A Critical Review of Recommender System

推荐系统研究现状的理解

Dr. Aixin Sun 孙爱欣
NTU Singapore

AI TIME AMiner AI 2000 学者专场

Revisiting RecSys: A 5-Year Journey

1

Computer Science > Information Retrieval

arXiv:2005.13829 (cs)

[Submitted on 28 May 2020 (v1), last revised 2 Jun 2020 (this version, v2)]

A Re-visit of the Popularity Baseline in Recommender Systems

Yitong Ji, Aixin Sun, Jie Zhang, Chenliang Li

View PDF

Popularity is often included in experimental evaluation to provide a reference performance for a recommendation ta popularity baseline is defined and evaluated, we top-tier conferences including KDD, WWW, SIG open source toolkits. We note that the widely ad simply ranks items based on the number of inte data. We argue that the current evaluation of po

2

Computer Science > Information Retrieva

arXiv:2010.11060 (cs)

[Submitted on 21 Oct 2020 (v1), last revised 30 Oct 2022 (this version, v4)]

A Critical Study on Data Leakage in Recommender System Offline Evaluation

Yitong Ji, Aixin Sun, Jie Zhang, Chenliang Li

View PDF

Recommender models are hard to evaluate, particularly under offline setting. In this paper, we provide a comprehensive and critical analysis of the data leakage issue in recommender system offline evaluation. Data leakage is caused by not observing global timeline in evaluating recommenders, e.g., train/test data split does not follow global timeline. As a result, a model learns from the user-item interactions that are not expected to be available at prediction time. We first show the temporal

3

Computer Science > Information Retriev

arXiv:2210.04149 (cs)

[Submitted on 9 Oct 2022 (v1), last revised 25 Apr 2023 (this version, v2)]

Take a Fresh Look at Recommender Systems from an Evaluation Standpoint

Aixin Sun

View PDF

Recommendation has become a prominent area of research in the field of Evaluation is also a tra by a few counter-intuit this perspectives pape m an evaluation stand all, hit rate, or NDCG, ey focus here is on ho g a recommender algo

4

Computer Science > Information Retrieva

arXiv:2307.09985 (cs)

[Submitted on 19 Jul 2023 (v1), last revised 24 Mar 2024 (this version, v3)]

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

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

View PDF

HTML (experimental)

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 of the MovieLens dataset and explain the

5

Computer Science > Information Retrieval

arXiv:2404.13375 (cs)

[Submitted on 20 Apr 2024]

Beyond Collaborative Filtering: A Relook at Task Formulation in Recommender Systems

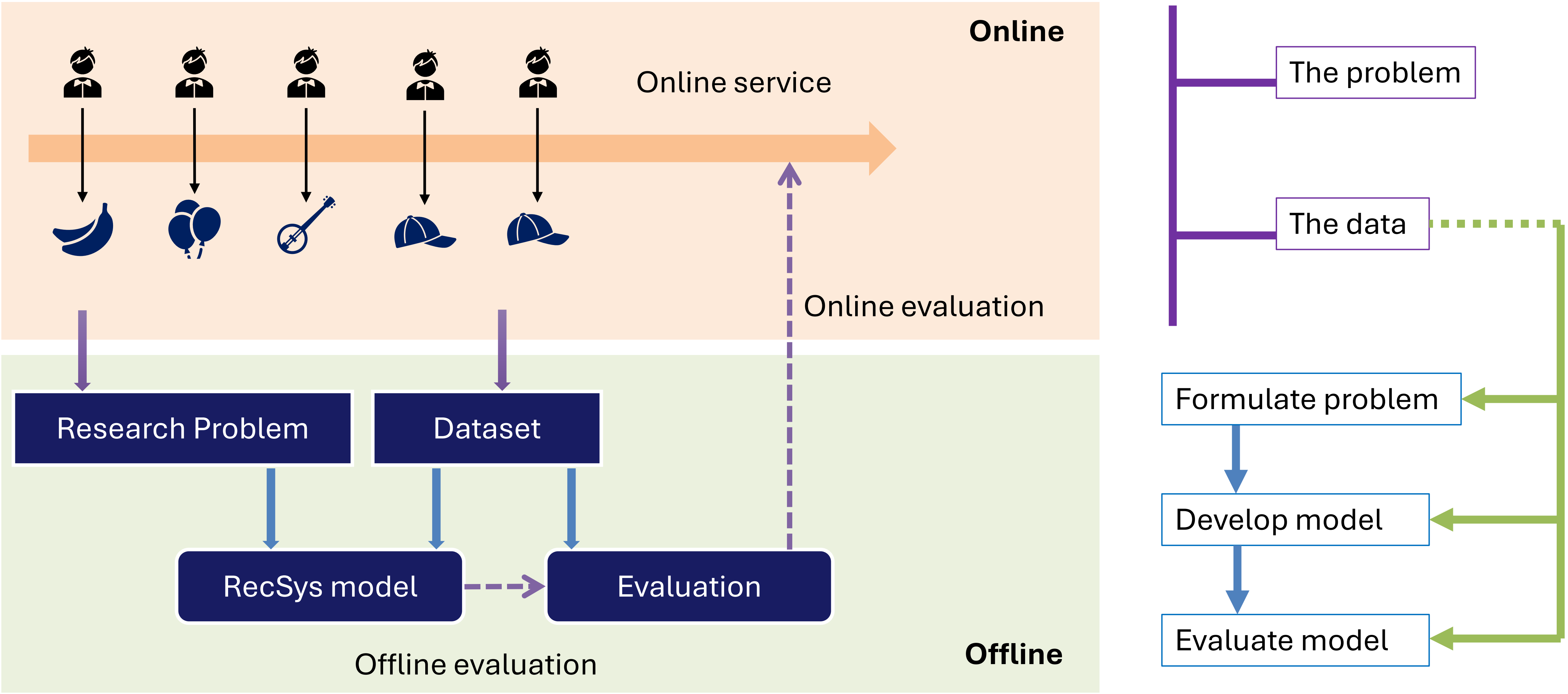
Aixin Sun

View PDF

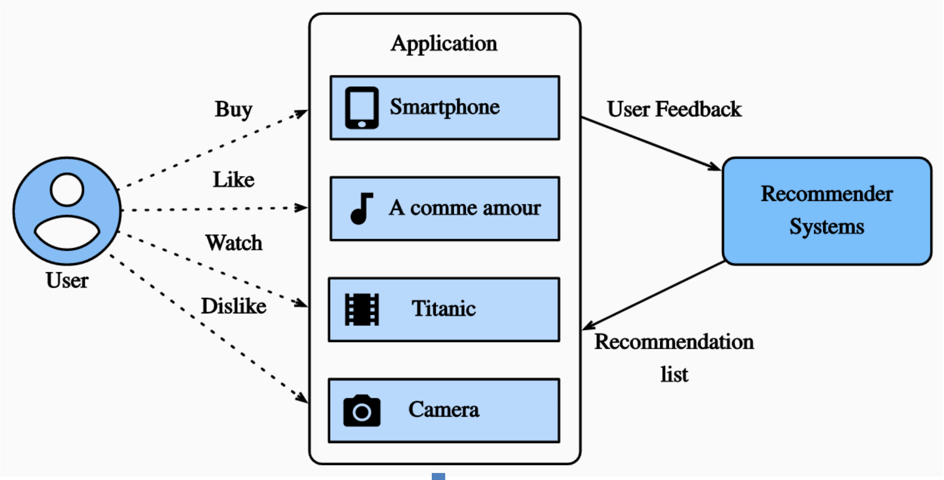
(experimental)

ecSys) have become indispensable in foundly influencing our everyday experiences. ficance, academic research in RecSys often research tasks from real-world contexts, formulation and more generalizable findings.

