AI6122 Text Data Management & Analysis

Topic 10: Recommender Systems

Dr. Sun Aixin

A simple introduction with (personal) opinion

Introduction

- Recommendation systems (recommenders, RecSys) have become an important research area since the appearance of the first papers on collaborative filtering in the **mid-1990**s
  - There has been much work done both in the industry and academia on developing new approaches to recommender systems over the last decade.
  - RecSys is a problem-rich research area.
  - RecSys help users to deal with information overload and provide personalized recommendations, content, and services to them.
- Roots of recommender systems can be traced back to the extensive work in many areas
  - cognitive science, approximation theory, information retrieval, forecasting theories
  - also links to management science and consumer choice modeling in marketing
- (published in 2005, and remains applicable)

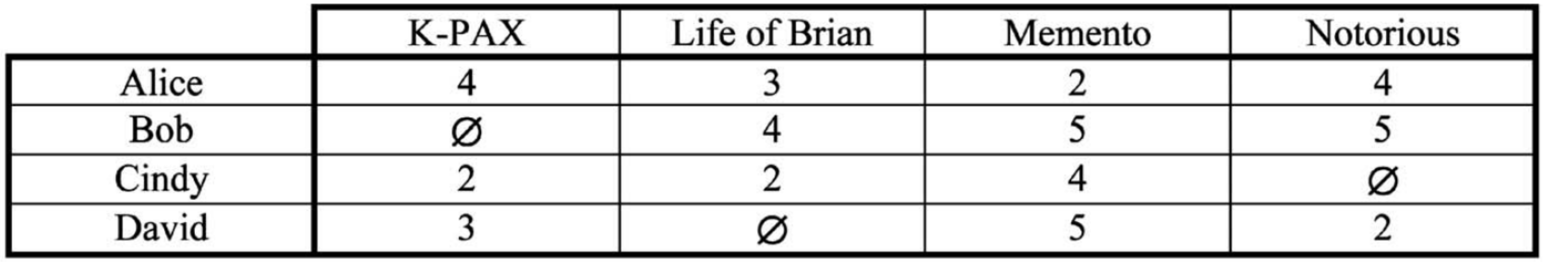
G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," in IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp. 734-749, June 2005, doi: 10.1109/TKDE.2005.99.

Problem definition

- Recommender System problem
  - Let C be the set of all users; user space can also be very large, millions in some cases.
  - Let S be the set of all possible items that can be recommended. The space of S can be very large, ranging in hundreds of thousands or even millions of items, e.g., books.
  - Let u be a utility function that measures the usefulness of item s to user c,
  - Then, for each user $c \in C$, we want to choose such item $s' \in S$ that maximizes the user's utility.
- "usefulness" is often referred to as user preference
  - Explicit feedback: John rated movie Gladiator a rating 5 (among 1 to 5).
  - Implicit feedback: John bought movie Gladiator
- Non-personal recommendation: popularity

Rating prediction vs Top-N recommendation

A Fragment of a Rating Matrix for a Movie Recommender System



	K-PAX	Life of Brian	Memento	Notorious
Alice	4	3	2	4
Bob	∅	4	5	5
Cindy	2	2	4	∅
David	3	∅	5	2

- Users can be in millions, and items can be in millions as well
  - Sparsity: the rating matrix is **very very very sparse**; We have few known entries, but many many unknow entries
  - The key problem: to estimate the utility function u from known entries, then apply u to estimate the unknown entries
  - Unknown entries: user has never seen the item; user has seen the item but does not give any response
- Two problems:
  - **Rating prediction**: will Marry give a rating 5 to Gladiator?
  - **Top-N recommendation**: which10 movies shall a system recommend to Marry?

Approaches

- **Content-based** recommendations
  - John liked Gladiator, then John likely will like movies similar to Gladiator
- **Collaborative** recommendations
  - John shares similar tastes as a few other users
  - Then John likely will like movies preferred by those users.
- Hybrid approaches
  - Combine multiple recommendations like collaborative and content-based methods.
- Deep learning models
  - There are a plenty of deep learning models

Content-based recommendation

- If a user likes items A, B, C, then we can find items that are similar to A, B, and C, and recommend these items to the user.

- Information retrieval
  - Query: Items A, B, C, e.g., represented by keywords or other features
  - Documents: The remaining items (as candidate items) to be "retrieved", represented as keywords or other features in the same feature space
  - Relevance model:
    - cosine similarity or other relevance models.
    - Or model as a classification problem with liked items and disliked items.
- Limitations:
  - Items are limited by the features than can be used to characteries them.
  - Overspecification: the user is limited to being recommended items that are similar to those already preferred.

