Recommender Systems

Marie Al-Ghossein

Crossing Minds

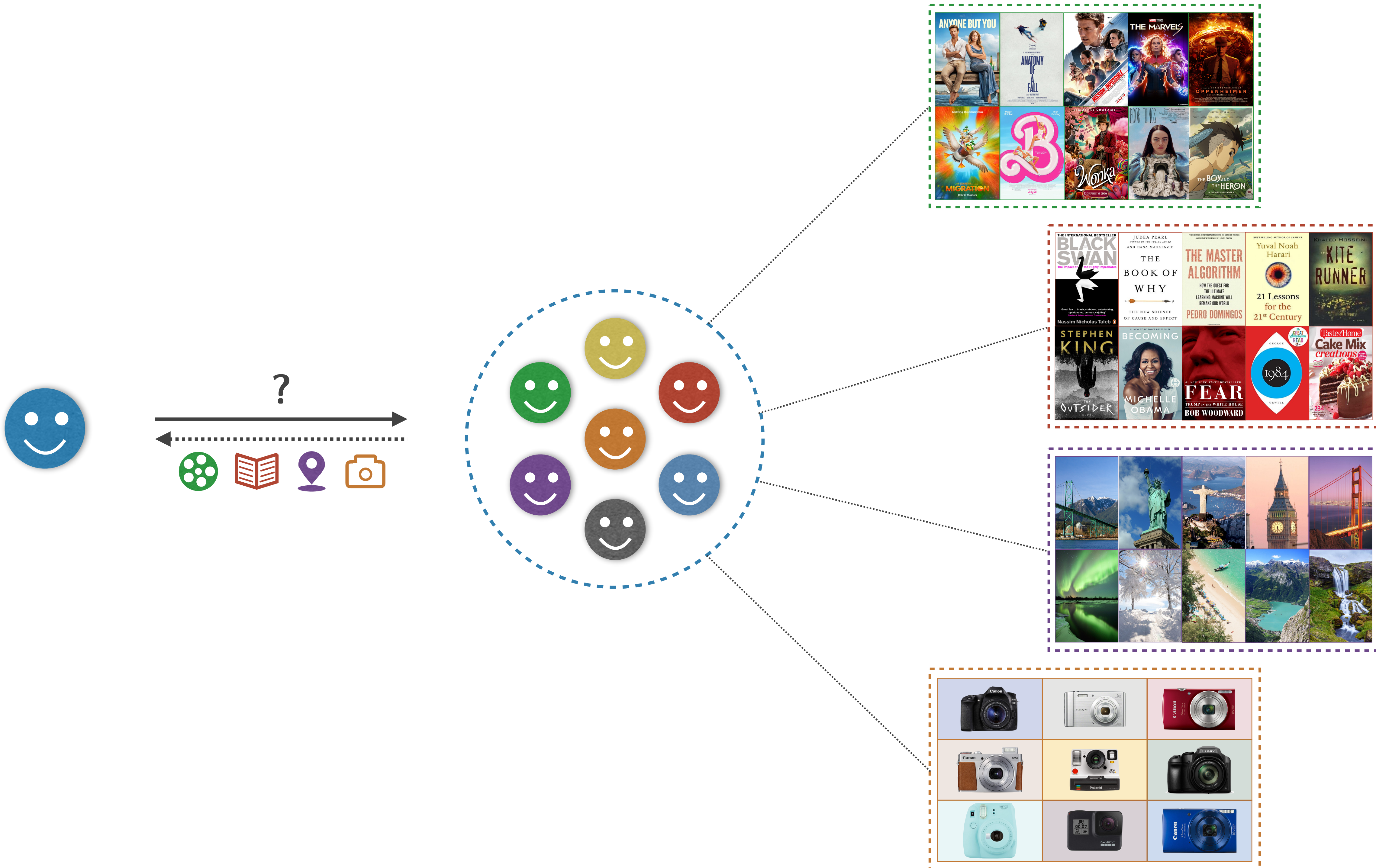
June 2, 2025

Outline

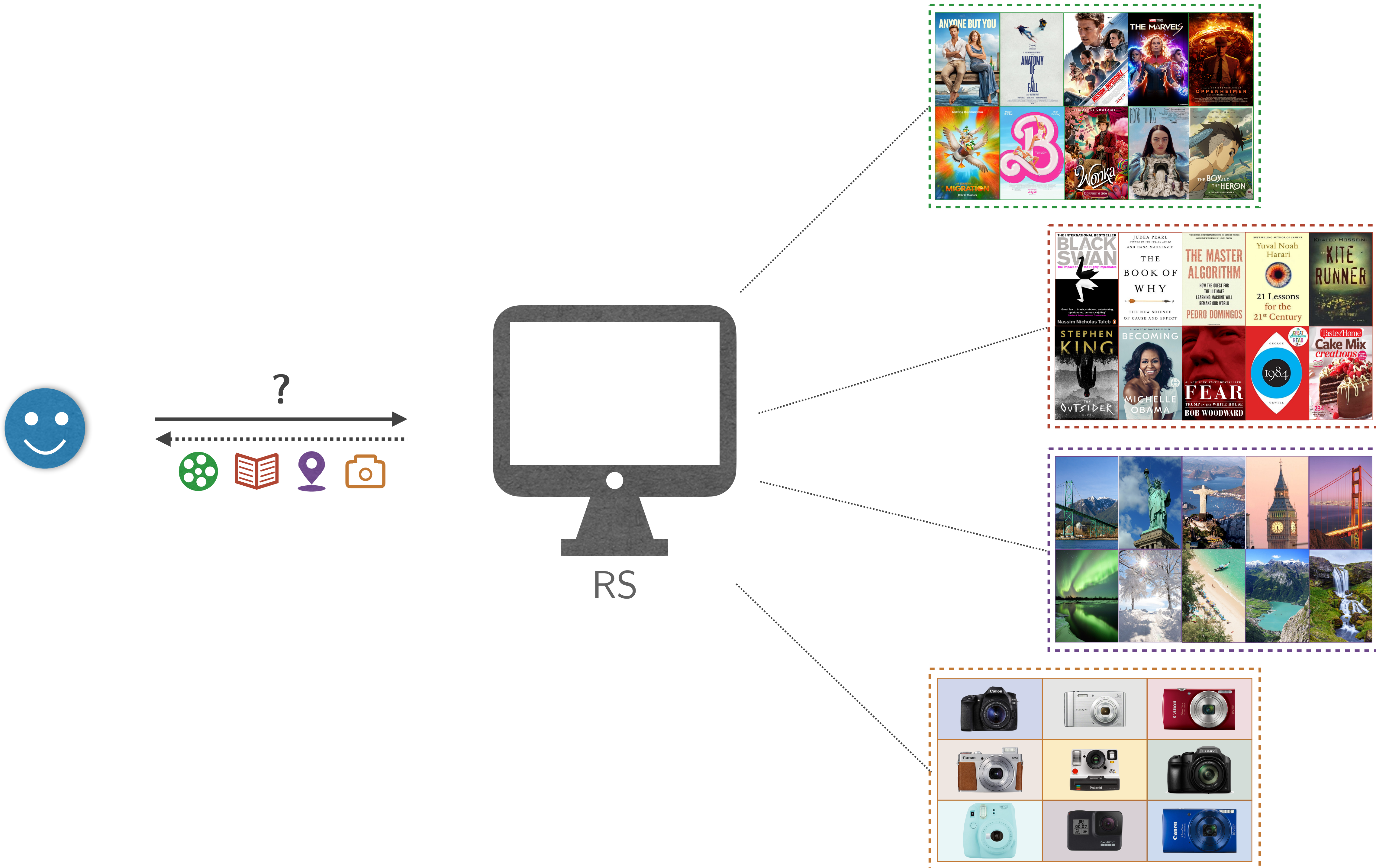
1. Introduction
2. The Recommendation Problem
3. User Feedback
4. Challenges
5. Evaluation of RS
6. Non-Personalized Recommendation
7. Collaborative Filtering
8. Content-Based Filtering
9. Hybrid Approaches
10. Context-Aware Recommendation
11. Advanced Topics for RS (RL, DL, and LLMs)
12. A Practical Example: Recommendation in Location-Based Social Networks

Introduction

The (black) art of recommendation



The (black) art of recommendation



The value of recommender systems

For consumers

- Find good and relevant products
- Improve decision making
- Help them overcome the *information overload*