RecSys: The Online and the Offline



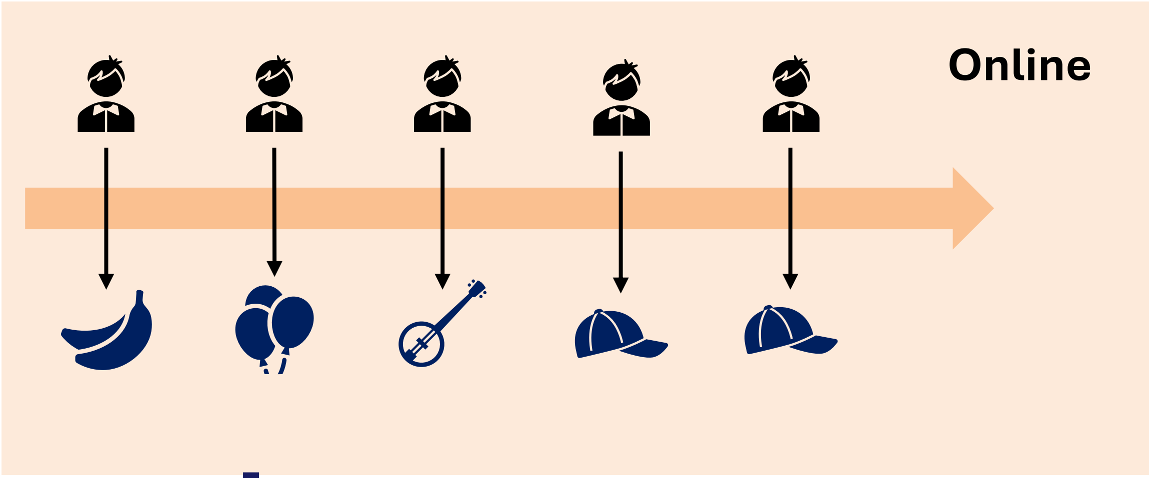
RecSys: The Problem Setting



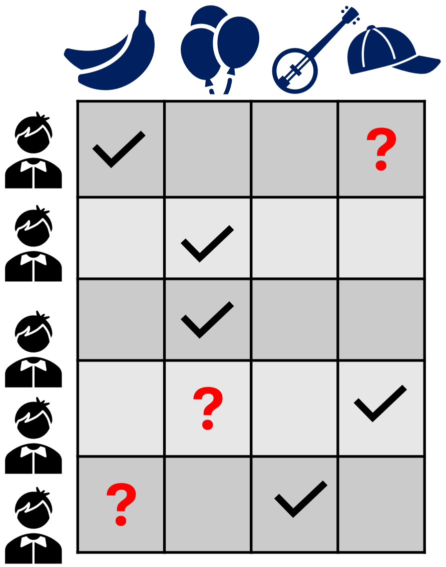
Task formulation

{User}
{Item}
{User – Item interaction}

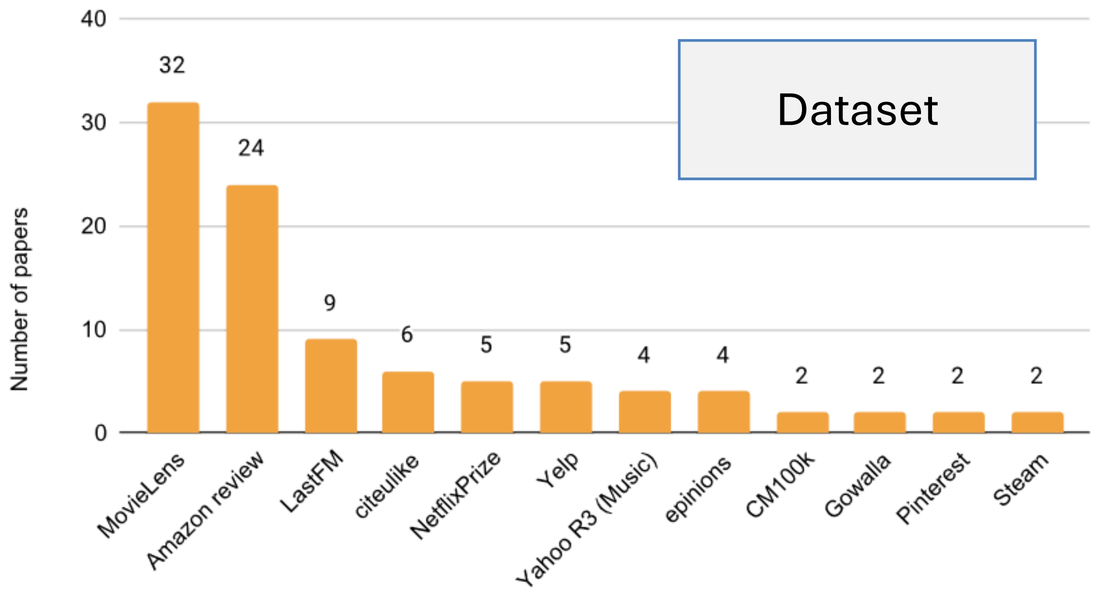
Image source: https://d2l.ai/chapter_recommender-systems/recsys-intro.html



Data projection



RecSys: The Current Status

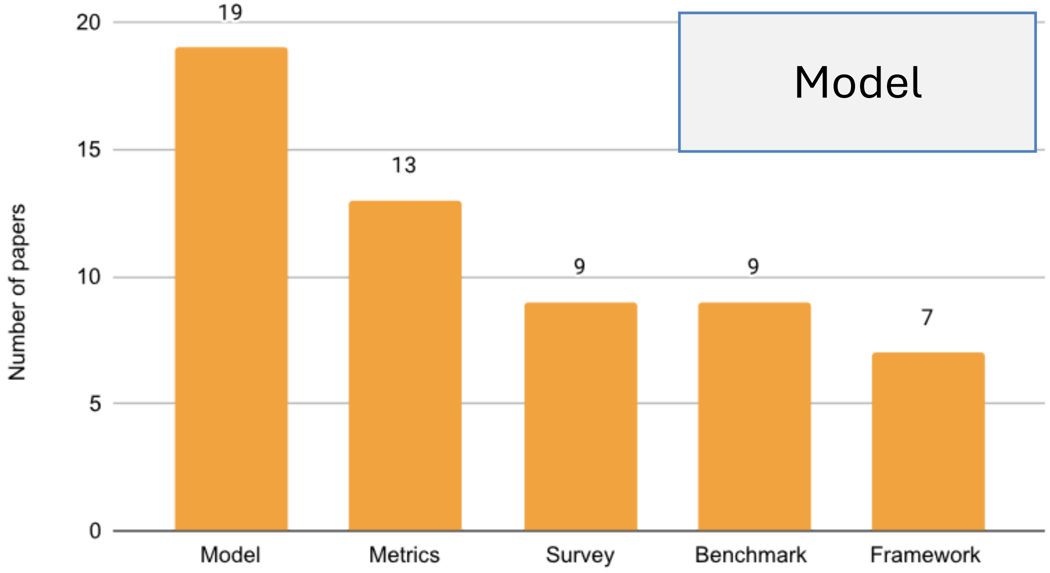


Exploring the Landscape of Recommender Systems Evaluation: Practices and Perspectives

CHRISTINE BAUER, Paris Lodron University Salzburg, Austria
EVA ZANGERLE, University of Innsbruck, Austria
ALAN SAID, University of Gothenburg, Sweden

TORS 2024

"the same few (and relatively old) datasets (i.e., **MovieLens**, Amazon review dataset) are used extensively"



RESEARCH-ARTICLE

Are we really making much progress? A worrying analysis of recent neural recommendation approaches

Authors: Maurizio Ferrari Dacrema, Paolo Cremonesi, Dietmar Jannach Authors Info & Claims

RecSys '19: Proceedings of the 13th ACM Conference on Recommender Systems • September 2019 • Pages 101–109 • https://doi.org/10.1145/3298689.3347058

RecSys: Evaluation

RecBaselines2023: a new dataset for choosing baselines for recommender models

Veronika Ivanova
veronika.ivanova88@yandex.ru
National Research University Higher School of Economics
Moscow, Russian Federation

Oleg Lashinin
o.a.lashinin@tinkoff.ru
Tinkoff
Moscow, Russian Federation

Marina Ananyeva
m.ananyeva@tinkoff.ru
National Research University Higher School of Economics
Moscow, Russian Federation

Sergey Kolesnikov
scitator@gmail.com
Tinkoff
Moscow, Russian Federation

ABSTRACT

The number of proposed recommender algorithms continues to grow. The authors propose new approaches and compare them with existing models, called baselines. Due to the large number of recommender models, it is difficult to estimate which algorithms to choose in the article. To solve this problem, we have collected

However, there are no rigid guidelines that defin
list of essential baselines. Inaccurate selection of
to incorrect conclusions about the performanc
model. Subsequent papers [9] on the reproducib
of existing work have demonstrated this fact. Fo
recent papers [22, 45], the authors report that f

Lack of baselines?

Even if baselines are compared

SHORT-PAPER OPEN ACCESS

Everyone's a Winner! On Hyperparameter Tuning of Recommendation Models

Authors: Faisal Shehzad, Dietmar Jannach Authors Info & Claims

RecSys '23: Proceedings of the 17th ACM Conference on Recommender Systems • September 2023 • Pages 652–657 • https://doi.org/10.1145/3604915.3609488

Published: 14 September 2023 Publication History

Check for updates

Shall we reference large-scale evaluations?

Large-scale Evaluations

with the full ranking of the models. Ferrari Dacrema et al. [41] and its extended version Ferrari Dacrema et al. [40] perform a reproducibility study, critically analyzing the performance of 12 neural recommendation approaches in comparison to well-tuned, established, non-neural baseline methods. Their work identifies several methodological issues and finds that 11 of the 12 analyzed approaches are outperformed by far simpler, yet well-tuned, methods (e.g., nearest-neighbor or content-based approaches). In a similar vein, Latifi and Jannach [61] perform a reproducibility study where they benchmark Graph Neural Networks (GNN) against an effective session-based nearest neighbor method. Also, this work finds that the conceptually simpler method outperforms the GNN-based method. Anelli et al. [9] perform a reproducibility study, systematically comparing 10 collaborative filtering algorithms (including approaches based on nearest-neighbors, matrix factorization, linear models, and techniques based on deep learning). Different to Ferrari Dacrema et al. [40, 41], Anelli et al. [9] benchmark all algorithms using the very same datasets (MovieLens-1M [48], Amazon Digital Music [74], and epinions [92]) and the identical evaluation protocol. Based on their study on modest-sized datasets, they conclude—similarly to other works—that the latest models are often not the best-performing ones. Kouki et al. [59] compare 14 models (8 baseline and 6 deep learning) for session-based recommendations using 8 different popular evaluation metrics. After an offline evaluation, they selected the 5 algorithms that performed the best and ran a second round of evaluation using human experts (user study). Reference [90] provides benchmarks across several datasets, recommendation approaches, and metrics; beyond that, this work introduces the toolkit daisyRec. Zhu et al. [99] compare 24 models for click-through rate (CTR) prediction on multiple dataset settings. Their evaluation framework for CTR (including the benchmarking tools, evaluation protocols, and experimental settings) is publicly available. Latifi et al. [62] focus on sequential recommendation problems, for which they compare the Transformer-based BERT4Rec method [89] to nearest-neighbor methods, showing that the nearest-neighbor methods achieve comparable performance to BERT4Rec for the smaller datasets, whereas BERT4Rec outperforms the simple methods when the datasets are larger.

Exploring the Landscape of Recommender Systems Evaluation: Practices and Perspectives

CHRISTINE BAUER, Paris Lodron University Salzburg, Austria
EVA ZANGERLE, University of Innsbruck, Austria
ALAN SAID, University of Gothenburg, Sweden

TORS 2024

Recommender systems research and practice are fast-developing topics with growing adoption in a wide variety of information access scenarios. In this article, we present an overview of research specifically focused on the evaluation of recommender systems. We perform a systematic literature review, in which we analyze 57 papers spanning six years (2017–2022). Focusing on the processes surrounding evaluation, we dial in on the methods applied, the datasets utilized, and the metrics used. Our study shows that the predominant experiment type in research on the evaluation of recommender systems is offline experimentation and that online evaluations are primarily used in combination with other experimentation methods, e.g., an offline experiment. Furthermore, we find that only a few datasets (MovieLens, Amazon review dataset) are widely used, while many datasets are used in only a few papers each. We observe a similar scenario when analyzing the employed performance metrics—a few metrics are widely used (precision, normalized Discounted Cumulative Gain, and Recall), while many others are used in only a few papers. Overall, our review indicates that beyond-accuracy qualities are rarely assessed. Our analysis shows that the research community working on evaluation has focused on the development of evaluation in a rather narrow scope, with the majority of experiments focusing on a few metrics, datasets, and methods.