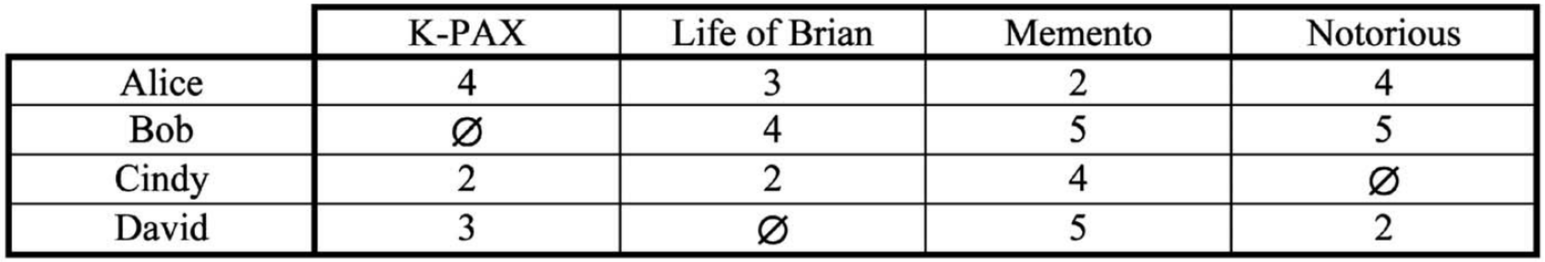
Collaborative recommender systems

- Also known as collaborative filtering systems
  - try to predict the utility of items for a particular user c based on the items previously rated by **other users**

- Memory-based collaborative algorithms:
  - **User-User collaborative filtering**: Given a user, find her N most similar users, then recommend based on the aggregated preferences by these N similar users.
    - Similarity between users can be computed based on the items that have been rated by both users: Users A rated movies: M1, M2, M3, M4, M5. User B rated movies: M2, M4, M5, M6, M7, M8.
  - **Item-Item collaborative filtering**: Given an item, find its N most similar items based on user interactions. Then we recommend items that are similar to the already preferred items of a user.
    - Similarity between items is computed by their common interactions: Item A was rated by U1, U2, U3, U4, U5, Item B was rated by U2, U4, U5, U6, U8.