The value of recommender systems

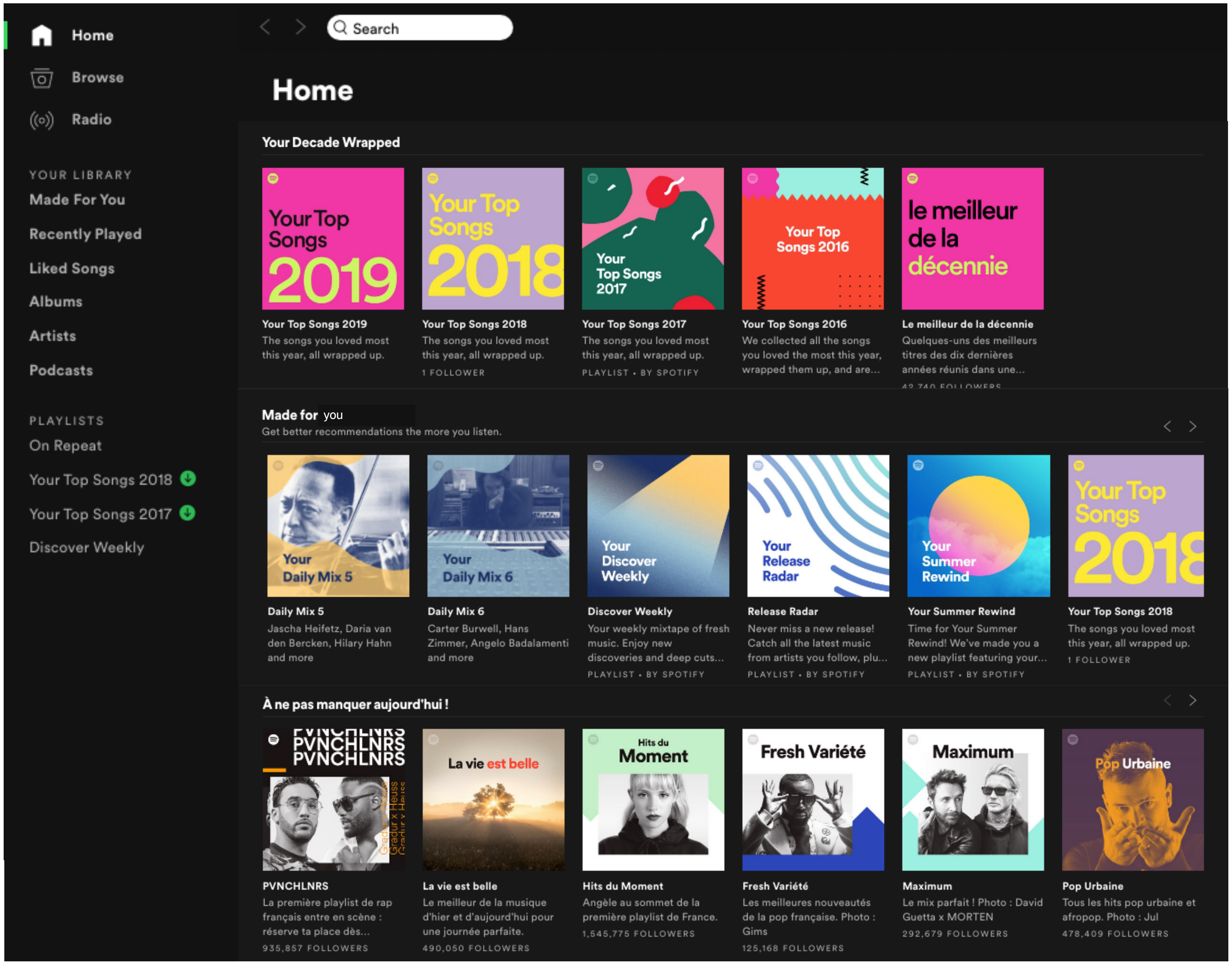
For consumers

- Find good and relevant products
- Improve decision making
- Help them overcome the *information overload*

For service providers

- Increase the number of products sold
- Sell more diverse products
- Increase the user satisfaction and fidelity
- Learn more about consumers

Personalization at Spotify



The value of recommender systems

"Our recommender system influences choice for about 80% of hours streamed at Netflix. [...] We think the combined effect of personalization and recommendations save us more than $1B per year." (2015)

C. A. Gomez-Uribe and N. Hunt. *The Netflix Recommender System: Algorithms, Business Value, and Innovation.* ACM Trans. Management Inf. Syst. 6(4), 2015.

The value of recommender systems

"Our recommender system influences choice for about 80% of hours streamed at Netflix. [...] We think the combined effect of personalization and recommendations save us more than $1B per year." (2015)

C. A. Gomez-Uribe and N. Hunt. *The Netflix Recommender System: Algorithms, Business Value, and Innovation.* ACM Trans. Management Inf. Syst. 6(4), 2015.

"35% of what consumers purchase on Amazon comes from product recommendations." (2013)

https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers

The value of recommender systems

"Our recommender system influences choice for about 80% of hours streamed at Netflix. [...] We think the combined effect of personalization and recommendations save us more than $1B per year." (2015)

C. A. Gomez-Uribe and N. Hunt. *The Netflix Recommender System: Algorithms, Business Value, and Innovation.* ACM Trans. Management Inf. Syst. 6(4), 2015.

"35% of what consumers purchase on Amazon comes from product recommendations." (2013)

https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers

"Recommendations account for about 60% of all video clicks from the homepage." (2010)

J. Davidson et al. *The YouTube Video Recommendation System.* In RecSys'10, 2010.

The value of recommender systems

"Our recommender system influences choice for about 80% of hours streamed at Netflix. [...] We think the combined effect of personalization and recommendations save us more than $1B per year." (2015)

C. A. Gomez-Uribe and N. Hunt. *The Netflix Recommender System: Algorithms, Business Value, and Innovation.* ACM Trans. Management Inf. Syst. 6(4), 2015.

"35% of what consumers purchase on Amazon comes from product recommendations." (2013)

https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers

"Recommendations account for about 60% of all video clicks from the homepage." (2010)

J. Davidson et al. *The YouTube Video Recommendation System.* In RecSys'10, 2010.

"Recommendations generate 38% more clicks compared to popularity-based recommendations." (2009)

A. S. Das et al. *Google News Personalization: Scalable Online Collaborative Filtering.* In WWW '07, 2007.

Measuring the business value of recommender systems

- Click-Through Rate (CTR)

Dietmar Jannach and Michael Jugovac. 2019. *Measuring the Business Value of Recommender Systems*. CoRR abs/1908.08328 (2019). arXiv:1908.08328

Measuring the business value of recommender systems

- Click-Through Rate (CTR)
- Conversion rates

Dietmar Jannach and Michael Jugovac. 2019. *Measuring the Business Value of Recommender Systems.* CoRR abs/1908.08328 (2019). arXiv:1908.08328

Measuring the business value of recommender systems

- Click-Through Rate (CTR)
- Conversion rates
- Sales and revenue

Dietmar Jannach and Michael Jugovac. 2019. *Measuring the Business Value of Recommender Systems*. CoRR abs/1908.08328 (2019). arXiv:1908.08328

Measuring the business value of recommender systems

- Click-Through Rate (CTR)
- Conversion rates
- Sales and revenue
- Effects on sales distributions

Dietmar Jannach and Michael Jugovac. 2019. *Measuring the Business Value of Recommender Systems*. CoRR abs/1908.08328 (2019). arXiv:1908.08328

Measuring the business value of recommender systems

- Click-Through Rate (CTR)
- Conversion rates
- Sales and revenue
- Effects on sales distributions
- User engagement and behavior

Dietmar Jannach and Michael Jugovac. 2019. *Measuring the Business Value of Recommender Systems.* CoRR abs/1908.08328 (2019). arXiv:1908.08328

What is a recommender system?

→ A Recommender System (RS) provides personalized suggestions of **items** for **users**, the items being drawn from a large catalog, based on knowledge of the user, the items, and **interactions** between users and items.

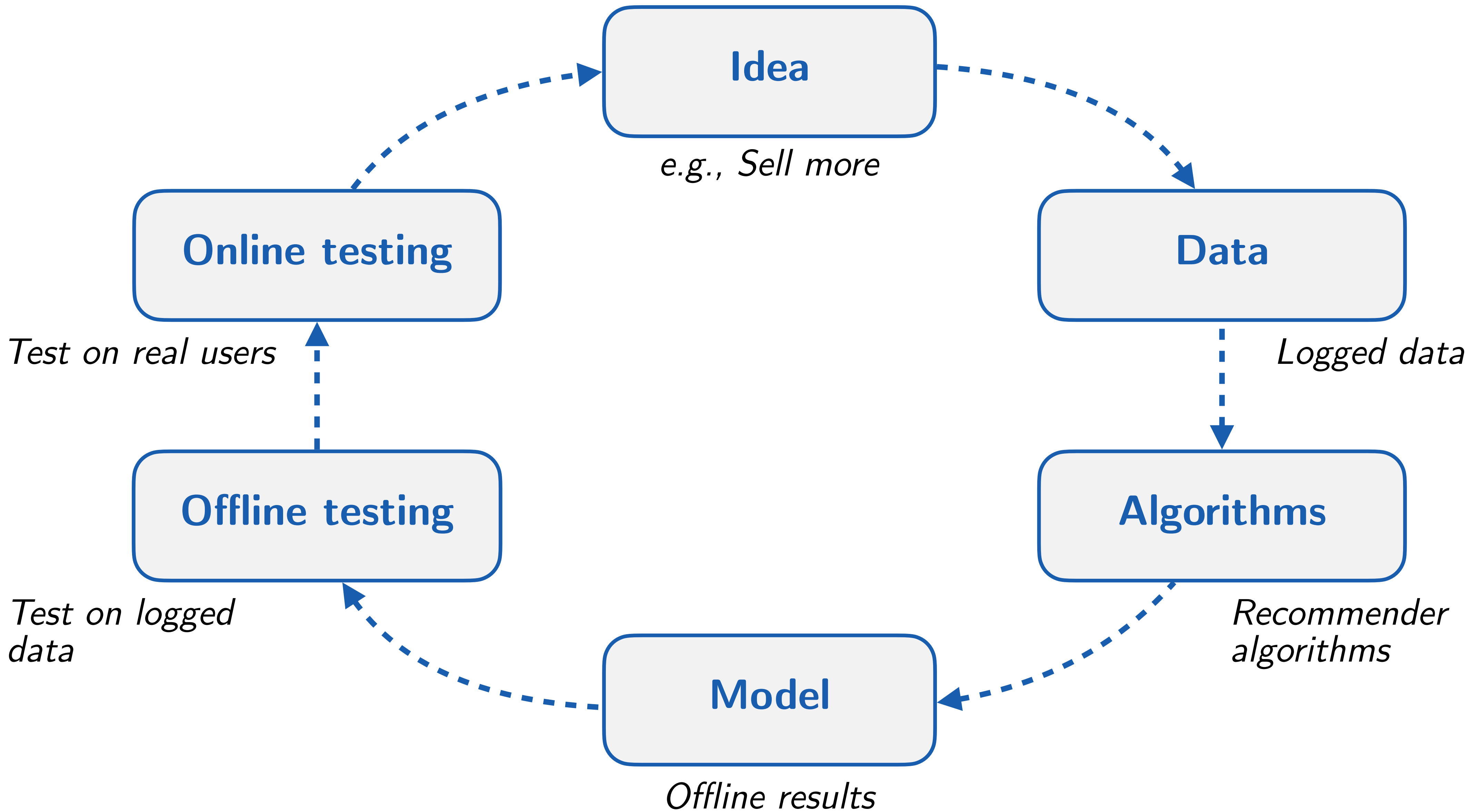
What is a recommender system?

- Knowledge of the user
  - Historical data - interactions with items
    - Purchases, clicks, views, ratings,...
  - Demographics
    - Age, gender, nationality, political views, income,...
  - Context
    - Device, location, time of day/week/year, alone or in a group,...

What is a recommender system?