RESEARCH-ARTICLE

Are we really making much progress? A worrying analysis of recent neural recommendation approaches

Authors: Maurizio Ferrari Dacrema, Paolo Cremonesi, Dietmar Jannach Authors Info & Claims

RecSys '19: Proceedings of the 13th ACM Conference on Recommender Systems • September 2019 • Pages 101–109 • https://doi.org/10.1145/3298689.3347058

On the Generalizability and Predictability of Recommender Systems

NeurIPS 2022

Duncan McElfresh[*1], Sujay Khandagale[*1], Jonathan Valverde[*1,3],
John P. Dickerson[2,3], Colin White[1]
[1]Abacus.AI, [2]ArthurAI, [3]University of Maryland

In this work, we show that the best algorithm and hyperparameters are highly dependent on the dataset and user-defined performance metric. Specifically, we run the first large-scale study of rec-sys approaches by comparing 24 algorithms across 85 datasets and 315 metrics. For each dataset and algorithm pair, we test up to 100 hyperparameters (given a 10 hour time limit per pair). The codebase that we release, which includes a unified API for a large, diverse set of algorithms, datasets, and metrics, may be of independent interest. We show that the algorithms do not *generalize* – the set of algorithms which perform well changes substantially across dataset and across performance metrics. Furthermore, the best hyperparameters of a rec-sys algorithm on one dataset often perform significantly worse than the best hyperparameters on a different dataset. Although we show that there are no universal algorithms that work well on most datasets, we *do* show that various meta-features of the dataset can be used to *predict* the performance of rec-sys algorithms. In fact, the same meta-features are also predictive of the runtime of rec-sys algorithms as well as the "dataset hardness" – how challenging it is to find a high-performing model on a particular dataset.

Table 1: The relative performance of each rec-sys algorithm depends on the dataset and metric. This table shows the mean, min (best) and max (worst) rank achieved by all 20 algorithms over all 85 datasets, over 10 accuracy and hit-rate metrics at all cutoffs tested. This includes metrics NDCG, precision, recall, Prec.-Rec.-Min-density, hit-rate, F1, MAP, MAP-Min-density, ARHR, and MRR.

Rank	Item-KNN	P3alpha	SLIM-BPR	EASE-R	RP3beta	SVD	SLIM-ElasticNet	iALS	NMF	User-KNN	MF-Funk	TopPop	MF-Asy	MF-BPR	Mult-VAE	U-neural	GlobalEffects	CoClustering	Random	SlopeOne
Min.	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	9	7
Max.	14	18	14	18	17	16	17	19	14	17	18	19	16	17	20	20	20	19	20	20
Mean	2.3	4.2	4.7	5.3	6	6	7	7	7.1	7.6	9.4	10.4	10.7	11.2	11.7	12.3	13.3	14.9	16.2	16.7

RecSys: The Current Status, but Why?

Dataset

MovieLens has been used by ≈70% of RecSys papers. **Is MovieLens a representative dataset?**

Model

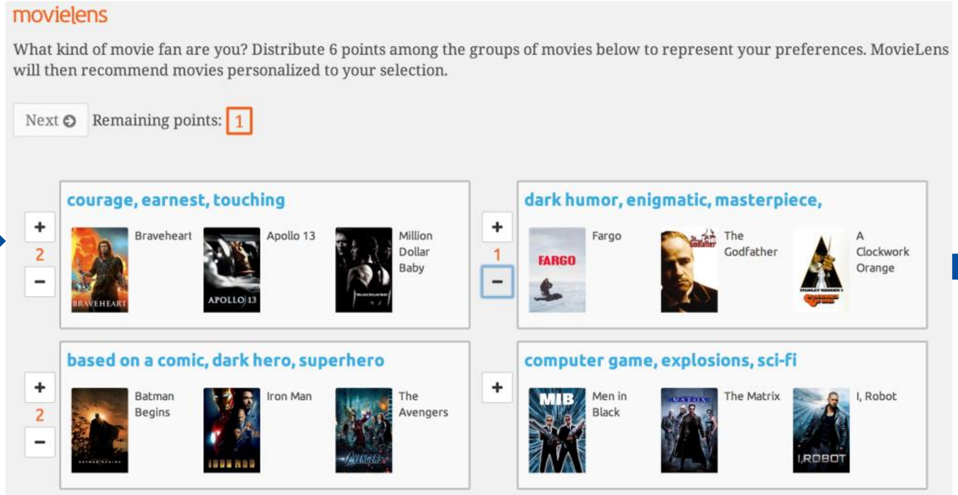
There are so many models available. Is there **a shared understanding** on which models shall be used as baselines?

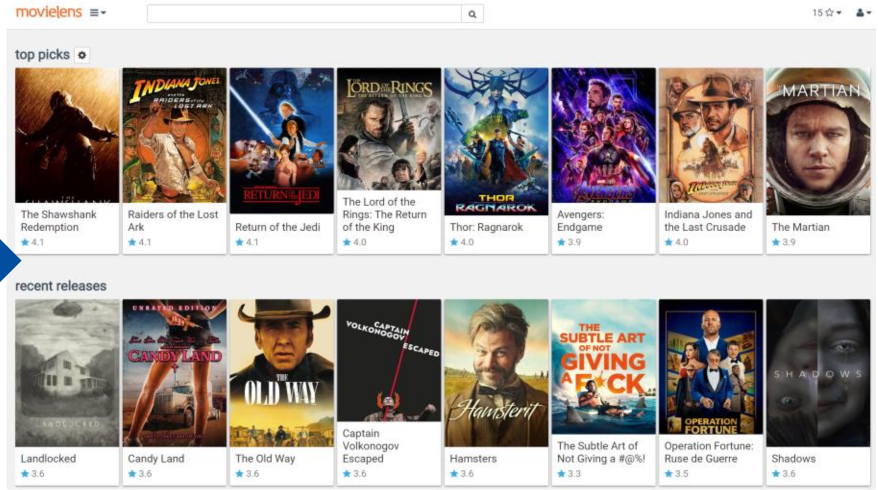
Evaluation

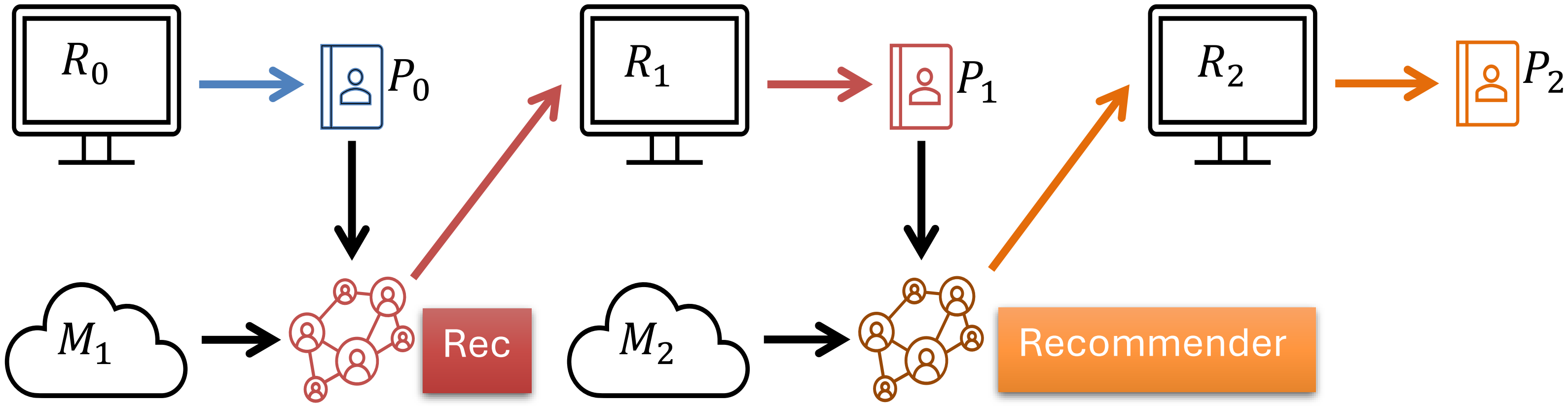
Why **item-KNN** remains a strong performer?

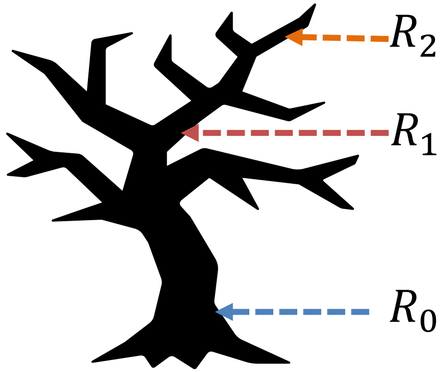
Shall We Re-look at the Dataset?