User kNN vs Item kNN

A Fragment of a Rating Matrix for a Movie Recommender System

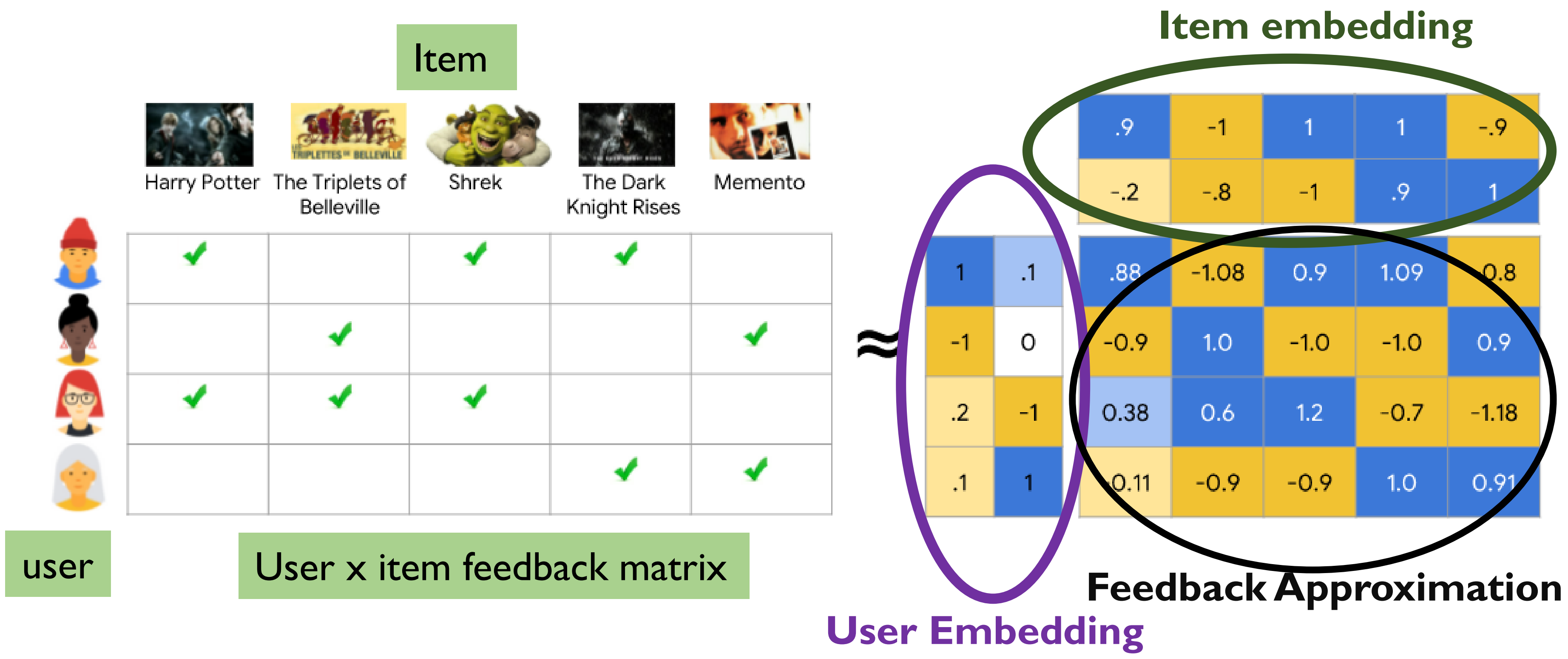


	K-PAX	Life of Brian	Memento	Notorious
Alice	4	3	2	4
Bob	∅	4	5	5
Cindy	2	2	4	∅
David	3	∅	5	2

- User kNN: user is represented by a vector of items (each item is one dimension)
- Item kNN: item is represented by a vector of users (each user is one dimension)
- Choose similarity function and aggregation

Collaborative recommender systems

- Model-based collaborative algorithms: learn a low-dimensional user (resp. item) representation, such that user and item product indicate preference.
  - Matrix factorization, deep learning, and many other models



	Harry Potter	The Triplets of Belleville	Shrek	The Dark Knight Rises	Memento
	✓		✓	✓	
		✓			✓
	✓	✓	✓		
				✓	✓

Item embedding

.9	-1	1	1	-.9
-.2	-.8	-1	.9	1

User Embedding

1	.1
-1	0
.2	-1
.1	1

Feedback Approximation

.88	-1.08	0.9	1.09	-0.8
-0.9	1.0	-1.0	-1.0	0.9
0.38	0.6	1.2	-0.7	-1.18
-0.11	-0.9	-0.9	1.0	0.91

Source: https://developers.google.com/machine-learning/recommendation/collaborative/matrix

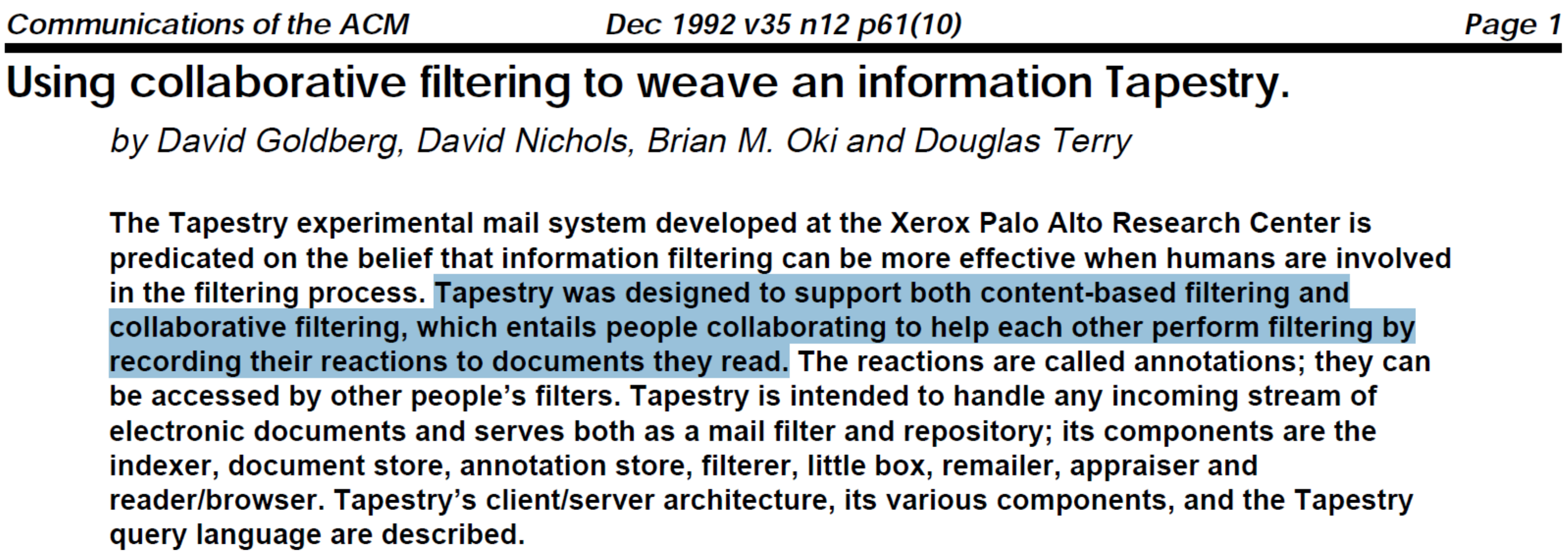
There are many studies on recommendation systems