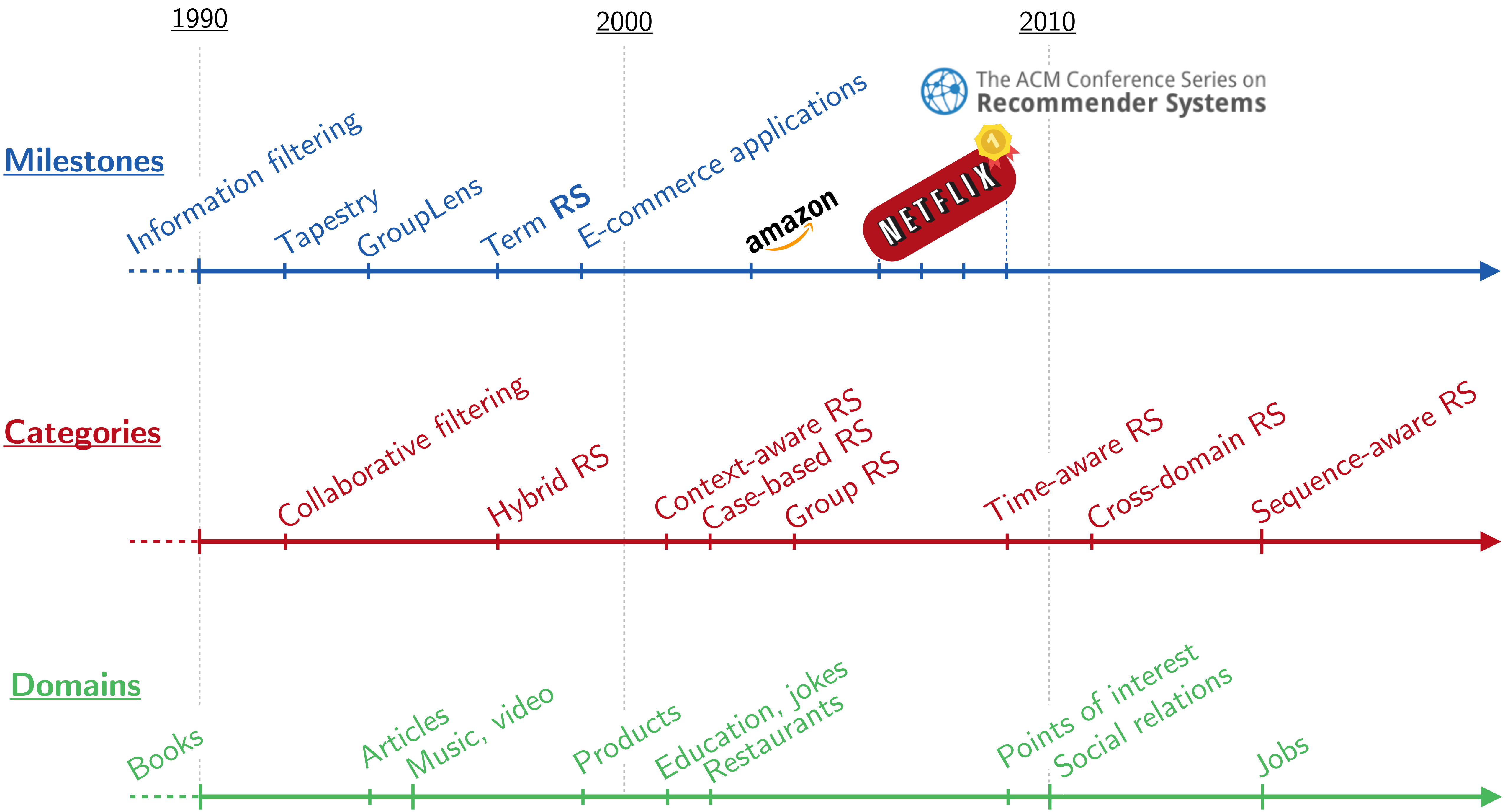
- Knowledge of the user
  - Historical data - interactions with items
    - Purchases, clicks, views, ratings,...
  - Demographics
    - Age, gender, nationality, political views, income,...
  - Context
    - Device, location, time of day/week/year, alone or in a group,...
- Knowledge of the item
  - Content information
    - Features, tags, text, images, multimedia signals,...

Building a recommender system

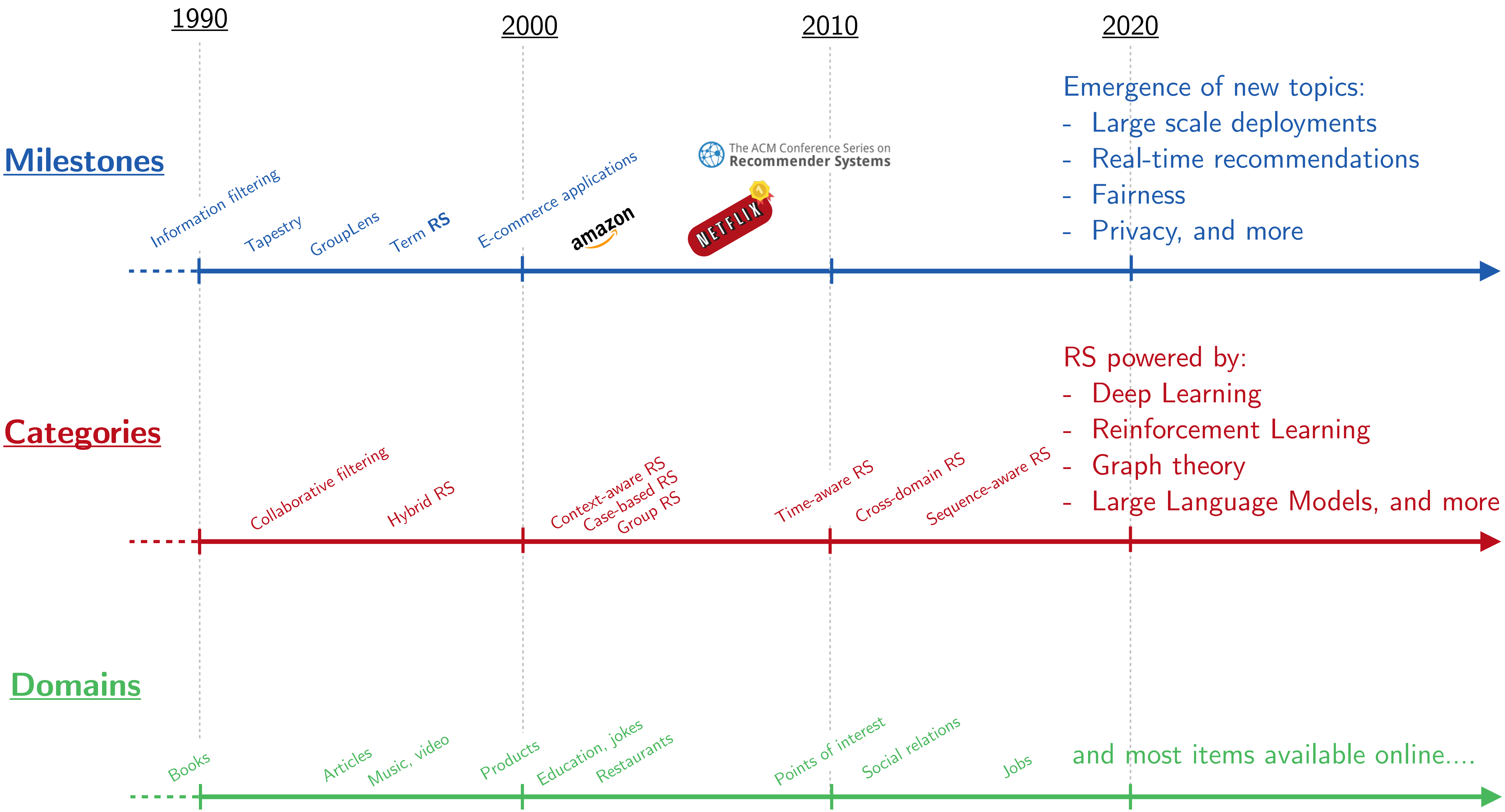


Kim Falk. 2019. *Practical Recommender Systems.* Manning, 2019.

Historical perspective on recommender systems



Historical perspective on recommender systems



The recommendation problem

The Netflix Prize (2006)

- **End-goal:** Recommend movies to Netflix users
- Competition problem:
  - Predict movie ratings on a 5-star scale
  - Dataset of 100M ratings
    from 480k users
    on 18k movies
  - $1M prize for 10% improvement over accuracy of existing solution

★★★★★	*New favorite!!!*
★★★★☆	*Enjoyed it!*
★★★☆☆	*Mixed feelings.*
★★☆☆☆	*Did not like it.*
★☆☆☆☆	*Hated it!*

Ratings

James Bennett and Stan Lanning. *The Netflix Prize.* In Proc. KDD Cup and Workshop, 2007.

The Netflix Prize (2006)

- **End-goal:** Recommend movies to Netflix users
- Competition problem:
  - Predict movie ratings on a 5-star scale
  - Dataset of 100M ratings
    from 480k users
    on 18k movies
  - $1M prize for 10% improvement over accuracy of existing solution

Ratings

→ Recommendation problem formulated as **rating prediction**

James Bennett and Stan Lanning. *The Netflix Prize.* In Proc. KDD Cup and Workshop, 2007.

The recommendation problem

Rating prediction

- The task of the RS is to predict the rating that a user will give to an item he did not rate.
  - Objective: Minimize the error between predicted and observed ratings
  - Recommendation: Items having the highest ratings for a user

The recommendation problem

Rating prediction

- The task of the RS is to predict the rating that a user will give to an item he did not rate.
  - Objective: Minimize the error between predicted and observed ratings
  - Recommendation: Items having the highest ratings for a user

Limitation: In real-world RS, it is more valuable to provide relevant top recommendations than accurately predicting ratings for all items, including those that are at the bottom list.

The recommendation problem

Rating prediction

- The task of the RS is to predict the rating that a user will give to an item he did not rate.
  - Objective: Minimize the error between predicted and observed ratings
  - Recommendation: Items having the highest ratings for a user

<u>Limitation</u>: In real-world RS, it is more valuable to provide relevant top recommendations than accurately predicting ratings for all items, including those that are at the bottom list.

Top-N recommendation

- The task of the RS is to predict the relevance of an item for a user, *i.e.*, predict whether the user will choose an item or not.