The MovieLens dataset



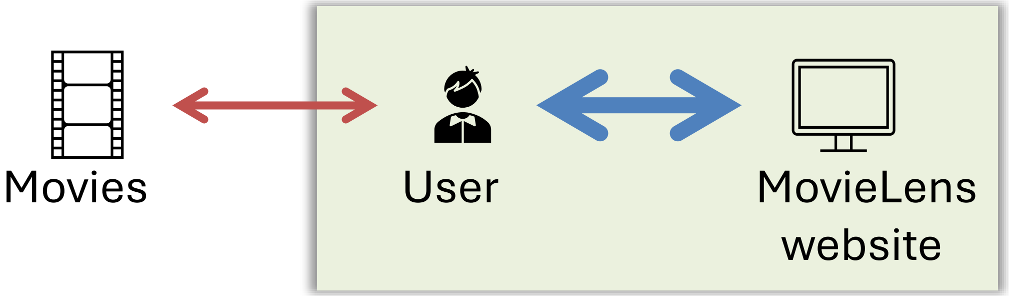




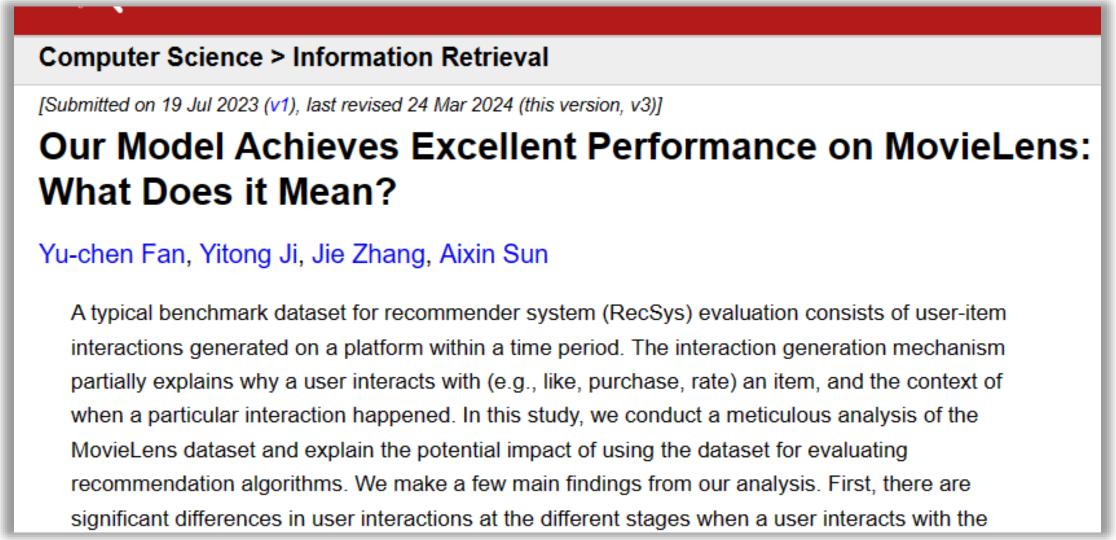


MovieLens: One of the Two Kinds of Interactions

- **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
  - More than half users complete all ratings in **ONE day**
  - Cold-start dataset for “static preference”



Computer Science > Information Retrieval

[Submitted on 19 Jul 2023 (v1), last revised 24 Mar 2024 (this version, v3)]

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

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

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 of the MovieLens dataset and explain the potential impact of using the dataset for evaluating recommendation algorithms. We make a few main findings from our analysis. First, there are significant differences in user interactions at the different stages when a user interacts with the

RecSys: The Current Status, but Why?

The problem

The data

Formulate problem

Develop model

Evaluate model

Data defines task

Model

Is there a shared understanding on which models shall be used as baselines?

Evaluation

Why item-KNN remains a strong performer?

Training data → RecSys model → Test data

Data defines the task

88 papers in RecSys conferences (2020 – 2022)

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

Take a Fresh Look at Recommender Systems from an Evaluation Standpoint

Aixin Sun
School of Computer Science and Engineering
Nanyang Technological University
Singapore
axsun@ntu.edu.sg

ABSTRACT

Recommendation has become a prominent area of research in the field of Information Retrieval (IR). Evaluation is also a traditional research topic in this community. Motivated by a few counter-intuitive observations reported in recent studies, this perspectives paper takes a fresh look at recommender systems from an evaluation standpoint. Rather than examining metrics like recall, hit rate, or NDCG, or perspectives like novelty and diversity, the key focus here is on *how these metrics are calculated when evaluating a recommender algorithm*. Specifically, the commonly used train/test data splits and their consequences are re-examined. We begin by examining common data splitting methods, such as random split or leave-one-out, and discuss why the popularity baseline is poorly defined under such splits. We then move on to explore the two implications of neglecting a global timeline during evaluation: *data leakage* and *oversimplification of user preference modeling*. Afterwards, we present new perspectives on recommender systems, including techniques for evaluating algorithm performance that more

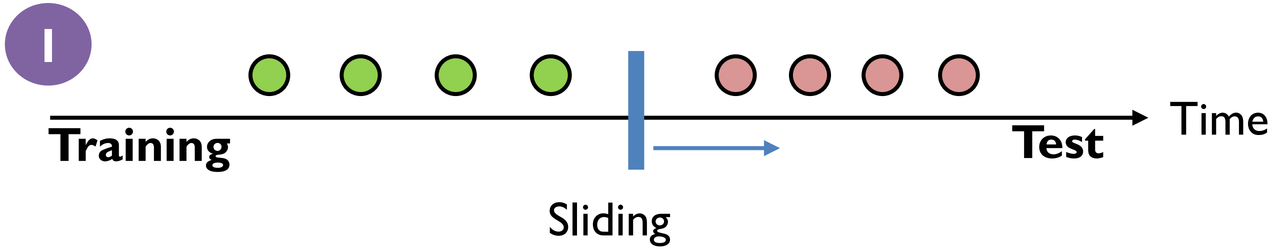
This is a strong indication of research interests on Recommender Systems (RecSys) in the Information Retrieval (IR) community. As evaluation is also a traditional research topic in IR, it is interesting to study how recommendation algorithms are evaluated in general. More interestingly, a few recent papers report counter-intuitive observations made from experiments on recommender system, both in offline and online settings [18, 26, 37, 38, 40].

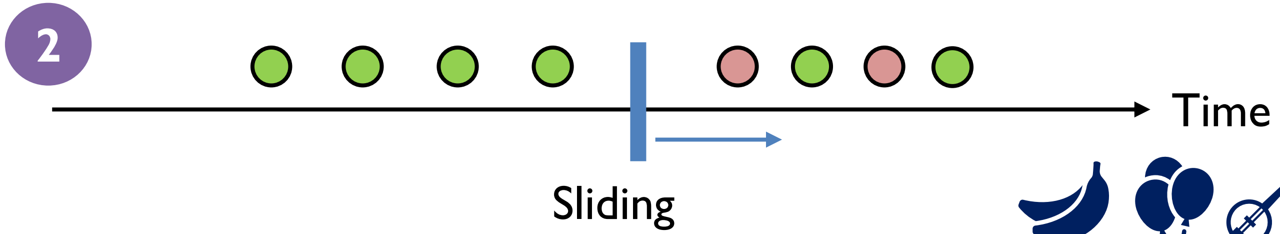
Here are some example counter-intuitive observations. Ji et al. [18] report that both users who spend more time and users who have many interactions with a recommendation system receive poorer recommendations, compared to users who spend less time or who have relatively fewer interactions with the system. This observation holds o… l-
els (*i.e.*, BPR [33], N…]
and TiSASRec [25]) c… 1,
Yelp, Amazon-music… t
footwear vendor, through online experiments, Sysko-Romanczuk et al. [37] observe that "experience with the vendor showed a nega-

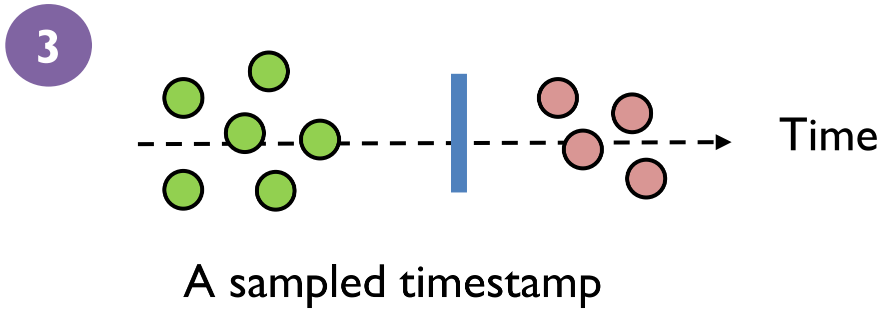
SIGIR 2023

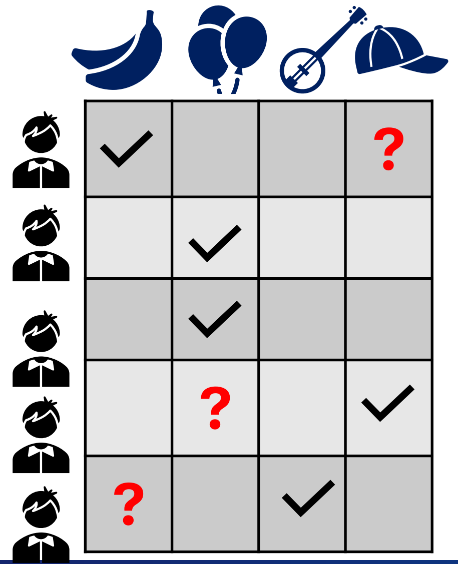
Training data → RecSys model → Test data

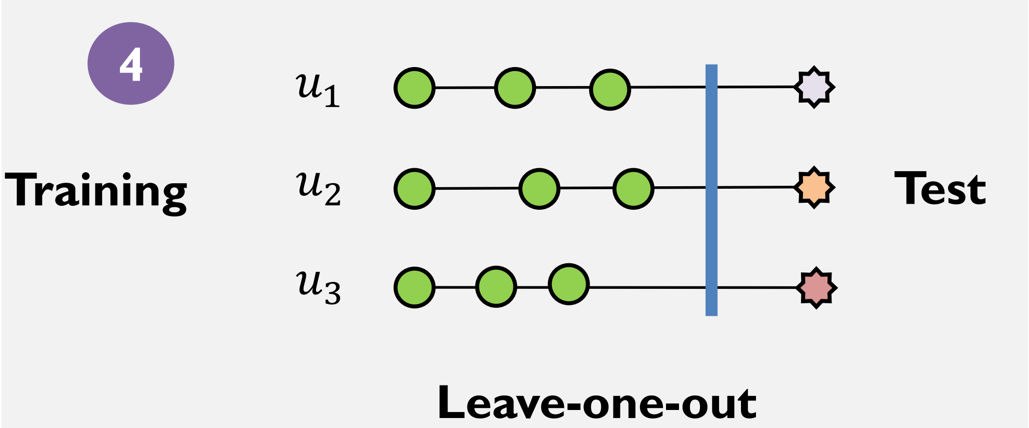
Per.	Train/test split
34%	Random split
25%	Leave-one-out

