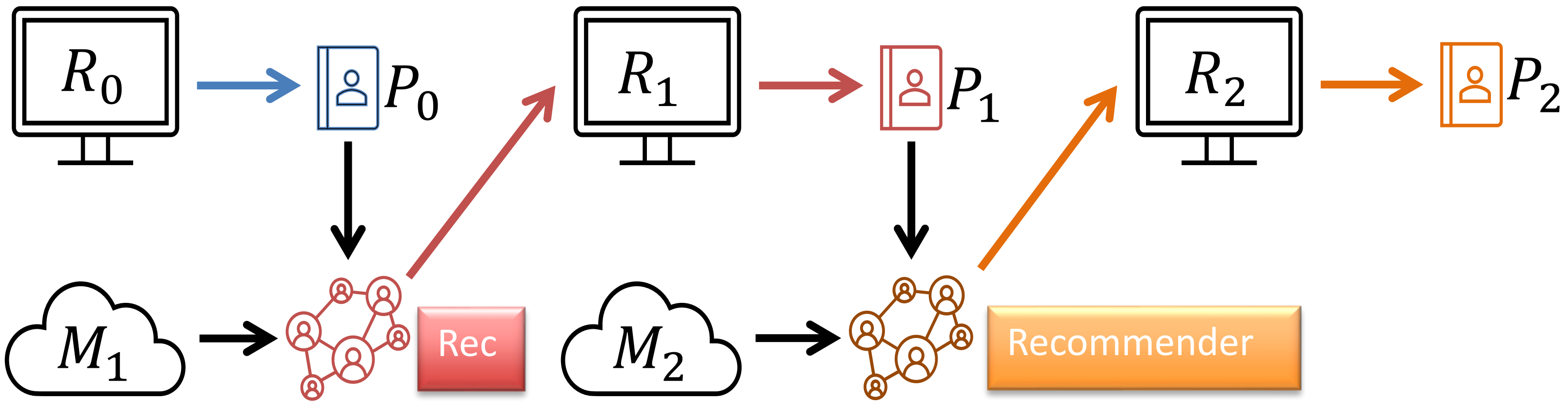
Static/offline dataset

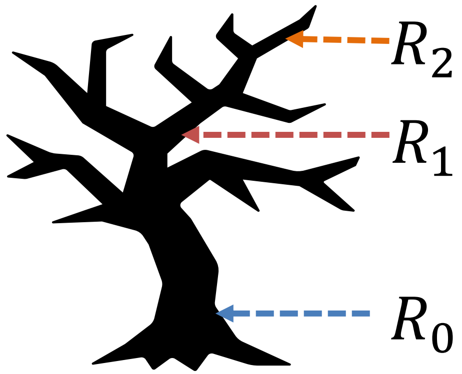
Stream/online data

The MovieLens dataset

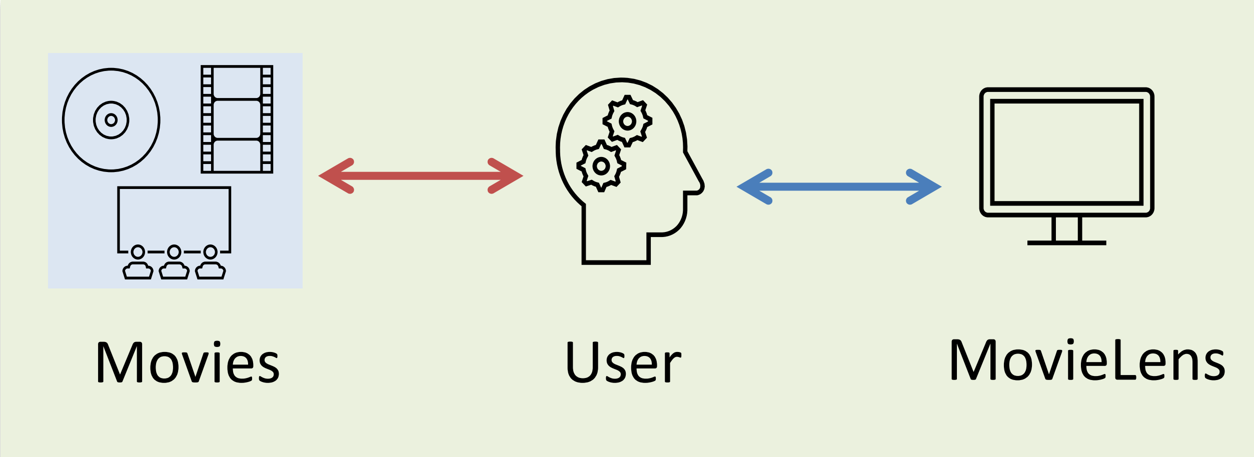






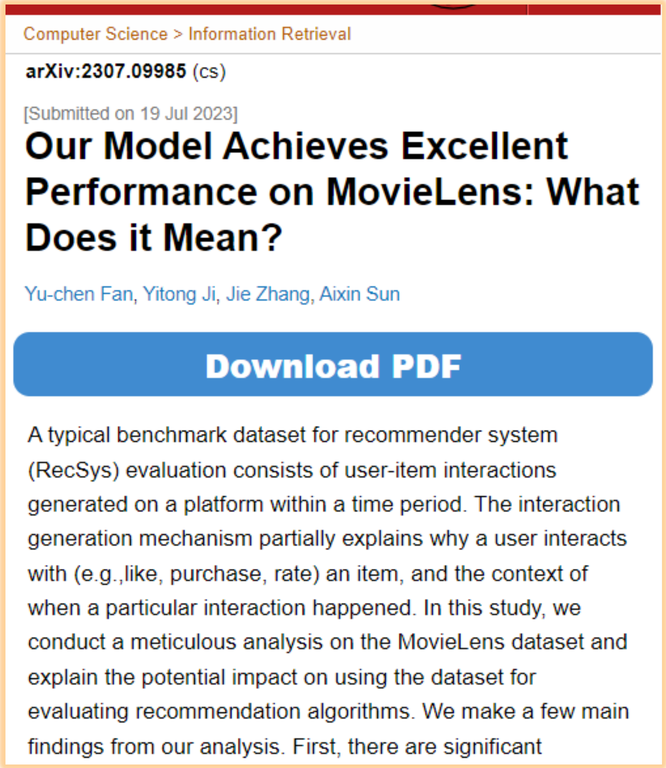


Two kinds of interactions



- **User-Movie Interaction**
  - There is a decision process to decide which movie to watch next