General formulation

- Let $\mathcal{U}$ be the set of users and $\mathcal{I}$ the set of items.
- Let f be a utility function that measures the usefulness, e.g., rating or relevance, of an item for a user: $f : \mathcal{U} \times \mathcal{I} \to \mathcal{R}$, where $\mathcal{R}$ is a totally ordered set.
- For each user $u \in \mathcal{U}$, we recommend the items maximizing f :

$$\forall u \in \mathcal{U}, i^* = \arg\max_{i \in \mathcal{I}} f(u, i)$$

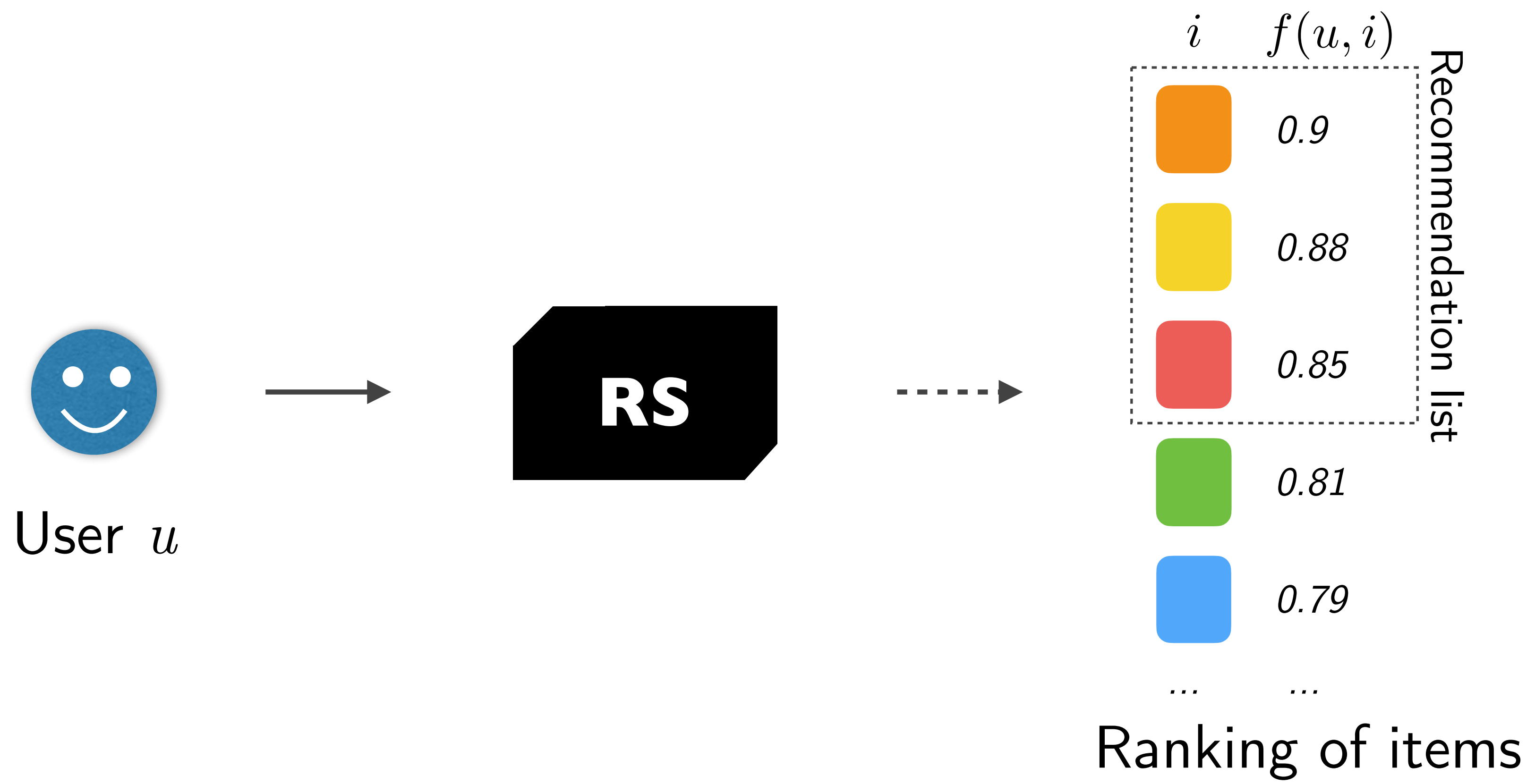
f can represent the rating or the relevance of an item

A recommendation algorithm

Algorithm 1 A recommendation algorithm

Input: user u, number of items to recommend N

1 **for** i in $\mathcal{I} \setminus \mathcal{I}_u$ **do**
2   Predict $\hat{r}_{ui}$
3 **end for**
4 Create list of $\mathcal{I} \setminus \mathcal{I}_u$ items ordered by decreasing order of $\hat{r}_{ui}$
5 Return N first items of the list

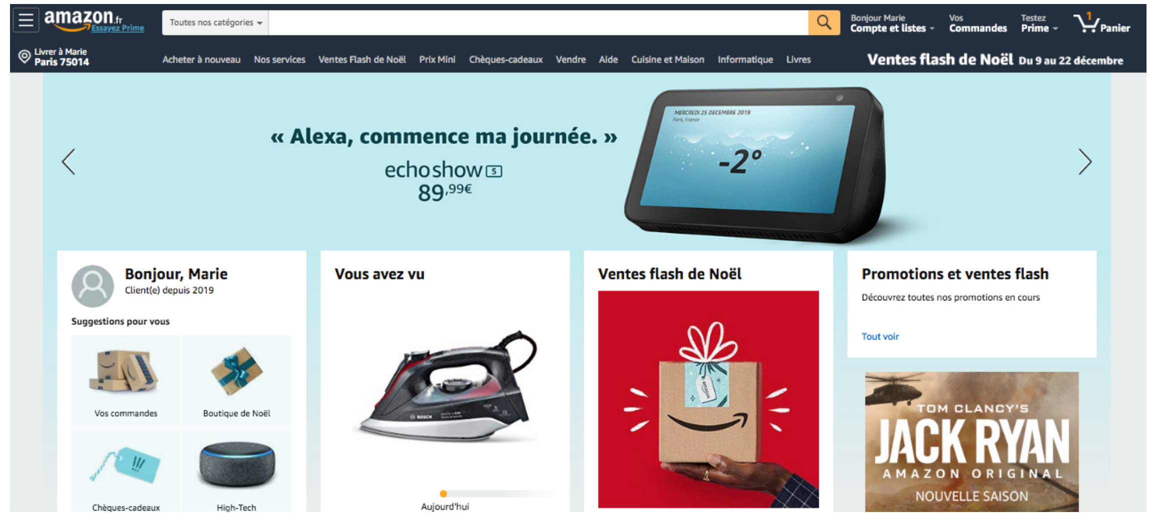


User feedback

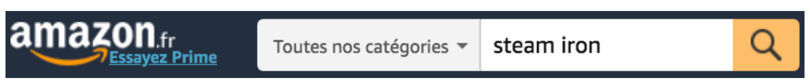
What is a recommender system?

- Knowledge of the user
  - Historical data - interactions with items
    - Purchases, clicks, views, ratings,...
  - Demographics
    - Age, gender, nationality, political views, income,...
  - Context
    - Device, location, time of day/week/year, alone or in a group,...
- Knowledge of the item
  - Content information
    - Features, tags, text, images, multimedia signals,...

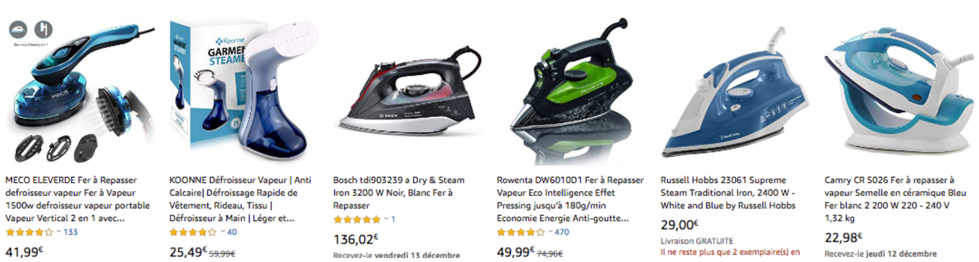
Browsing on Amazon...



Click on card #2



Search with keywords “w_1 w_2”



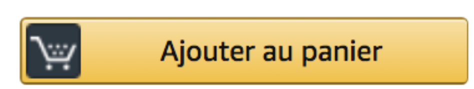
Impression with items #1, 2, 3, 4, 5, 6

Click on item #3

Informations sur le produit

Marque	Bosch
Numéro de modèle	TDI903239A
Couleur	Noir, Blanc
Poids de l'article	2,12 Kg
Dimensions du colis	33,8 x 20,2 x 12,4 cm
Capacité	0.4 litres
Puissance	3200 Watts
Matériau	Aluminium

Read description of item #3

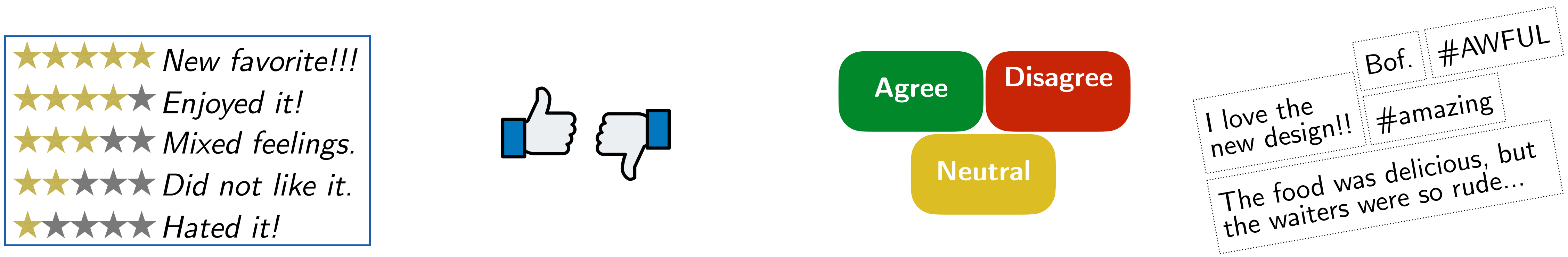


Add item #3 to cart

Rate and review item #3

Explicit feedback

- Explicit feedback is a form of feedback directly reported by the user to the system.



Ratings, 5-star scale **Like/Dislike** **Ordinal scale** **Textual reviews**

Explicit feedback

- Explicit feedback is a form of feedback directly reported by the user to the system.

Explicit feedback

- Explicit feedback is a form of feedback directly reported by the user to the system.
- Limitations:
  - Users do not always provide explicit feedback.

Explicit feedback

- Explicit feedback is a form of feedback directly reported by the user to the system.
- Limitations:
  - Users do not always provide explicit feedback.
  - The system may be unable to collect explicit feedback, e.g., cannot collect rating of people for Linkedin's "*People You May Know*".

Explicit feedback

- Explicit feedback is a form of feedback directly reported by the user to the system.
- Limitations:
  - Users do not always provide explicit feedback.
  - The system may be unable to collect explicit feedback, e.g., cannot collect rating of people for Linkedin's "*People You May Know*".
  - Is the rating revealing the true opinion of the user?