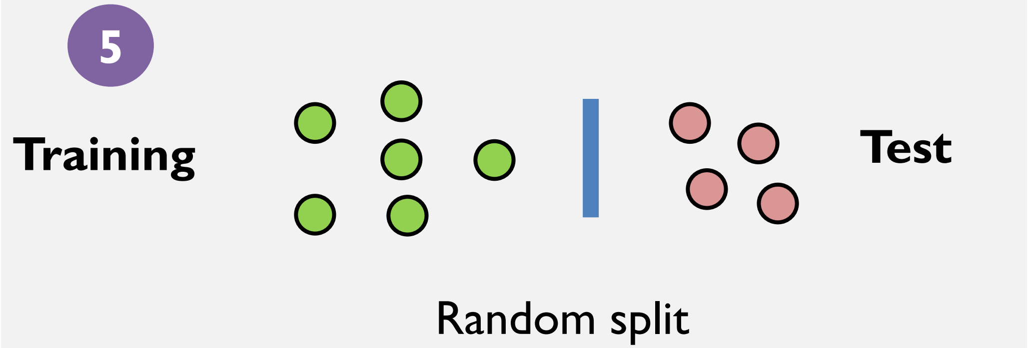






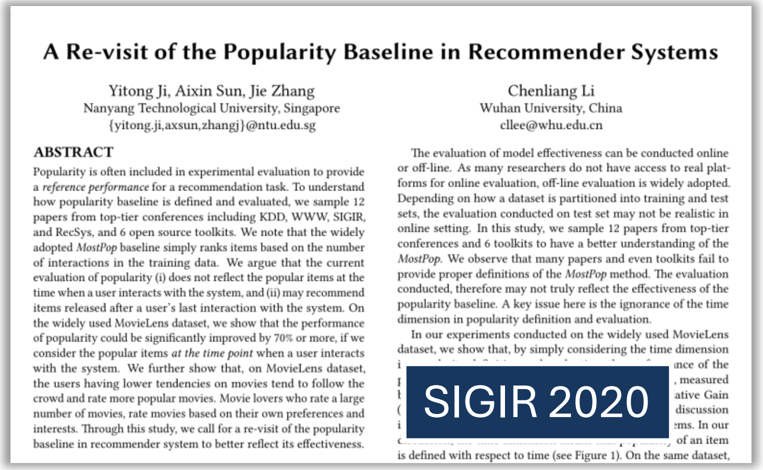






Popularity in RecSys Research: Defined by the Training Set

- Partition the data into train and test
- **Item popularity**: number of interactions in training set
- Popularity **following time?**
  - At time t_{x1} for user u_1
  - At time t_{x2} for user u_2
  - At time t_{x3} for user u_3



A Re-visit of the Popularity Baseline in Recommender Systems

Yitong Ji, Aixin Sun, Jie Zhang
Nanyang Technological University, Singapore
{yitong.ji,axsun,zhangj}@ntu.edu.sg

Chenliang Li
Wuhan University, China
cllee@whu.edu.cn

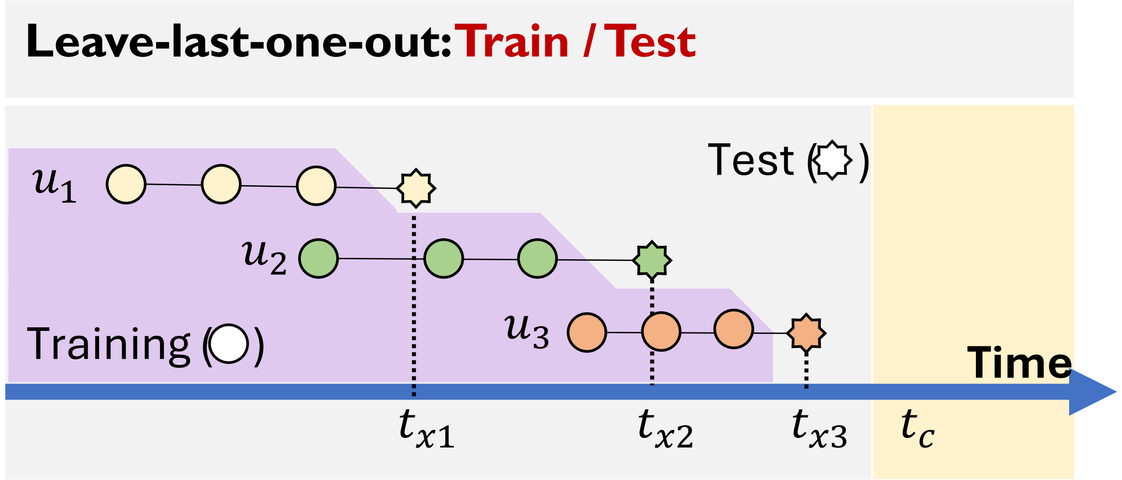
ABSTRACT

Popularity is often included in experimental evaluation to provide a *reference performance* for a recommendation task. To understand how popularity baseline is defined and evaluated, we sample 12 papers from top-tier conferences including KDD, WWW, SIGIR, and RecSys, and 6 open source toolkits. We note that the widely adopted *MostPop* baseline simply ranks items based on the number of interactions in the training data. We argue that the current evaluation of popularity (i) does not reflect the popular items at the time when a user interacts with the system, and (ii) may recommend items released after a user's last interaction with the system. On the widely used MovieLens dataset, we show that the performance of popularity could be significantly improved by 70% or more, if we consider the popular items *at the time point* when a user interacts with the system. We further show that, on MovieLens dataset, the users having lower tendencies on movies tend to follow the crowd and rate more popular movies. Movie lovers who rate a large number of movies, rate movies based on their own preferences and interests. Through this study, we call for a re-visit of the popularity baseline in recommender system to better reflect its effectiveness.

The evaluation of model effectiveness can be conducted online or off-line. As many researchers do not have access to real platforms for online evaluation, off-line evaluation is widely adopted. Depending on how a dataset is partitioned into training and test sets, the evaluation conducted on test set may not be realistic in online setting. In this study, we sample 12 papers from top-tier conferences and 6 toolkits to have a better understanding of the *MostPop*. We observe that many papers and even toolkits fail to provide proper definitions of the *MostPop* method. The evaluation conducted, therefore may not truly reflect the effectiveness of the popularity baseline. A key issue here is the ignorance of the time dimension in popularity definition and evaluation.

In our experiments conducted on the widely used MovieLens dataset, we show that, by simply considering the time dimension … ance of the … , measured … ative Gain … discussion … ems. In our … of an item is defined with respect to time (see Figure 1). On the same dataset,

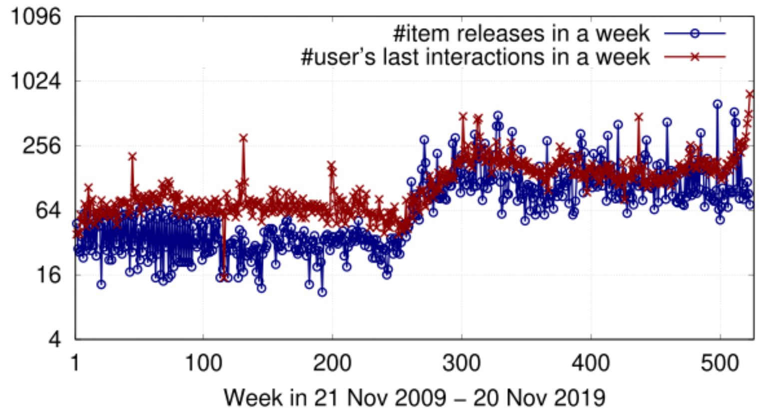
SIGIR 2020



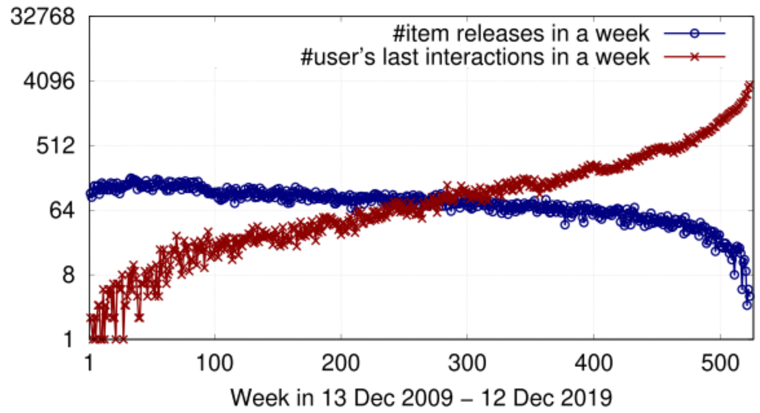
Is "Popularity" method meaningful?

Can be Observed on Datasets?

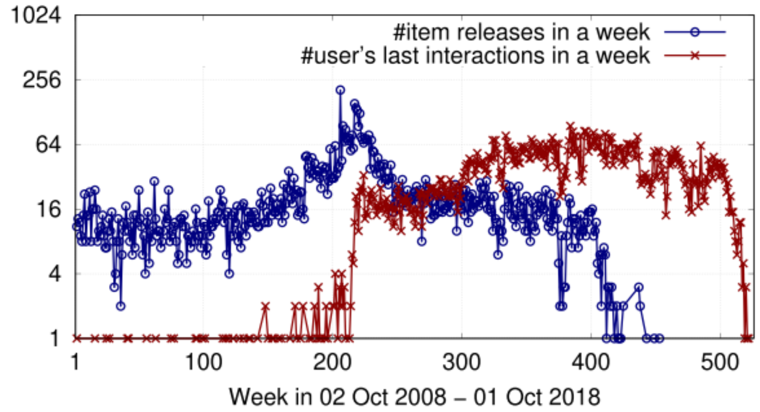
- **Blue** vs **Brown** points
  - No. of items new to each week
  - No. of users' last interaction
- Popularity seems not reasonable.
- **How about other models?**



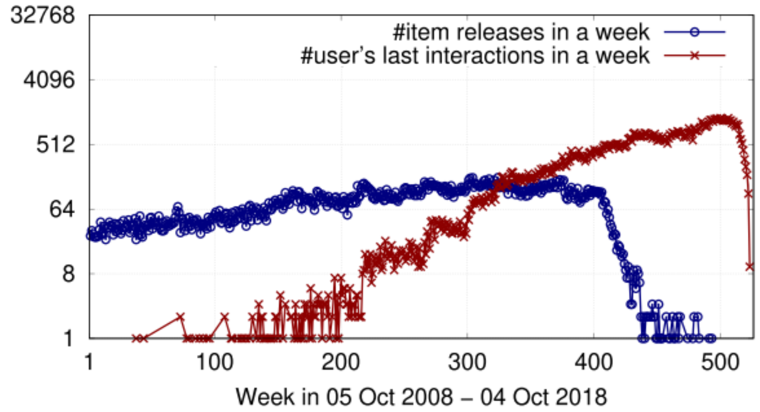
(a) MovieLens-25M



(b) Yelp



(c) Amazon-music



(d) Amazon-electronic

Data Leakage in RecSys

- A model is trained with **future data** with respect to the timepoint of test instance
  - At time t_{x1} for user u_1
  - At time t_{x2} for user u_2
  - At time t_{x3} for user u_3