- **User-MovieLens Interaction**
  - MovieLens guides users to recall what movies he/she has watched
  - Cold-start dataset for "static preference"

https://arxiv.org/abs/2307.09985



Computer Science > Information Retrieval

arXiv:2307.09985 (cs)

[Submitted on 19 Jul 2023]

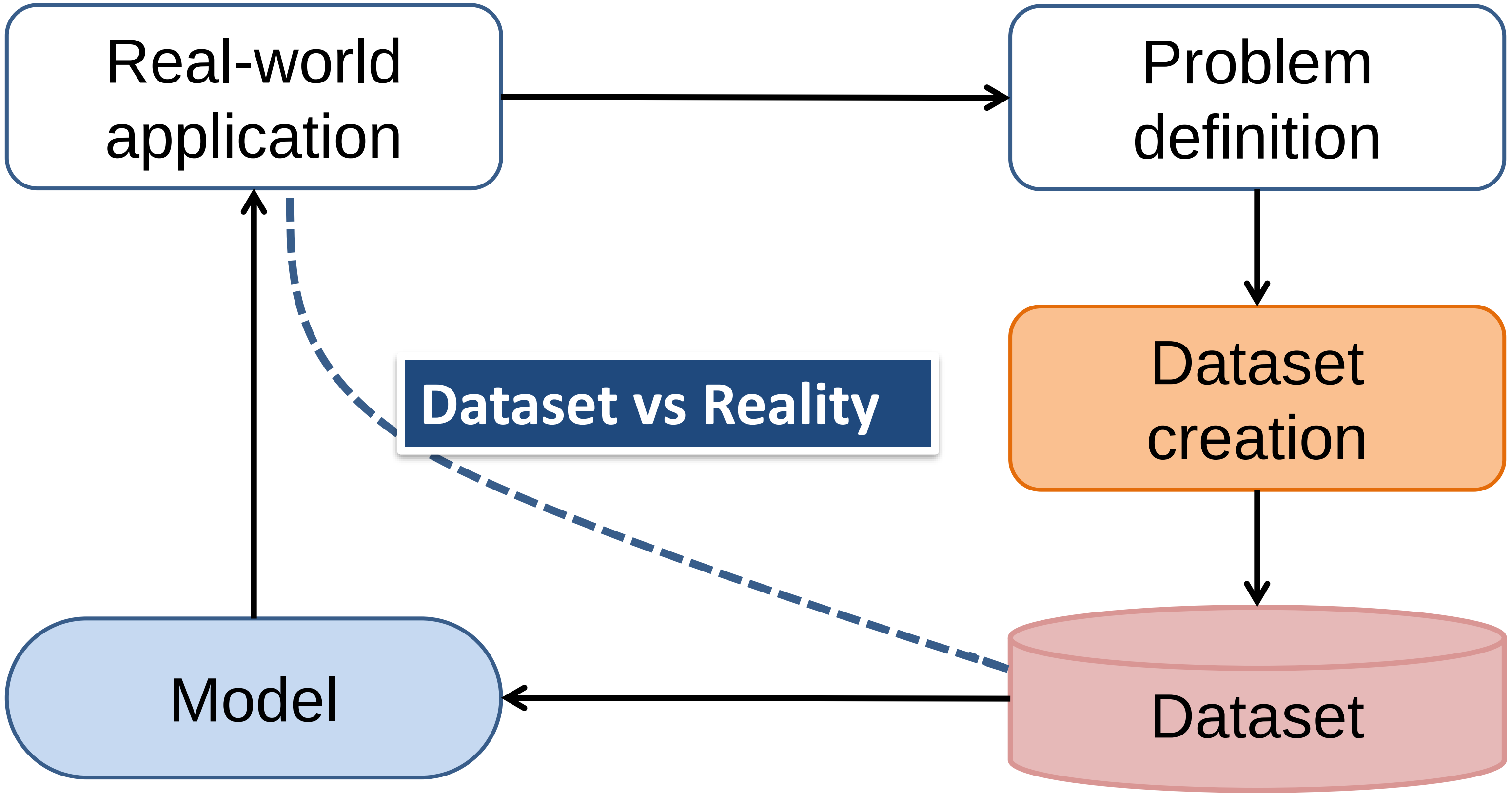
Our Model Achieves Excellent Performance on MovieLens: What Does it Mean?