Explicit feedback

- Explicit feedback is a form of feedback directly reported by the user to the system.
- Limitations:
  - Users do not always provide explicit feedback.
  - The system may be unable to collect explicit feedback, e.g., cannot collect rating of people for Linkedin's "*People You May Know*".
  - Is the rating revealing the true opinion of the user?
  - Is the user being influenced by his social circle when rating an item?

Explicit feedback

- Explicit feedback is a form of feedback directly reported by the user to the system.
- Limitations:
  - Users do not always provide explicit feedback.
  - The system may be unable to collect explicit feedback, e.g., cannot collect rating of people for Linkedin's "*People You May Know*".
  - Is the rating revealing the true opinion of the user?
  - Is the user being influenced by his social circle when rating an item?
  - Is the user rating the complete package or is there an item's detail affecting the overall rating?

Explicit feedback

- Explicit feedback is a form of feedback directly reported by the user to the system.
- Limitations:
  - Users do not always provide explicit feedback.
  - The system may be unable to collect explicit feedback, e.g., cannot collect rating of people for Linkedin's "*People You May Know*".
  - Is the rating revealing the true opinion of the user?
  - Is the user being influenced by his social circle when rating an item?
  - Is the user rating the complete package or is there an item's detail affecting the overall rating?
  - Explicit feedback exhibit some biases, e.g., some users tend to give higher ratings than others for a similar level of appreciation.

Implicit feedback

- Implicit feedback is collected by the system without the intervention of the user.

Clicks

Informations sur le produit

Marque	Bosch
Numéro de modèle	TDI903239A
Couleur	Noir, Blanc
Poids de l'article	2,12 Kg
Dimensions du colis	33,8 x 20,2 x 12,4 cm
Capacité	0.4 litres
Puissance	3200 Watts
Matériau	Aluminium

Views of item's description

Purchases

Add to Wishlist

Implicit feedback

- Implicit feedback is collected by the system without the intervention of the user.

Clicks

Informations sur le produit

Marque	Bosch
Numéro de modèle	TDI903239A
Couleur	Noir, Blanc
Poids de l'article	2,12 Kg
Dimensions du colis	33,8 x 20,2 x 12,4 cm
Capacité	0.4 litres
Puissance	3200 Watts
Matériau	Aluminium

Views of item's description

Acheter cet article

Purchases

Ajouter à votre liste

Add to Wishlist

$\longrightarrow$ More abundant than explicit feedback

Implicit feedback

- Implicit feedback is collected by the system without the intervention of the user.

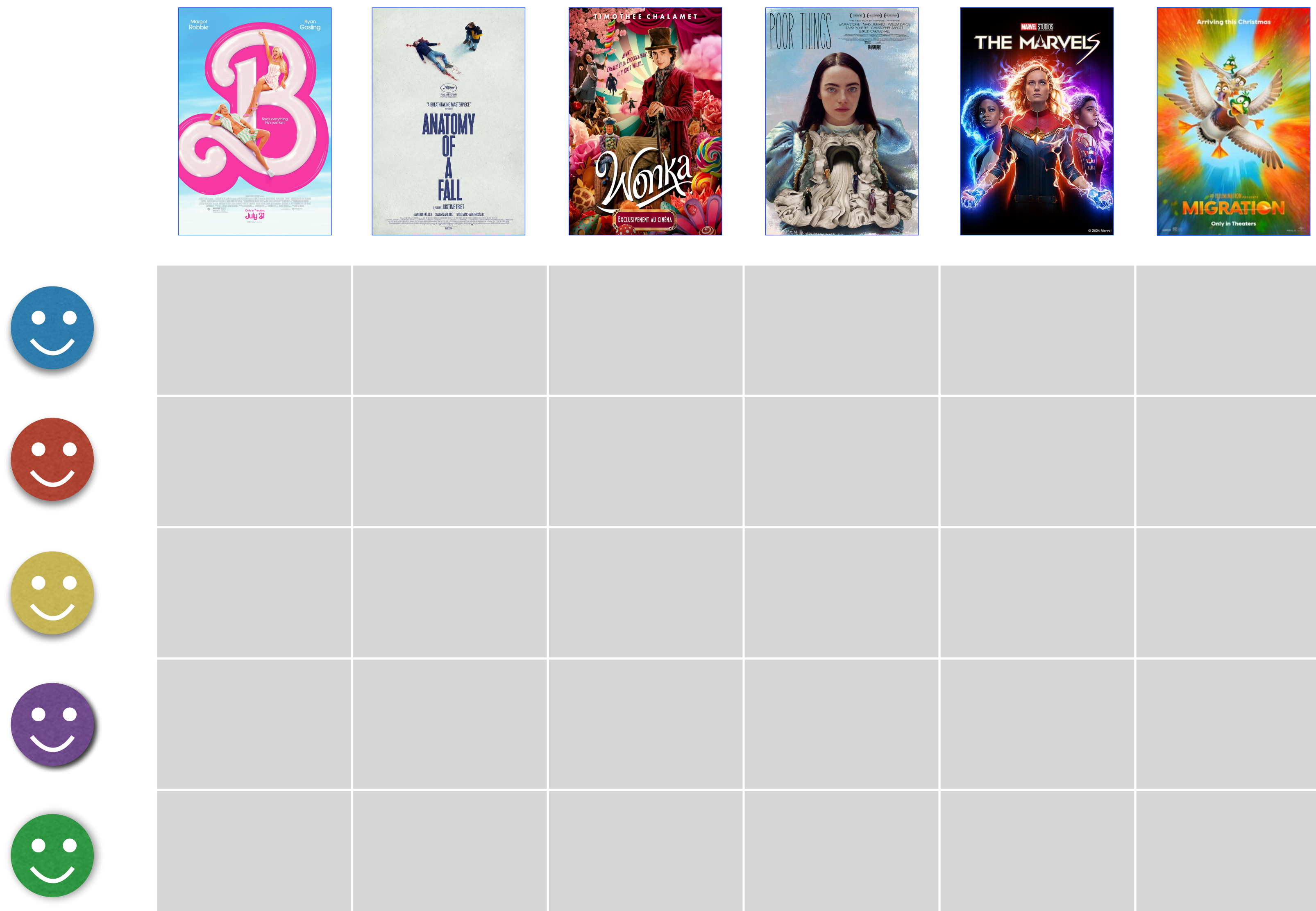
Implicit feedback

- Implicit feedback is collected by the system without the intervention of the user.
- Challenges:
  - Implicit feedback is inherently noisy.
    - Every *positive item*, i.e., item the user has interacted with, is not necessarily an item liked by the user.

Implicit feedback

- Implicit feedback is collected by the system without the intervention of the user.
- Challenges:
  - Implicit feedback is inherently noisy.
    - Every *positive item*, i.e., item the user has interacted with, is not necessarily an item liked by the user.
  - There is no negative feedback.
    - Negative items are a mixture of unliked items and unknown items.

Feedback matrix



Feedback matrix - Explicit feedback

4		5	1		
			5	4	3
	3	1	5		
2	2			5	4
	5		2		

Feedback matrix - Implicit feedback (Binary)

	1		1	1		
				1	1	1
		1	1	1		
	1	1			1	1
		1		1		

Feedback matrix - Implicit feedback (Weights)

2		2.8	0.8		
			3.2	2.4	1
	1.2	1	2		
0.4	0.9			2.1	1.3
	3		0		

Challenges

Rating data sparsity

- Users only interact with a small number items selected from a very large catalog of available items.

Rating data sparsity

- Users only interact with a small number items selected from a very large catalog of available items.

Examples of datasets:

NETFLIX

480k users
18k movies
100M ratings
Period of ~7 years

Density: **1.1%**

James Bennett and Stan Lanning. *The Netflix Prize.* In Proc. KDD Cup and Workshop, 2007.

ACCOR HOTELS
Feel Welcome

7.8M users
4.5k hotels
34.7M bookings
Period of ~4 years

Density: **0.1%**

Marie Al-Ghossein et al. *Exploiting Contextual and External Data for Hotel Recommendation.* In Proc. UMAP, 2018.

amazon

20.98M users
9.35M items
82.83M ratings
Period of ~18 years

Density: **0.00004%**

https://cseweb.ucsd.edu/~jmcauley/datasets.html#amazon_reviews

Cold-start

- The *cold-start* problem designates the problems of:
  - Recommending items to new users where the historical data is missing,
    - Recommendations cannot be computed before the user has rated a certain number of items, but the user expects to receive suggestions before that.
  - Recommending new items to existing users.

Cold-start

- The *cold-start* problem designates the problems of:
  - Recommending items to new users where the historical data is missing,
    - Recommendations cannot be computed before the user has rated a certain number of items, but the user expects to receive suggestions before that.
  - Recommending new items to existing users.
- A related problem is the *gray sheep* problem:
  - Recommending items to users who have very unique tastes, such that no other user (or very few users) has interacted with the same items.

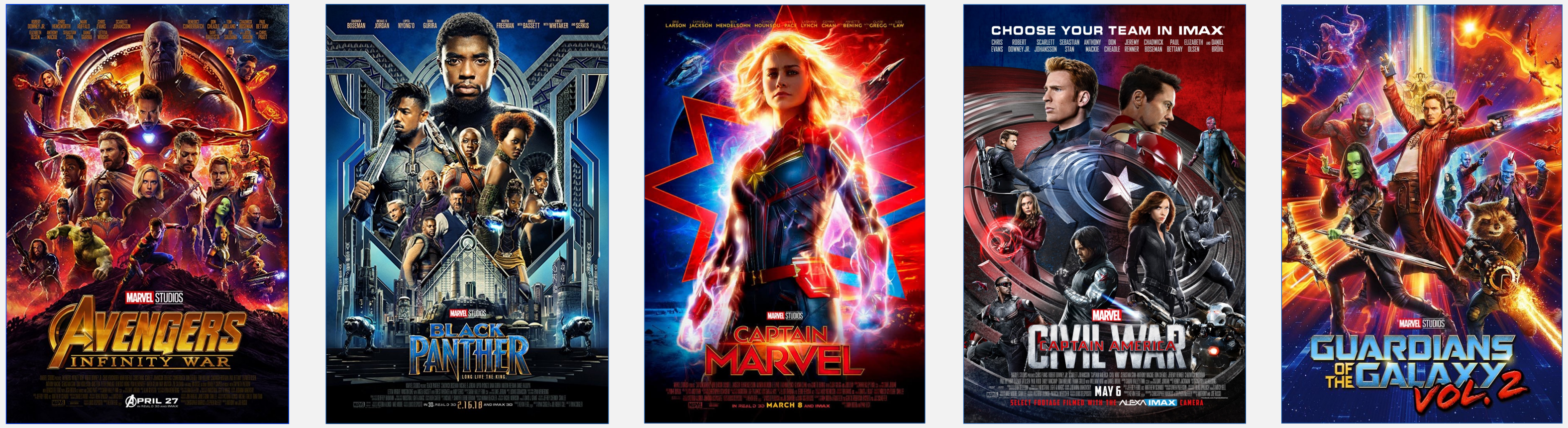
Overspecialization

- Recommendation algorithms tend to recommend items that are too similar to what the user has already experienced, resulting in overspecialized recommendations.