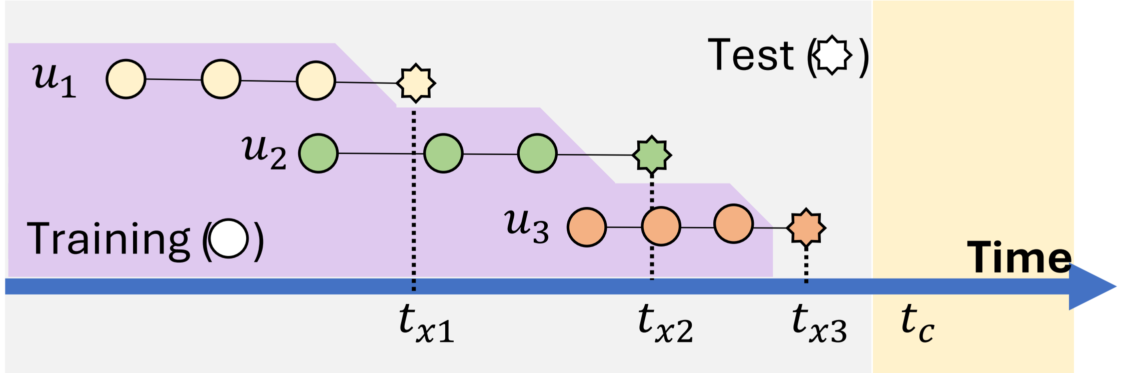
- Can we prove this?
- What are the impacts to our results?

A Critical Study on Data Leakage in Recommender System Offline Evaluation

YITONG JI, Nanyang Technological University, Singapore
AIXIN SUN, Nanyang Technological University, Singapore
JIE ZHANG, Nanyang Technological University, Singapore
CHENLIANG LI, Wuhan University, China

ACM TOIS 2023

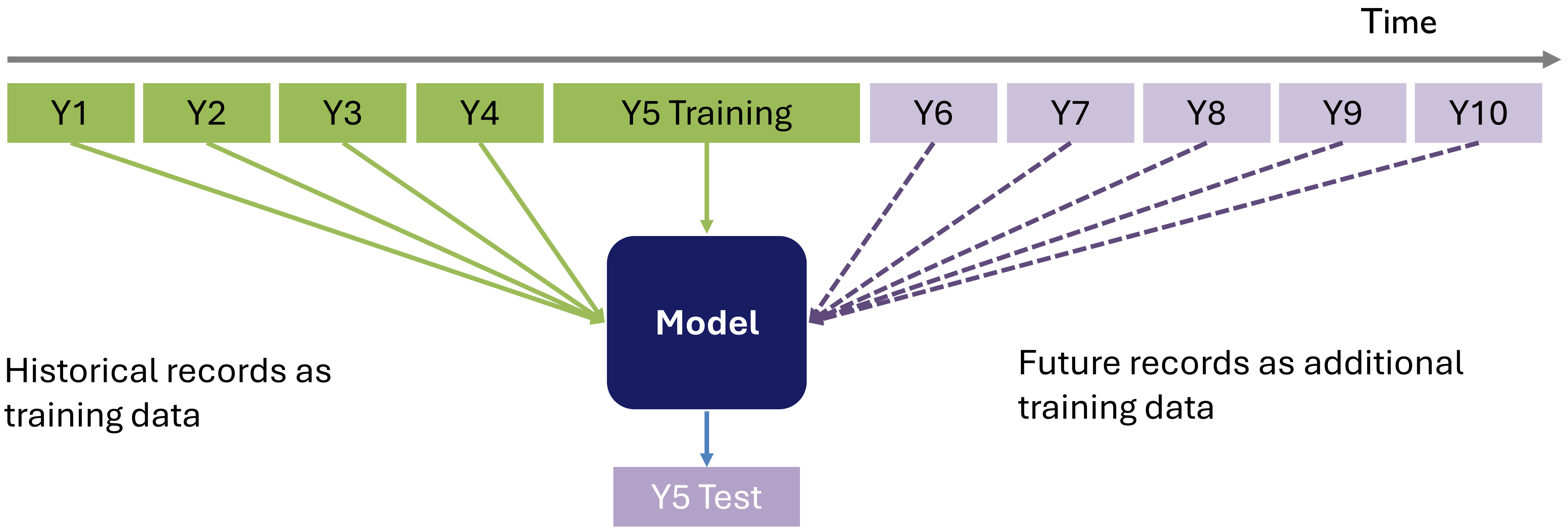
Recommender models are hard to evaluate, particularly under offline setting. In this paper, we provide a comprehensive and critical analysis of the data leakage issue in recommender system offline evaluation. Data leakage is caused by not observing global timeline in evaluating recommenders *e.g.*, train/test data split does not follow global timeline. As a result, a model learns from the user-item interactions that are not expected to be available at prediction time. We first show the temporal dynamics of user-item interactions along global timeline, then explain why data leakage exists for collaborative filtering models. Through carefully designed experiments, we show that all models indeed recommend future items that are not available at the time point of a test instance, as the result of data leakage. The experiments are conducted with four widely used baseline models - BPR, NeuMF, SASRec, and LightGCN, on four popular offline datasets - MovieLens-25M, Yelp, Amazon-music, and Amazon-electronic, adopting leave-last-one-out data split.[1] We further show that data leakage does impact models' recommendation accuracy. Their relative performance orders thus become unpredictable with different amount of leaked future data in training. To evaluate recommendation systems in a realistic manner in offline setting, we propose a timeline scheme, which calls for a revisit of the recommendation model design.



Applicable to all ML/DL- based models

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

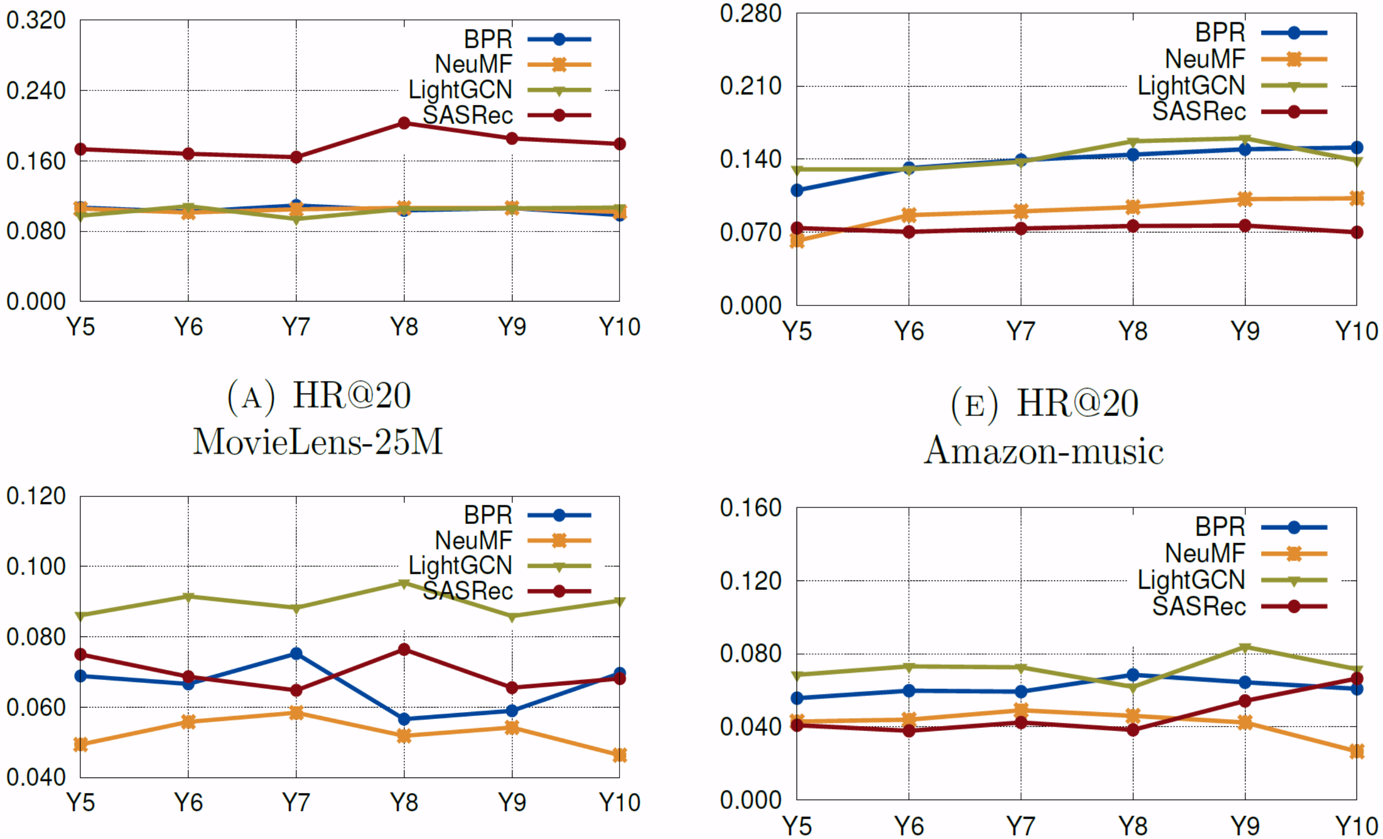
- **Future items**: the items are exclusively available only after the specific time point of a given test instance.

- All models recommend "future items" → **invalid recommendation**

Model	Dataset	MovieLens-25M		Yelp		Amazon-music		Amazon-electronic	
	Test year	Y5	Y7	Y5	Y7	Y5	Y7	Y5	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 RecSys Accuracy

- **Strictly speaking:**
  - The impact on recommendation accuracy is **not predictable.**
  - The **relative performance ordering** of the evaluated models does not exhibit consistent patterns.
- **Less strictly?**
  - The *relative* performance ordering *largely* remains
  - **Is there a reason behind?**



(A) HR@20
MovieLens-25M

(E) HR@20
Amazon-music

(C) HR@20
Yelp

(G) HR@20
Amazon-electronic

RecSys: The Current Status, but Why?