- Survey papers on different perspectives of RecSys:
  - A Survey on Session-based Recommender Systems. ACM Comput. Surv. 54(7): 154:1-154:38 (2022)
  - Deep Learning Based Recommender System: A Survey and New Perspectives. ACM Comput. Surv. 52(1): 5:1-5:38 (2019)
  - A Survey on Stream-Based Recommender Systems. ACM Comput. Surv. 54(5): 104:1-104:36 (2021)
  - Explainable Recommendation: A Survey and New Perspectives. Found. Trends Inf. Retr. 14(1): 1-101
  - A Survey on Knowledge Graph-Based Recommender Systems. CoRR abs/2003.00911 (2020)
  - Efficient Retrieval of Matrix Factorization-Based Top-k Recommendations: A Survey of Recent Approaches. J. Artif. Intell. Res. 70: 1441-1479 (2021)
  - A Survey on Reinforcement Learning for Recommender Systems. CoRR abs/2109.10665 (2021)
- And many more: https://dblp.org/search?q=recommend%20survey

Recommendation is hard

Open discussion

Recommendation: Information filtering


Communications of the ACM *Dec 1992 v35 n12 p61(10)* *Page 1*

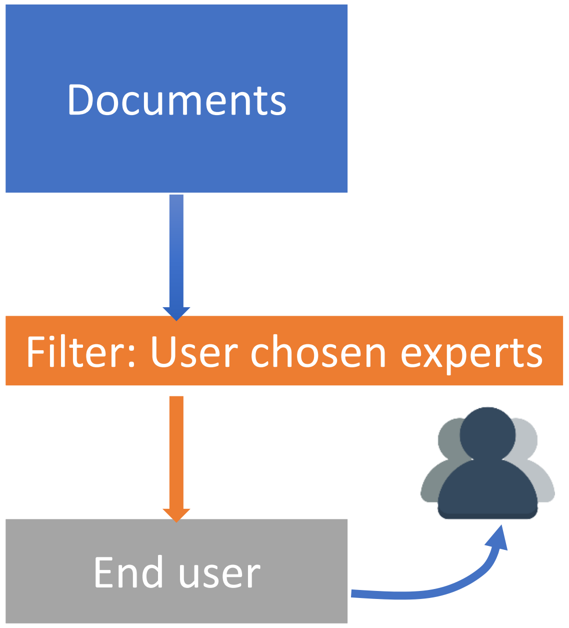
Using collaborative filtering to weave an information Tapestry.

by David Goldberg, David Nichols, Brian M. Oki and Douglas Terry

The Tapestry experimental mail system developed at the Xerox Palo Alto Research Center is predicated on the belief that information filtering can be more effective when humans are involved in the filtering process. Tapestry was designed to support both content-based filtering and collaborative filtering, which entails people collaborating to help each other perform filtering by recording their reactions to documents they read. The reactions are called annotations; they can be accessed by other people's filters. Tapestry is intended to handle any incoming stream of electronic documents and serves both as a mail filter and repository; its components are the indexer, document store, annotation store, filterer, little box, remailer, appraiser and reader/browser. Tapestry's client/server architecture, its various components, and the Tapestry query language are described.