Overspecialization

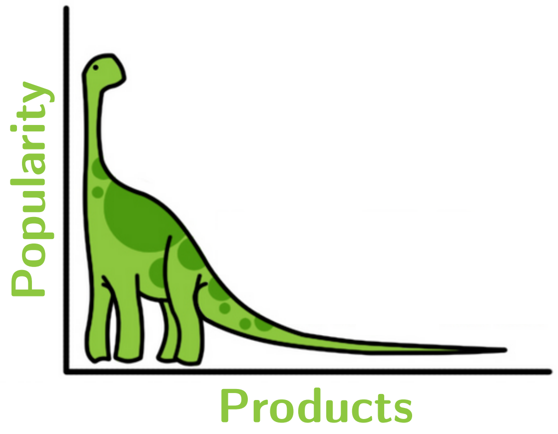
- Recommendation algorithms tend to recommend items that are too similar to what the user has already experienced, resulting in overspecialized recommendations.

Is this recommendation list useful?



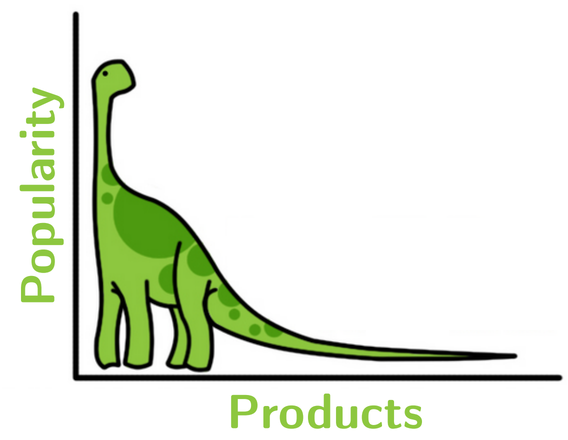
Popularity bias

- Most item catalogs exhibit the *long tail* effect,



Popularity bias

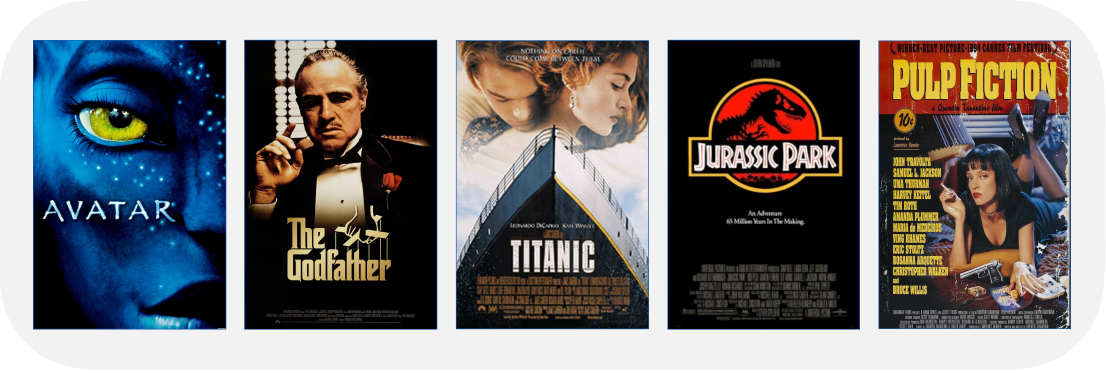
- Most item catalogs exhibit the *long tail* effect,



which raises two issues:

- Risk of recommending popular items to everyone,

Is this recommendation list useful?



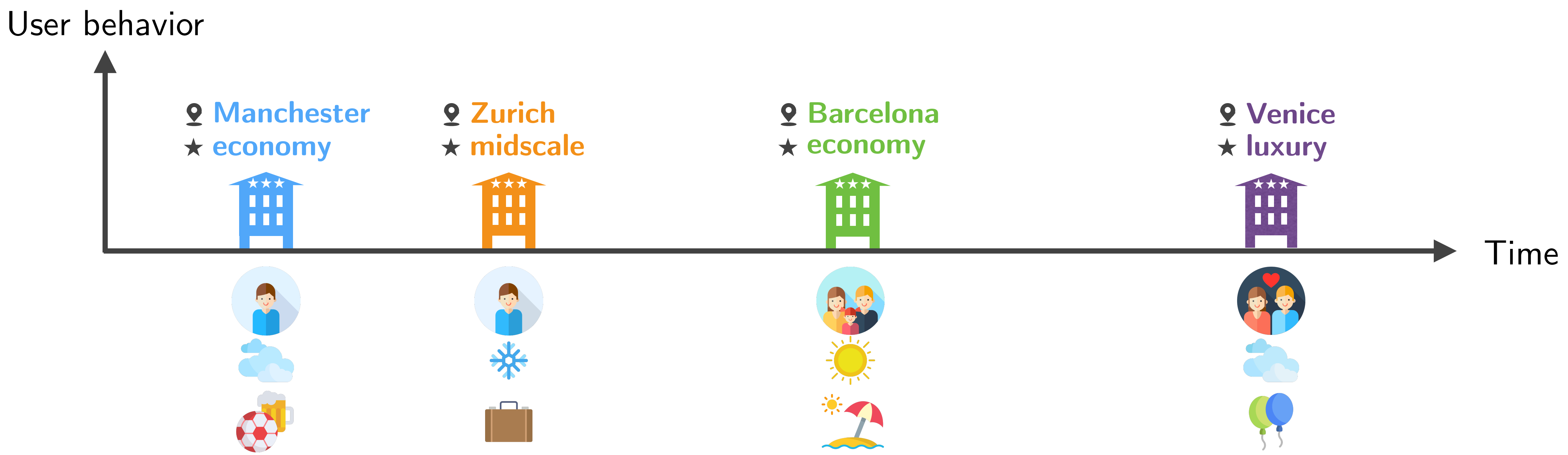
- Difficulty to promote long tail items with little feedback.

Temporal dynamics

- Users preferences and item perceptions are changing over time
  - Reasons: emergence of new items, seasonality factors, etc.

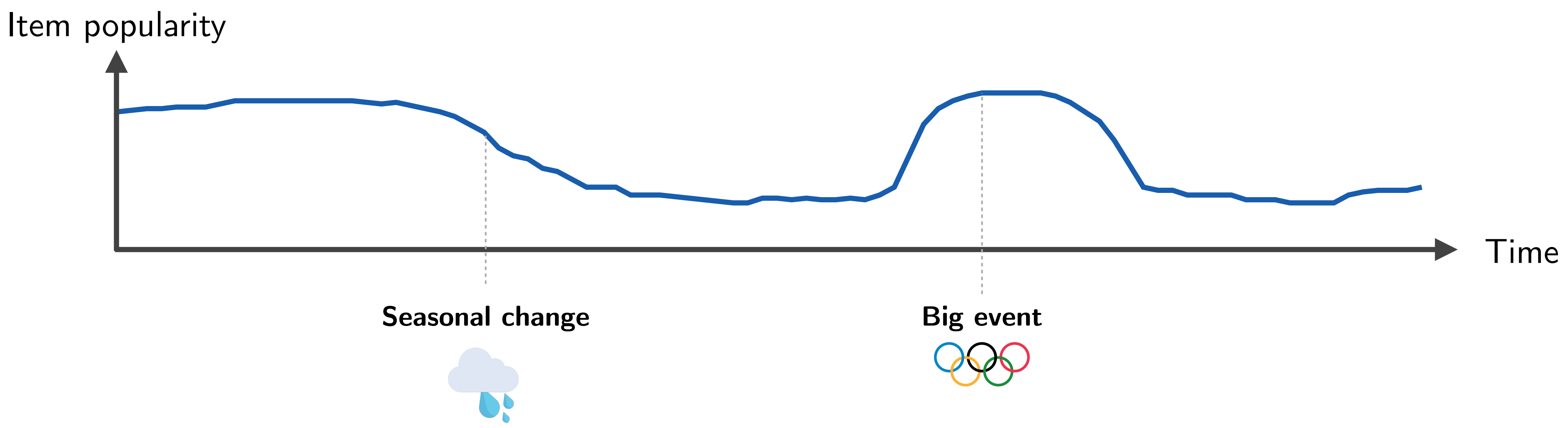
Temporal dynamics

- Users preferences and item perceptions are changing over time
  - Reasons: emergence of new items, seasonality factors, etc.