The problem

The data

Formulate problem

Develop model

Evaluate model

Research Problem Simplification

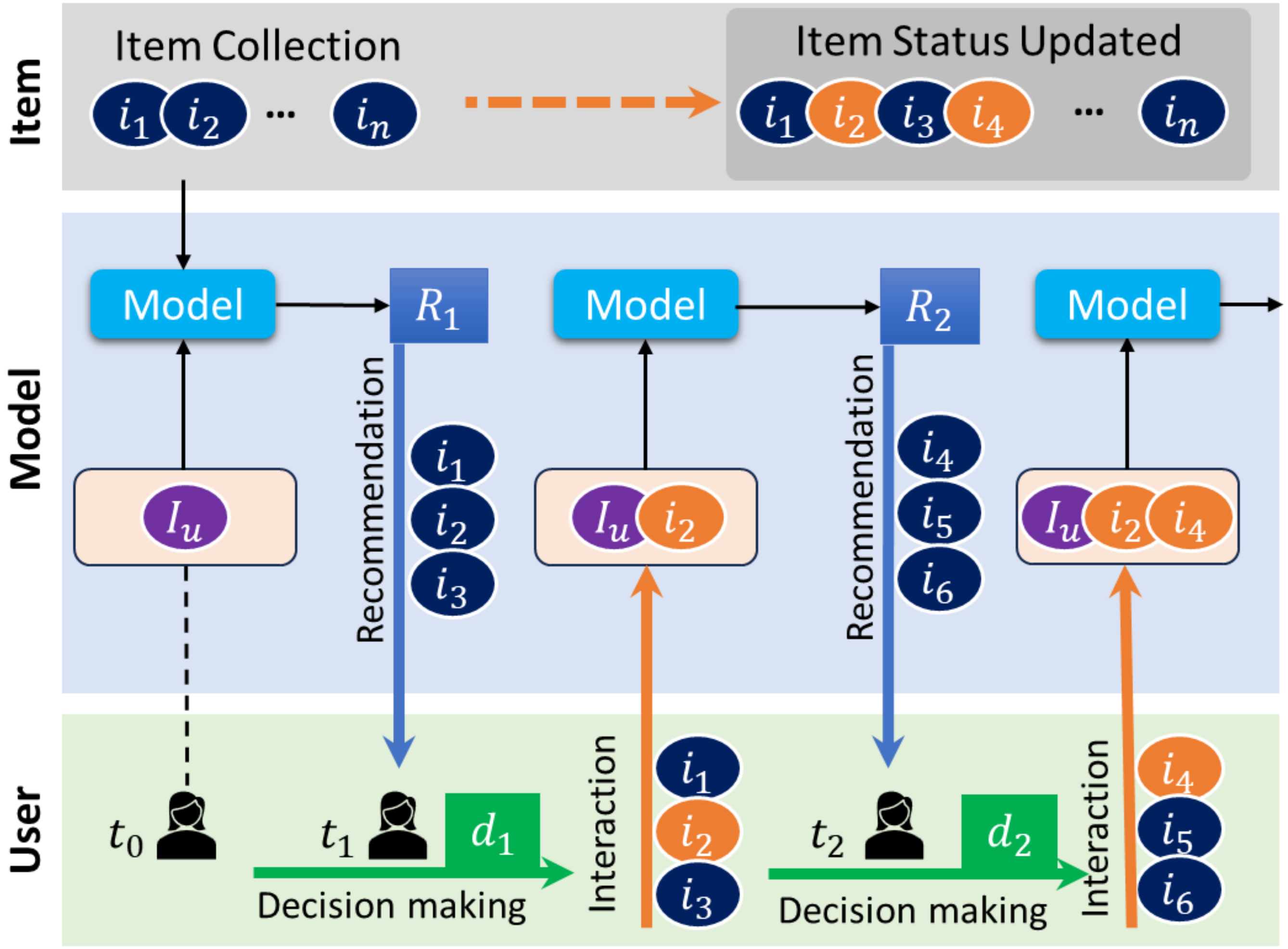
Model

Is there a shared understanding on which models shall be used as baselines?

Evaluation

Why item-KNN remains a strong performer?

“Training → RecSys model → Test” Reflect RecSys?

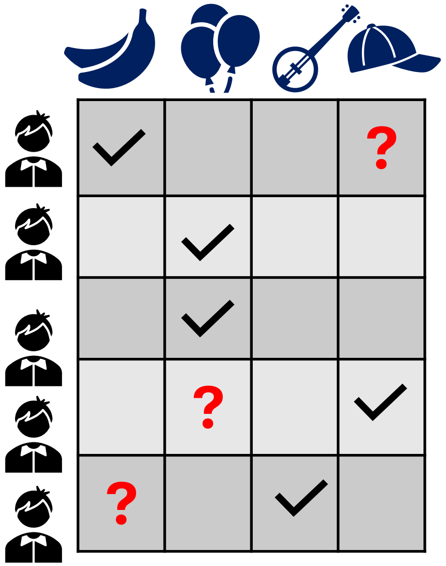


- RecSys aims to make **recommendations** for a **decision-making** process
- The decision-making is **dynamic** with two types of preferences
  - General preference
  - Current contextual factors

    →**item-kNN**

Current Context is Task-Specific and Dynamic

- **The abstraction: {User} {Item} {User-item}**
  → loss of the context
  - Movie recommendation?
  - E-commerce recommendation?
  - Hotel, POI recommendation?



- **Example: Food delivery recommendation** mobile apps
  - User input: User ID, delivery address
  - Task-specific factors:
    - Breakfast, lunch, dinner?
    - Repeat vs Exploration? → Significant different in item search space
    - Current context, user mood (make a good guess)

The Understanding of Current Practice

Dataset

An offline dataset usually **does not** capture dynamic changing context factors

Model

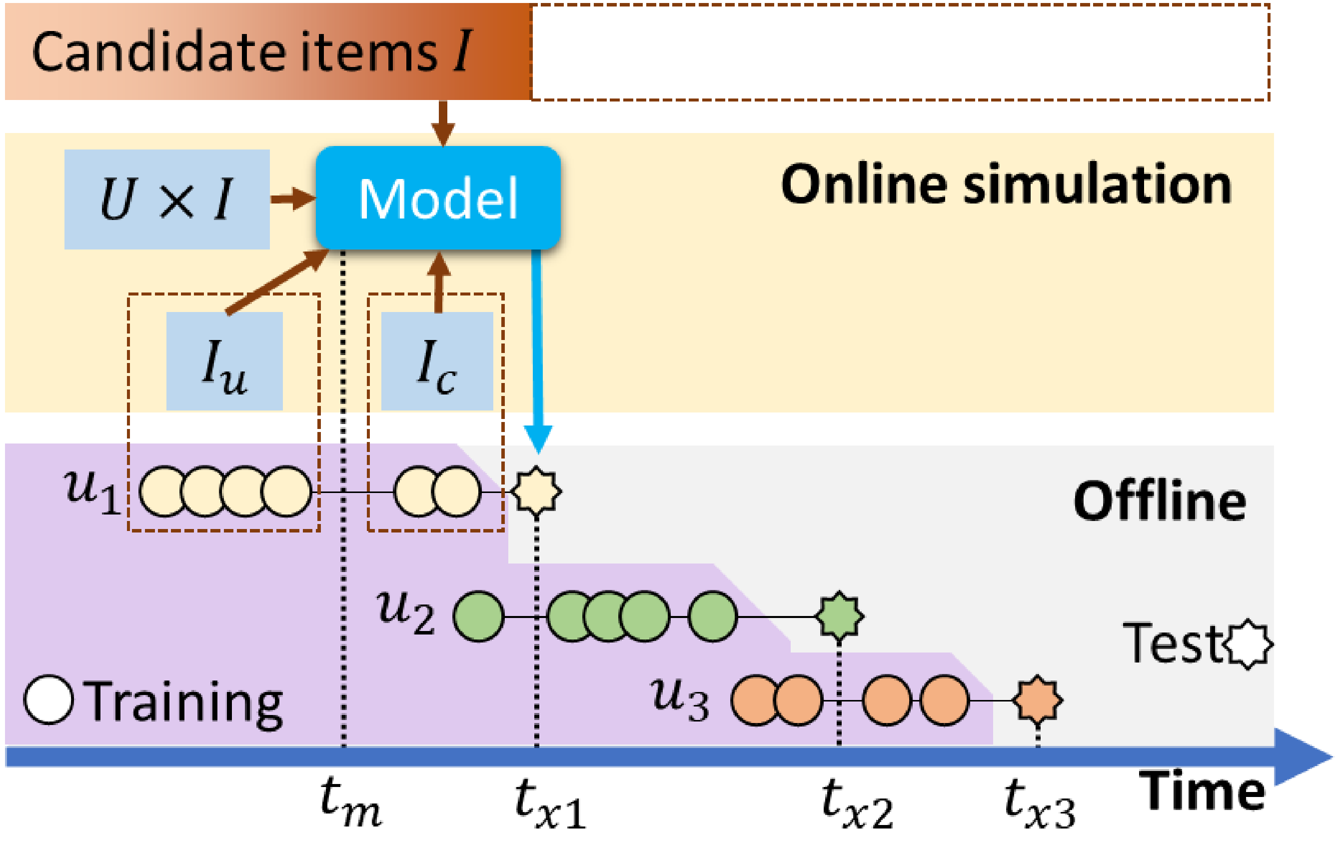
The model is trained based on **decision outcomes**, not the decision making

Hence only user **general preference** is learned over time

Evaluation

The evaluation is on the ability of RecSys models in capturing user **general preference**