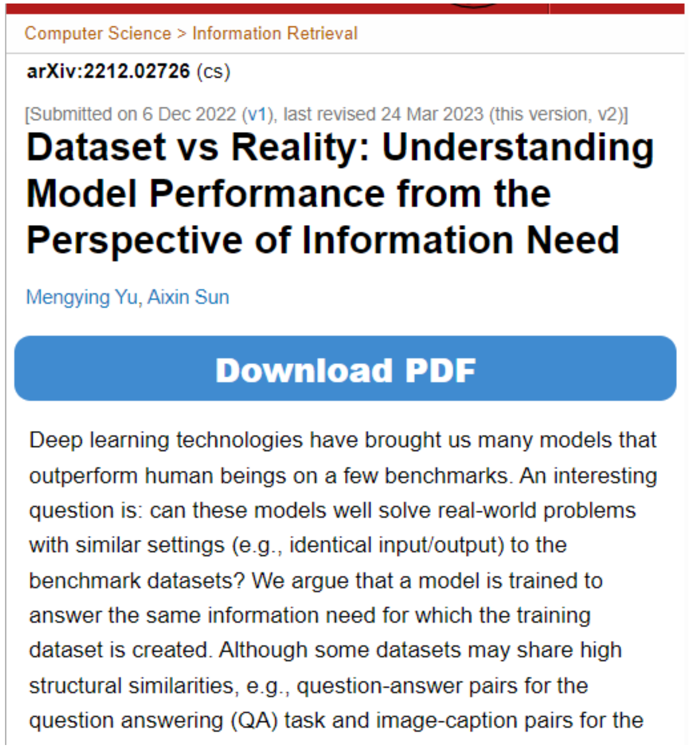
Yu-chen Fan, Yitong Ji, Jie Zhang, Aixin Sun

Download PDF

A typical benchmark dataset for recommender system (RecSys) evaluation consists of user-item interactions generated on a platform within a time period. The interaction generation mechanism partially explains why a user interacts with (e.g.,like, purchase, rate) an item, and the context of when a particular interaction happened. In this study, we conduct a meticulous analysis on the MovieLens dataset and explain the potential impact on using the dataset for evaluating recommendation algorithms. We make a few main findings from our analysis. First, there are significant

Dataset vs Reality




Computer Science > Information Retrieval

arXiv:2212.02726 (cs)

[Submitted on 6 Dec 2022 (v1), last revised 24 Mar 2023 (this version, v2)]

Dataset vs Reality: Understanding Model Performance from the Perspective of Information Need

Mengying Yu, Aixin Sun

Download PDF

Deep learning technologies have brought us many models that outperform human beings on a few benchmarks. An interesting question is: can these models well solve real-world problems with similar settings (e.g., identical input/output) to the benchmark datasets? We argue that a model is trained to answer the same information need for which the training dataset is created. Although some datasets may share high structural similarities, e.g., question-answer pairs for the question answering (QA) task and image-caption pairs for the

https://arxiv.org/abs/2212.02726

What is RecSys?