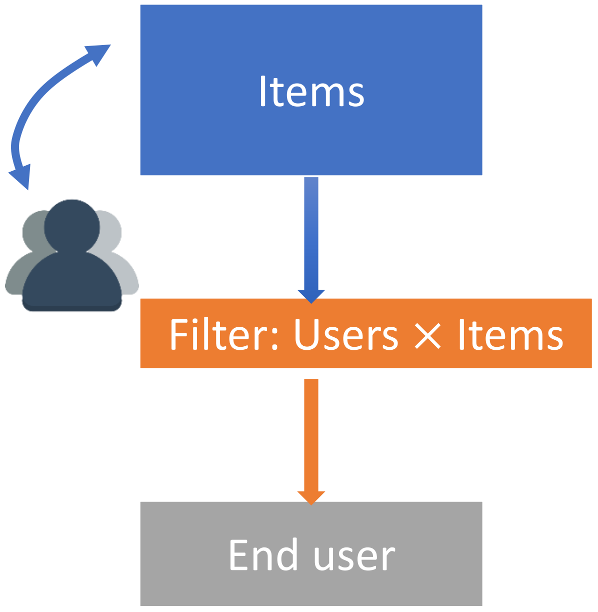
- Suppose you like to receive "interesting" documents from NetNews newsgroup,
  - But you don't know how to write a search expression that characterizes them, and you don't have time to read them all yourself.
  - However, you know that Smith and Jones read all of NetNews newsgroup, and reply to the more interesting documents.
- Tapestry allows you to filter on "documents replied to by Smith and Jones"

Recommender Systems -- 1992

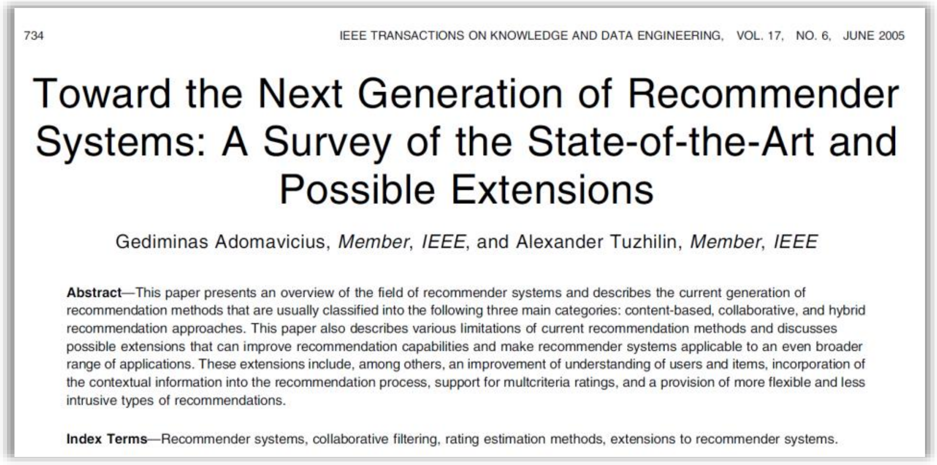


- User does not access all documents
- User trusts "recommendations" by self-defined "experts"
- Recommendation → information filter
  - Weibo
  - Twitter
  - WeChat?
- User information needs
  - Represented by the "experts" chosen by users