Temporal dynamics

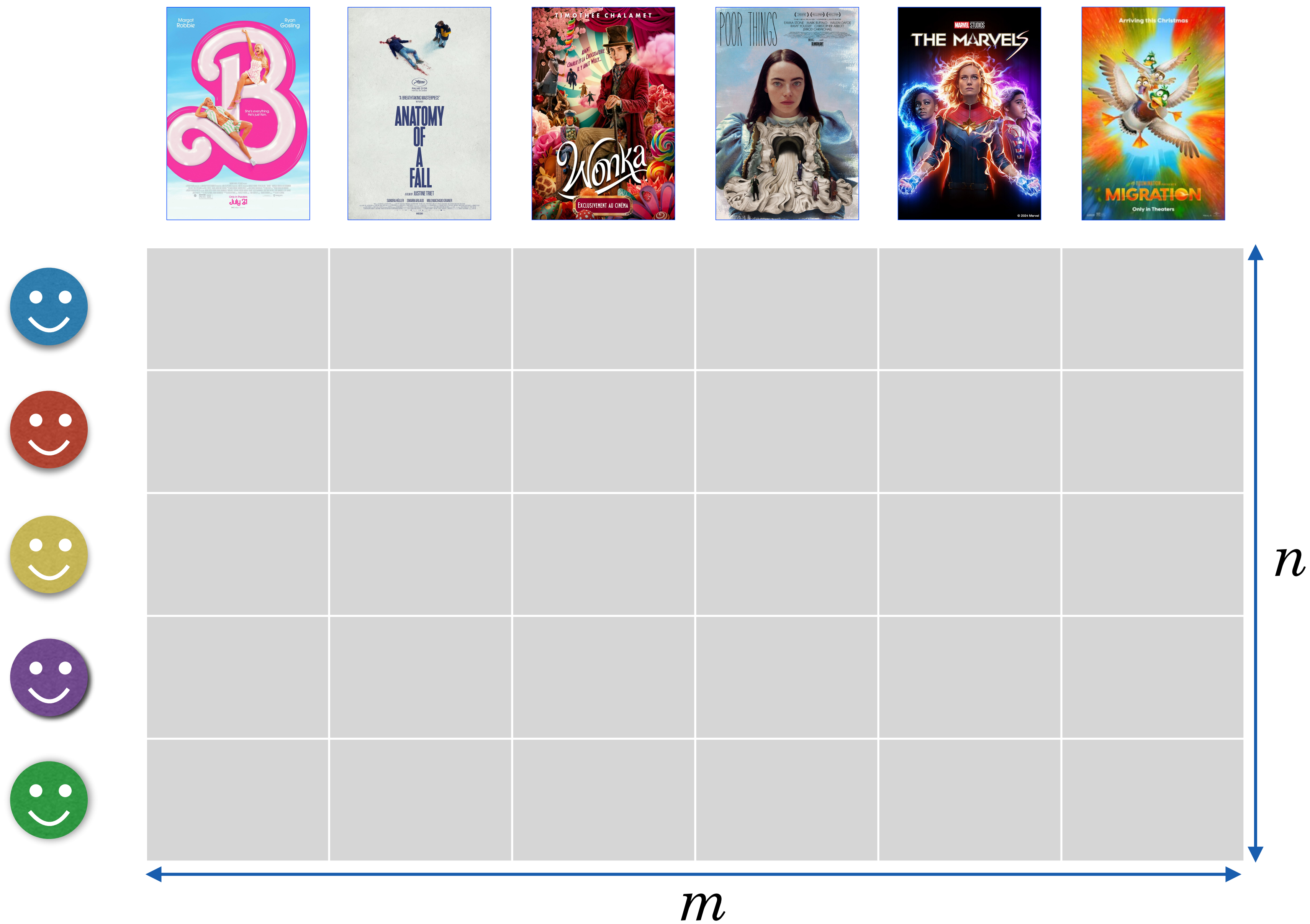
- Users preferences and item perceptions are changing over time
  - Reasons: emergence of new items, seasonality factors, etc.



Feedback matrix - Notations

Set of items: $\mathcal{I}$, set of items rated by user u: $\mathcal{I}_u$

Set of users: $\mathcal{U}$



Feedback matrix: R

Rating: r_{ui}, predicted rating: $\hat{r}_{ui}$

Evaluation of RS

Evaluation methods

- Offline evaluation
  - Performed on previously collected datasets and simulate the user behavior

Evaluation methods

- Offline evaluation
  - Performed on previously collected datasets and simulate the user behavior
  - Pros: Does not require an interaction with real users, allows replicability,

Evaluation methods

- Offline evaluation
  - Performed on previously collected datasets and simulate the user behavior
  - Pros: Does not require an interaction with real users, allows replicability,
  - Cons: Cannot measure the influence of the RS on the user behavior

Evaluation methods

- Offline evaluation
  - Performed on previously collected datasets and simulate the user behavior
  - Pros: Does not require an interaction with real users, allows replicability,
  - Cons: Cannot measure the influence of the RS on the user behavior
- User study
  - Recruit a small number of users and ask them to interact with the RS

Evaluation methods

- Offline evaluation
  - Performed on previously collected datasets and simulate the user behavior
  - Pros: Does not require an interaction with real users, allows replicability,
  - Cons: Cannot measure the influence of the RS on the user behavior
- User study
  - Recruit a small number of users and ask them to interact with the RS
  - Pros: Provides more insights than offline evaluation,

Evaluation methods

- Offline evaluation
  - Performed on previously collected datasets and simulate the user behavior
  - Pros: Does not require an interaction with real users, allows replicability,
  - Cons: Cannot measure the influence of the RS on the user behavior
- User study
  - Recruit a small number of users and ask them to interact with the RS
  - Pros: Provides more insights than offline evaluation,
  - Cons: May introduce biases in the collected data

Evaluation methods

- Offline evaluation
  - Performed on previously collected datasets and simulate the user behavior
  - Pros: Does not require an interaction with real users, allows replicability,
  - Cons: Cannot measure the influence of the RS on the user behavior
- User study
  - Recruit a small number of users and ask them to interact with the RS
  - Pros: Provides more insights than offline evaluation,
  - Cons: May introduce biases in the collected data
- Online evaluation
  - A/B testing on a real RS deployed online

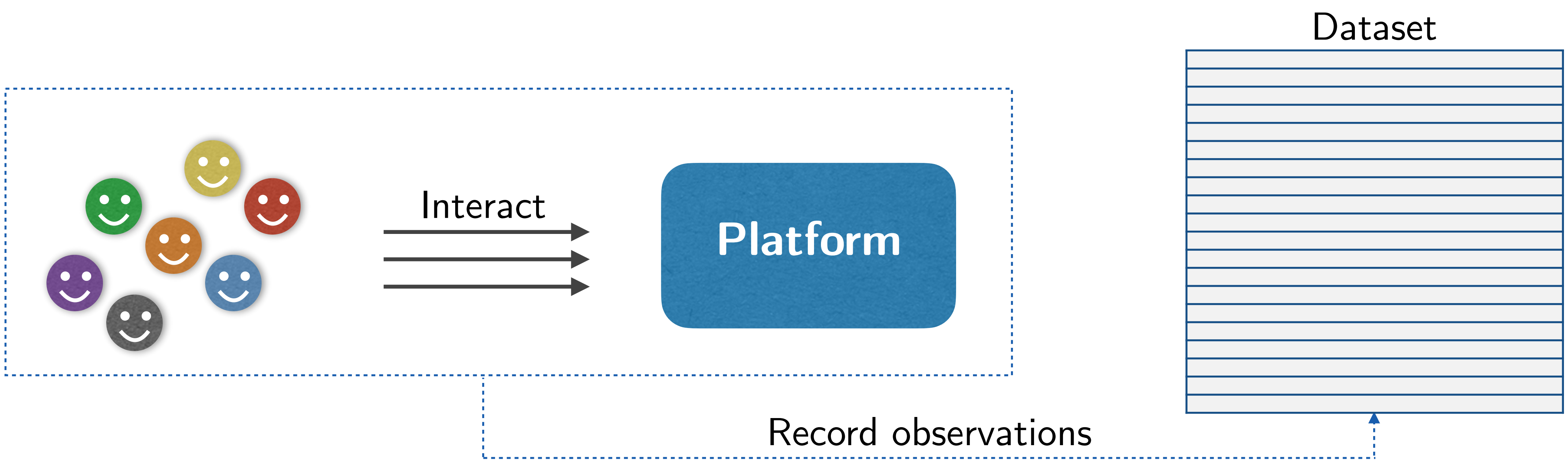
Evaluation methods

- Offline evaluation
  - Performed on previously collected datasets and simulate the user behavior
  - Pros: Does not require an interaction with real users, allows replicability,
  - Cons: Cannot measure the influence of the RS on the user behavior
- User study
  - Recruit a small number of users and ask them to interact with the RS
  - Pros: Provides more insights than offline evaluation,
  - Cons: May introduce biases in the collected data
- Online evaluation
  - A/B testing on a real RS deployed online
  - Pros: Provides the strongest evidence,