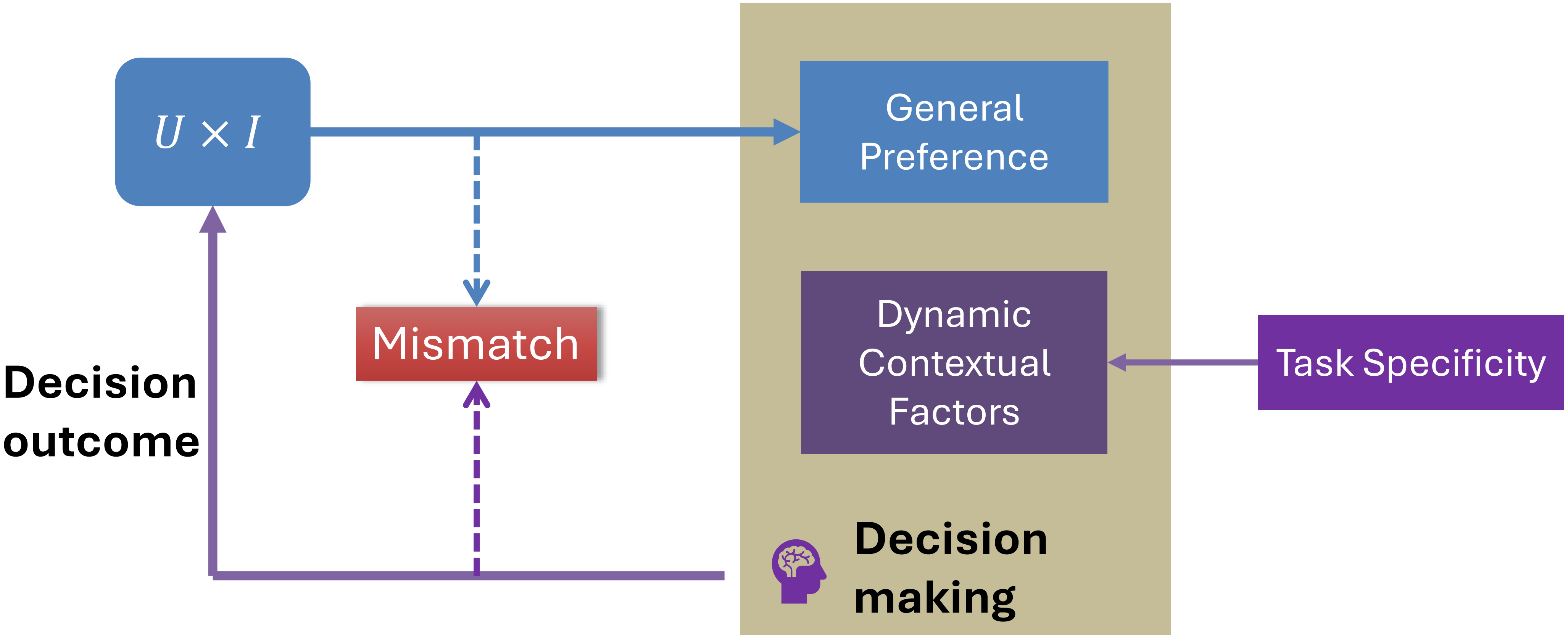
RecSys is a **Search Problem**: CF Generates Part of the Query



$U \times I$: user-item matrix of all users and all items
I_u: u_1 historical interactions
I_c: u_1 interactions in the current session

- **Query** in **implicit** form
  - General preference
  - Current context
- **Item collection**
  - Dynamically updated
- **Ranking**
  - Aiming for positive decision making

The Mismatch

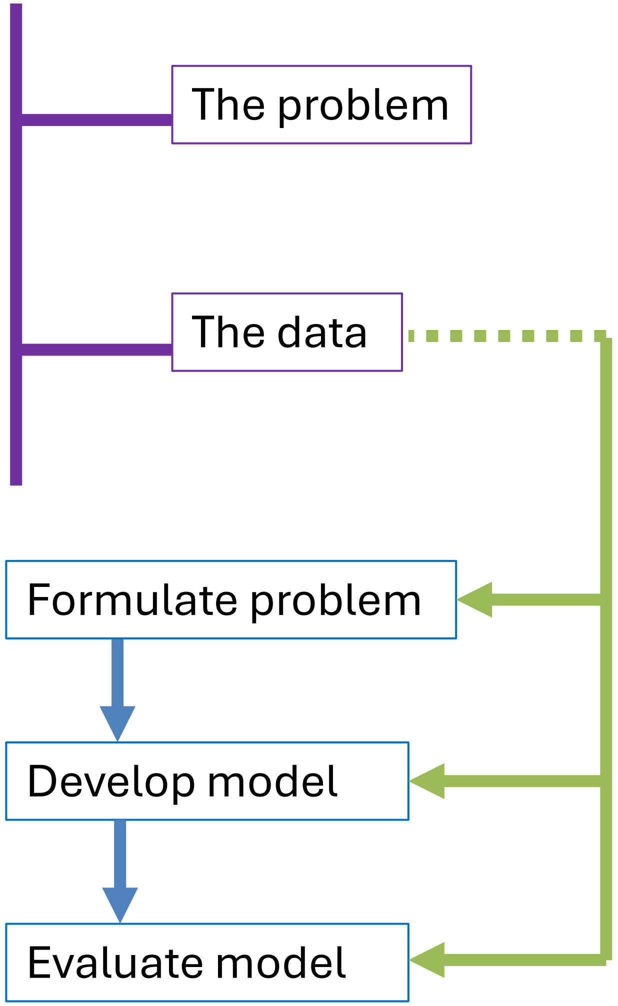


Understanding of RecSys

- User interaction/decision is influenced by **multiple** factors.
  - Long-term general preferences + Short-term dynamic contextual factors.
  - The **relative importance** of these factors **varies across applications**.

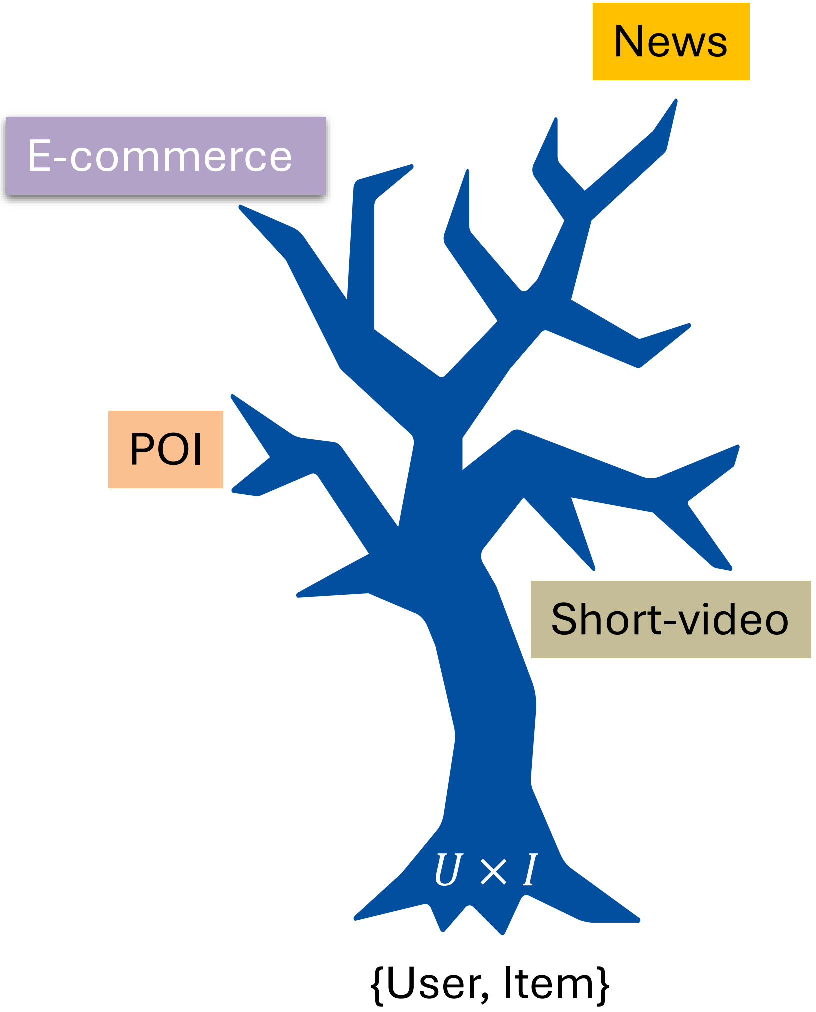
- CF is good at modeling user general preferences; Offline evaluation methods tend to focus on capturing general preferences
  - General preference is less time-dependent, change relatively slowly over time
  - Data leakage is less likely to significantly impact offline model results;
  - Hence, time dimension is often ignored in RecSys research/evaluation.

- When deployed online, models deemed good based on offline evaluation may exhibit unpredictable performance.
  - Depending on the **significance of dynamic factors** in that specific application.
  - If general preferences are predominant, then the model is more likely to perform well.



What's next?

- Extremely challenging to find a perfect offline evaluation scheme
  - Every model can be a winner remains
  - It is hard to find one model fitting all RecSys scenarios
- Models shall be designed and evaluated for a **pre-defined type of application**
- Item-kNN remains a strong baseline; The definition of "nearest" is feature engineering
  - Task dependent, and can be applied in a dynamic manner
  - There exist a diverse form of neighbours
  - Can be modelled by a sequential model if applied in a **session-based** manner



Thank you!

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

sig web

Beyond Collaborative Filtering: A Relook at Task Formulation in Recommender Systems

AIXIN SUN
Nanyang Technological University, Singapore

Recommender Systems (RecSys) have become indispensable in numerous applications, profoundly influencing our everyday experiences. Despite their practical significance, academic research in RecSys often abstracts the formulation of research tasks from real-world contexts, aiming for a clean problem formulation and more generalizable findings. However, it is observed that there is a lack of collective understanding in RecSys academic research. The root of this issue may lie in the simplification of research task definitions, and an overemphasis on modeling the decision outcomes rather than the decision-making process. That is, we often conceptualize RecSys as the task of predicting missing values in a *static* user-item interaction matrix, rather than predicting a user's decision on the next interaction within a *dynamic*, *changing*, and *application-specific* context. There exists a mismatch between the inputs accessible to a model and the information available to users during their decision-making process, yet the model is tasked to predict users' decisions. While collaborative filtering is effective in learning general preferences from historical records, it is crucial to also consider the dynamic contextual factors in practical settings. Defining research tasks based on application scenarios using domain-specific datasets may lead to more insightful findings. Accordingly, viable solutions and effective evaluations can emerge for different application scenarios.