Recommender Systems -- 2005



User information needs: defined by other "similar" users

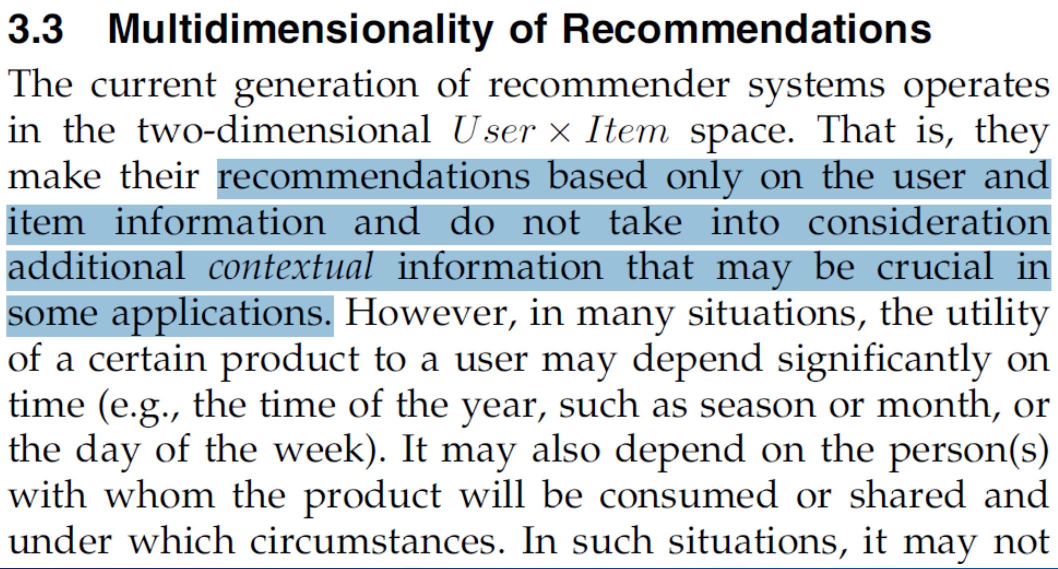

734 IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 17, NO. 6, JUNE 2005

Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions

Gediminas Adomavicius, *Member, IEEE*, and Alexander Tuzhilin, *Member, IEEE*

Abstract—This paper presents an overview of the field of recommender systems and describes the current generation of recommendation methods that are usually classified into the following three main categories: content-based, collaborative, and hybrid recommendation approaches. This paper also describes various limitations of current recommendation methods and discusses possible extensions that can improve recommendation capabilities and make recommender systems applicable to an even broader range of applications. These extensions include, among others, an improvement of understanding of users and items, incorporation of the contextual information into the recommendation process, support for multcriteria ratings, and a provision of more flexible and less intrusive types of recommendations.

Index Terms—Recommender systems, collaborative filtering, rating estimation methods, extensions to recommender systems.


3.3 Multidimensionality of Recommendations

The current generation of recommender systems operates in the two-dimensional *User* × *Item* space. That is, they make their recommendations based only on the user and item information and do not take into consideration additional *contextual* information that may be crucial in some applications. However, in many situations, the utility of a certain product to a user may depend significantly on time (e.g., the time of the year, such as season or month, or the day of the week). It may also depend on the person(s) with whom the product will be consumed or shared and under which circumstances. In such situations, it may not

Are we missing anything?

- Recommender System problem
  - Let C be the set of all users; user space can also be very large, millions in some cases.
  - Let S be the set of all possible items that can be recommended. The space of S can be very large, ranging in hundreds of thousands or even millions of items, e.g., books.
  - Let u be a utility function that measures the usefulness of item s to user c,
  - Then, for each user $c \in C$, we want to choose such item $s' \in S$ that maximizes the user's utility.

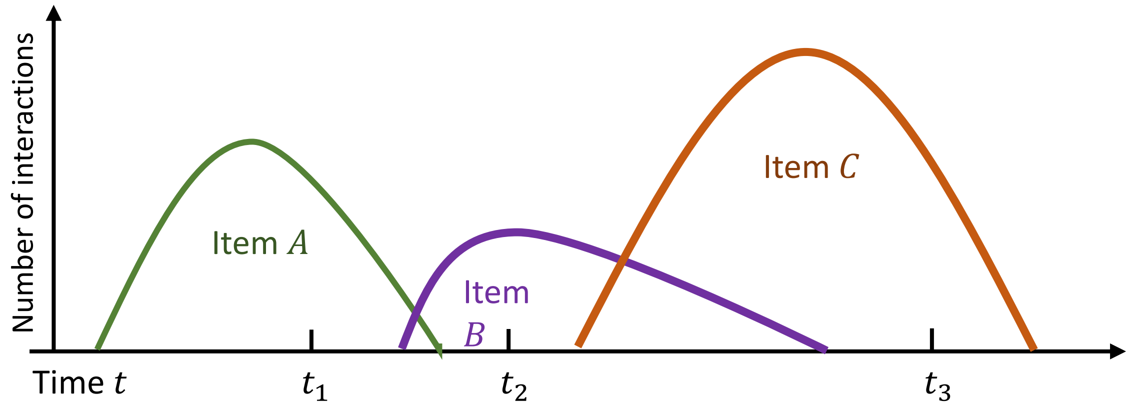
- "usefulness" is often referred to as user preference
  - Explicit feedback: John rated movie Gladiator a rating 5 (among 1 to 5).
  - Implicit feedback: John bought movie Gladiator

What about the time factor?

Recommendation by Popularity

- **Non-personal recommendation**: popularity
  - Most popular items in common understanding
  - Popularity reflects the trending items during a time period