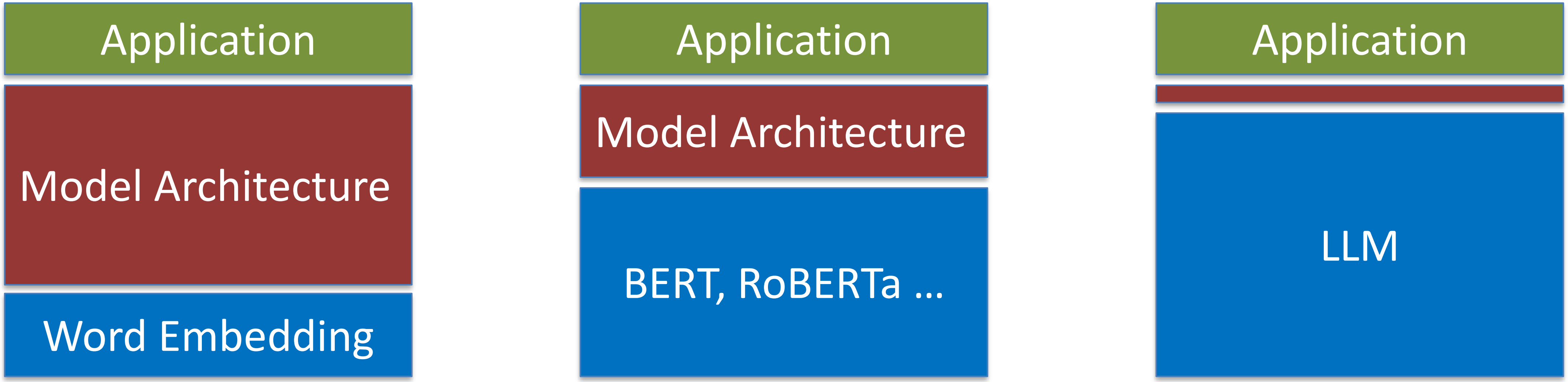
Dataset vs Reality?

Simulation of online process?

LLM?

Data	Offline dataset	Stream data
Evaluation	Train/test split	A/B testing
Metric	HitRate, NDCG…	CTR, CVR, GMV…
Model	A single model	Mixture of models?

LLM, Yet Another Solution to RecSys?



- How to present a scenario to LLM for a decision-making in a dynamic (online) setting?
- To what extent shall we trust the results on offline evaluation?

LLM, Yet Another Solution to RecSys?

- Disadvantages
  - Cannot consider business scenarios
  - Cannot access domain-specific user/item attributes
  - Unable to evaluate the business benefits brought by algorithms through offline evaluation

Academic Research

- Advantages
  - No need to consider implementation costs
  - No restrictions on the design of LLM-based recommenders
  - Potentially offer valuable insights for the industry

Acknowledgement

Ms. Ji Yitong

Mr. Fan Yu-chen

Dr. Zhang Jie

Dr. Li Chenliang

https://personal.ntu.edu.sg/axsun/

Computer Science > Information Retrieval

arXiv:2212.02726 (cs)

[Submitted on 6 Dec 2022 (v1), last revised 24 Mar 2023 (this version, v2)]

Dataset vs Reality: Understanding Model Performance from the Perspective of Information Need

Mengying Yu, Aixin Sun

Download PDF

Deep learning technologies have brought us many models that outperform human beings on a few benchmarks. An interesting question is: can these models well solve real-world problems with similar settings (e.g., identical input/output) to the benchmark datasets? We argue that a model is trained to answer the same information need for which the training dataset is created. Although some datasets may share high structural similarities, e.g., question-answer pairs for the

ACM DIGITAL LIBRARY

IR

RESEARCH-ARTICLE OPEN ACCESS

Take a Fresh Look at Recommender Systems from an Evaluation Standpoint

Author: Aixin Sun Authors Info & Claims

SIGIR '23: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval • July 2023 • Pages 2629–2638 • https://doi.org/10.1145/3539618.3591931

Published: 18 July 2023 Publication History Check for updates

ACM DIGITAL LIBRARY

ACM Transactions on Information Systems

RESEARCH-ARTICLE

A Critical Study on Data Leakage in Recommender System Offline Evaluation

Authors: Yitong Ji, Aixin Sun, Jie Zhang, Chenliang Li Authors Info & Claims

ACM Transactions on Information Systems, Volume 41, Issue 3 • Article No.: 75, pp 1–27 • https://doi.org/10.1145/3569930

Computer Science > Information Retrieval

arXiv:2307.09985 (cs)

[Submitted on 19 Jul 2023]

Our Model Achieves Excellent Performance on MovieLens: What Does it Mean?

Yu-chen Fan, Yitong Ji, Jie Zhang, Aixin Sun

Download PDF

A typical benchmark dataset for recommender system (RecSys) evaluation consists of user-item interactions generated on a platform within a time period. The interaction generation mechanism partially explains why a user interacts with (e.g.,like, purchase, rate) an item, and the context of when a particular interaction happened. In this study, we conduct a meticulous analysis on the MovieLens dataset and explain the potential impact on using the dataset for evaluating recommendation algorithms. We make a few main findings from our analysis. First, there are significant

感谢观看