- The most popular items are item A at t_1, B at t_2 and C at t_3 respectively



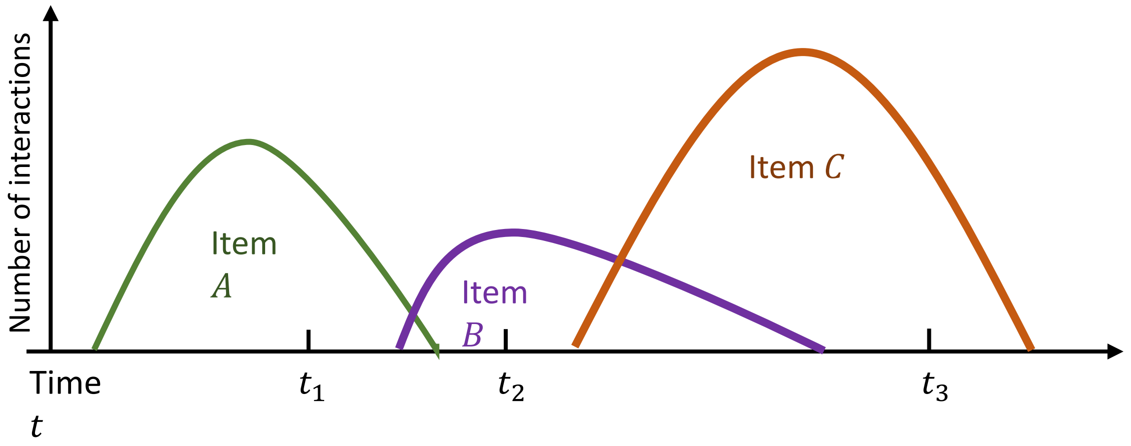
Recommendation by Popularity

- **Mainstream definition** of popularity in research paper
  - Non-personalized method
  - Overall popularity (number of explicit/implicit feedbacks) in the training set

Top-tier papers in KDD, SIGIR, WWW, RecSys	Open-source recommendation tools
MPR [1], entity2rec [2], CSE [3], TransFM [4], RKGE [5], NCE-PLRec [6], s^{2}Meta [7], NeuMF [8], FG-ACAE [9], set-based model [11] Thompson sampling bandit [10] AreWe Really Making Much Progress? A worrying Analysis of Recent Neural Recommendation Approaches [12]	LibRec [13], RecommendedLab [14], CaseRecommender [15], TagRec [16] Microsoft Recommender [17] Sequence-Based-Recommender [18]

Recommendation by Popularity

- Mainstream MostPop definition in research paper
  - Ignorance of the time dimension
  - The most popular items are item C followed by A and B, based on the training set.
  - It does not reflect the common understanding of popularity



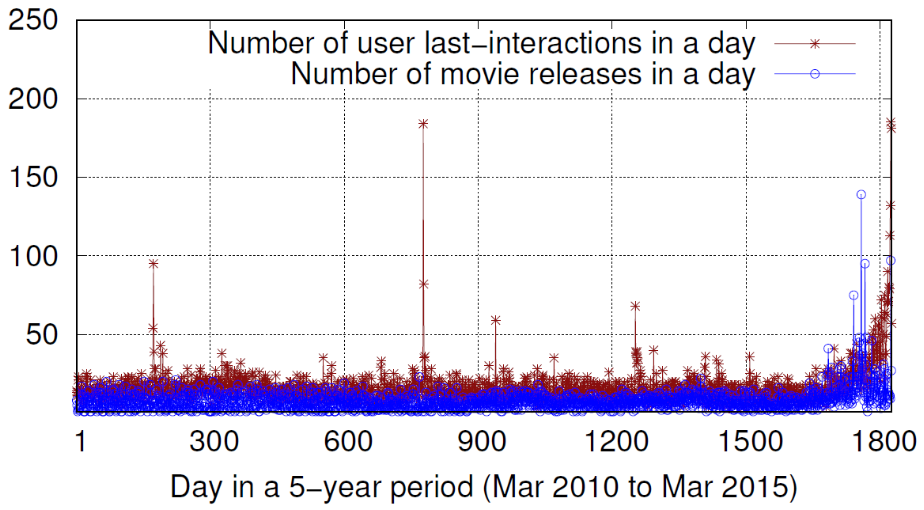
MostPop Definition vs Evaluation

- MostPop, by definition, does not consider time dimension in dataset.

- Dataset partitioning
  - Randomly sample a percentage of <user, item, interaction> as training set, and treat the remaining tuples as test set.
  - Randomly split a user's interactions into training and test set by ratio (e.g., 8:2)
  - Leave-one-out evaluation: Holding the last interaction of each user as test instances and treat the remaining interactions as training set

Offline Evaluation of MostPop

➢ Leave-one-out evaluation: Holding the last interaction of each user as test instances and treat the remaining interactions as training set

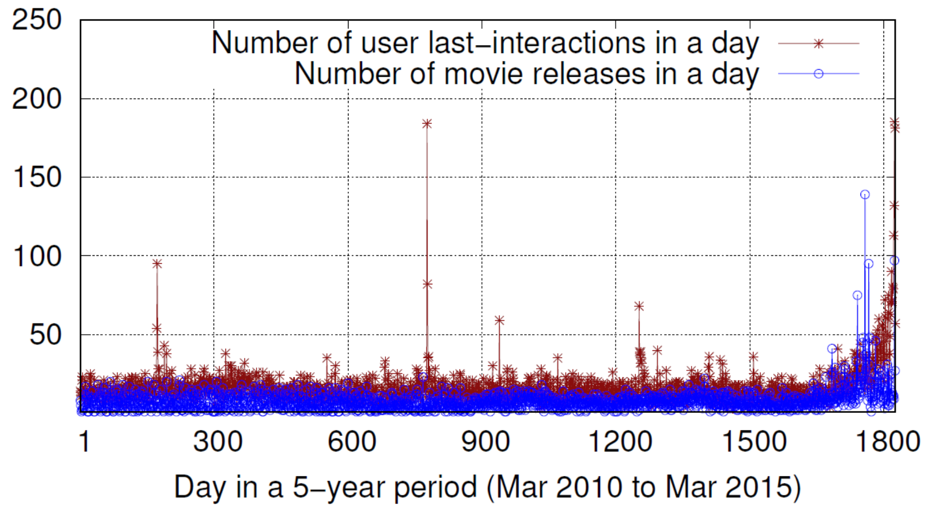


Number of movie releases and number of user last-interactions on each day in MovieLens

- Users may **leave the system** at any time.
- New items **join the system** at any time.

Offline Evaluation of MostPop

- Leave-one-out evaluation: Holding the last interaction of each user as test instances and treat the remaining interactions as training set

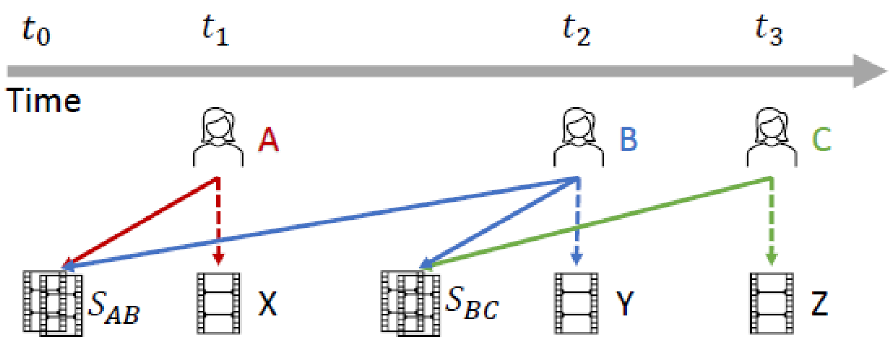


Number of movie releases and number of user last-interactions on each day in MovieLens

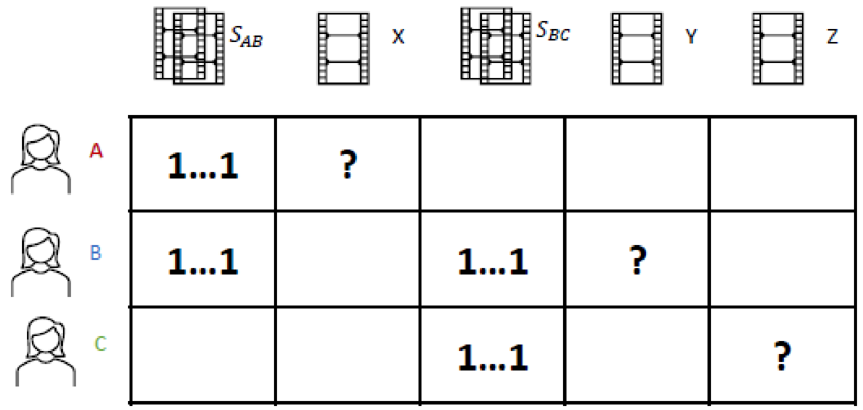
In common understanding, users leaving the system on the first day will not have opportunities to rate any movies released after the first day.

Using MostPop and leave-one-out setting, there is chance that movies released after the first day are recommended to users who leave the system on the first day.

Time factor in collaborative filtering models



(a) User-item interaction along global timeline. (best viewed in color)



	S_{AB}	X	S_{BC}	Y	Z
A	1…1	?			
B	1…1		1…1	?	
C			1…1		?

(b) User-item interaction matrix

- The problem becomes **much more difficult to address** if considering time
- Sparsity is extremely difficult to handle
- Many datasets do not provide much context information
- User choices are affected by existing models

Reference: https://arxiv.org/abs/2010.11060

Summary

- A brief introduction on recommender systems
- Explicit feedback vs implicit feedback
- Recommendation models

- Open discussion on recommender systems.