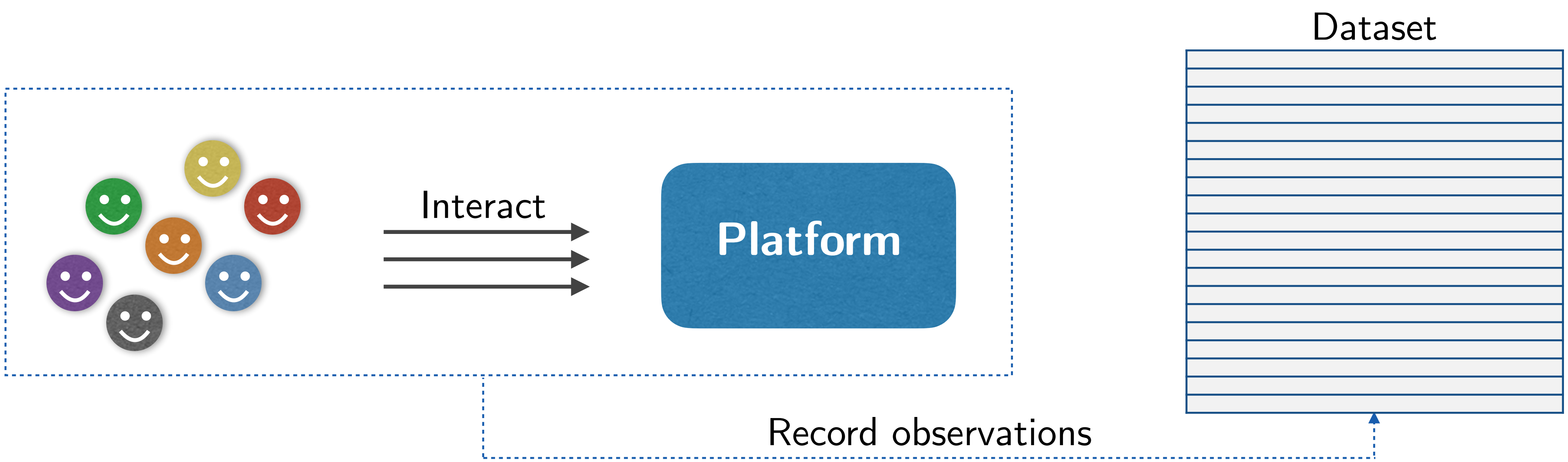
Evaluation methods

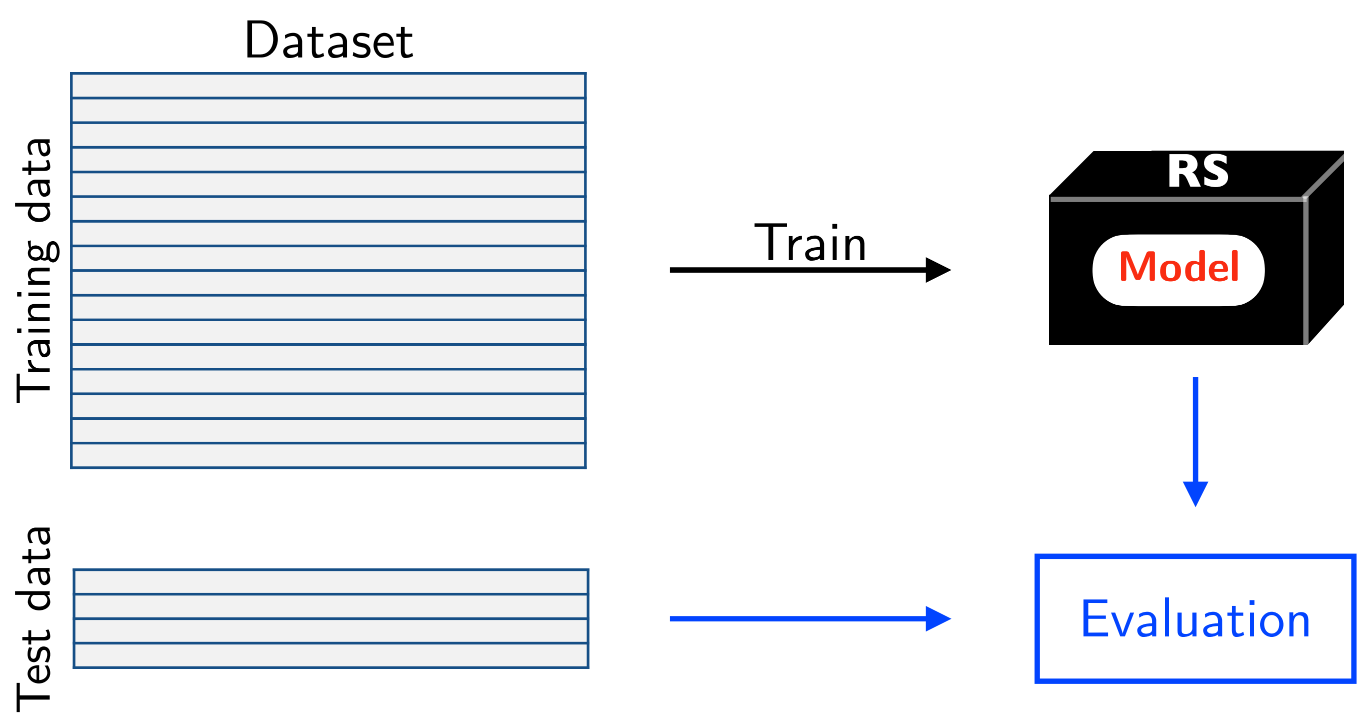
- Offline evaluation
  - Performed on previously collected datasets and simulate the user behavior
  - Pros: Does not require an interaction with real users, allows replicability,
  - Cons: Cannot measure the influence of the RS on the user behavior
- User study
  - Recruit a small number of users and ask them to interact with the RS
  - Pros: Provides more insights than offline evaluation,
  - Cons: May introduce biases in the collected data
- Online evaluation
  - A/B testing on a real RS deployed online
  - Pros: Provides the strongest evidence,
  - Cons: Risk of negatively affecting the real users's experience

Offline evaluation



Offline evaluation



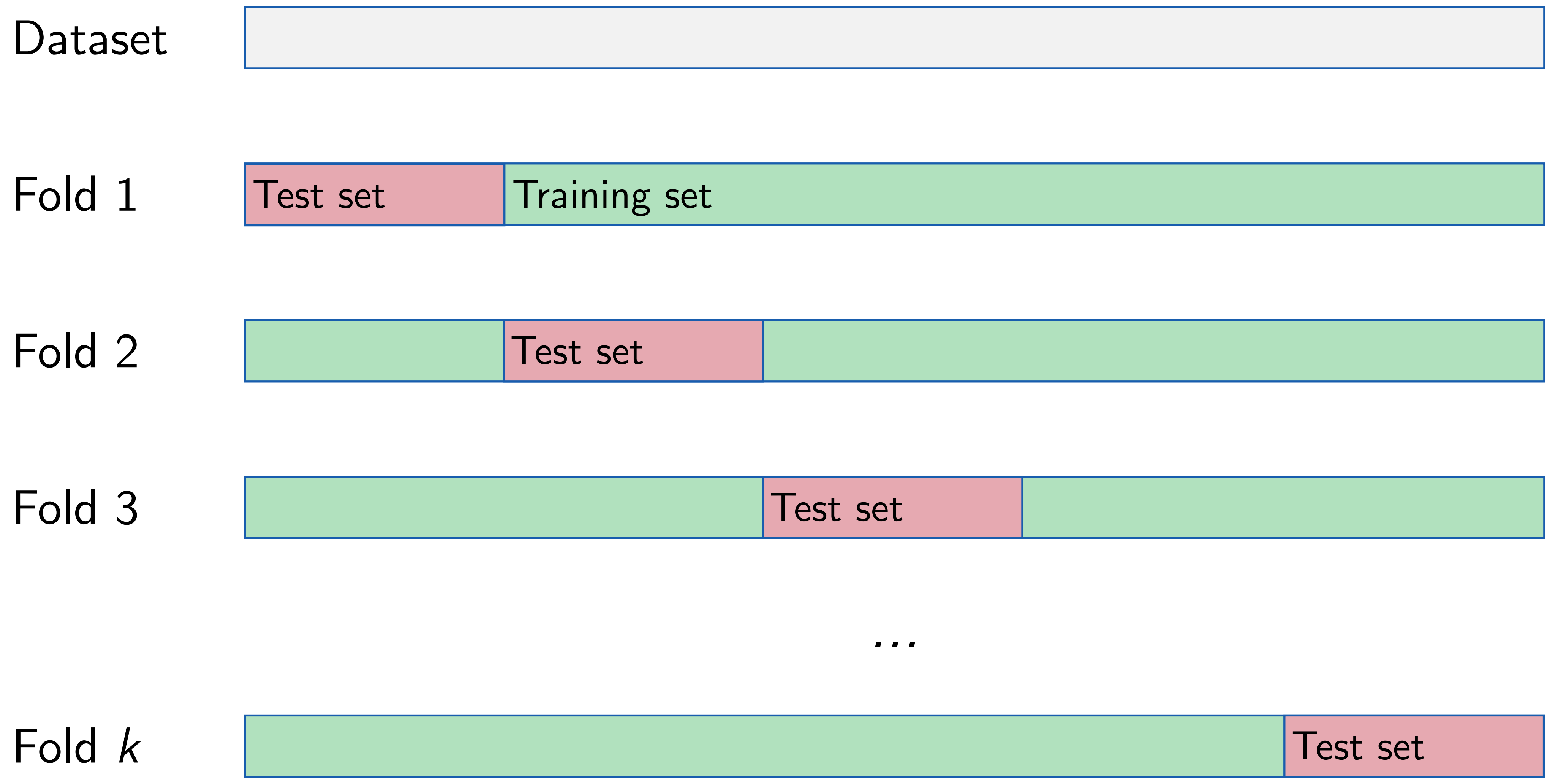


Offline evaluation

- Random split

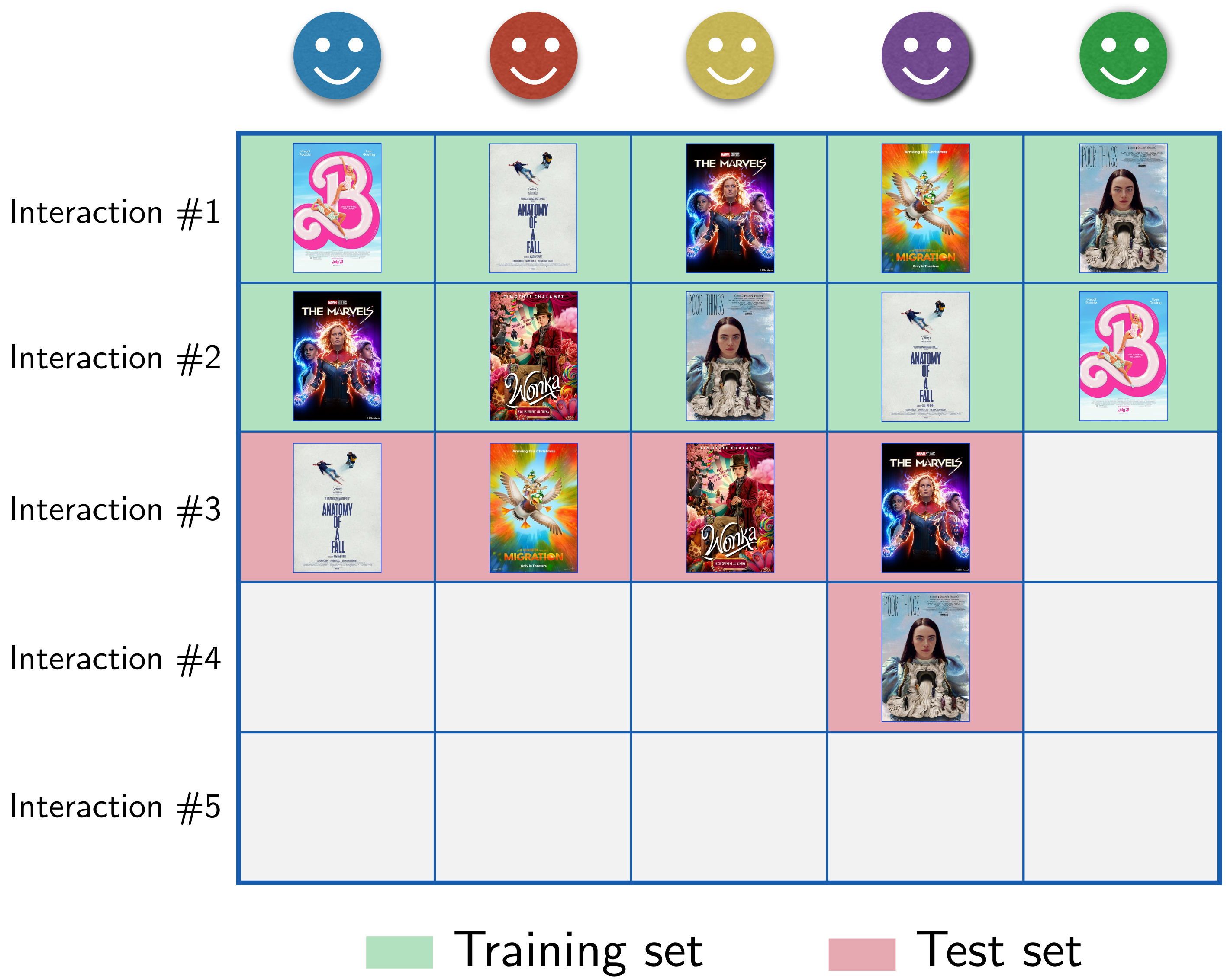
Offline evaluation

- Random split, k-fold cross validation



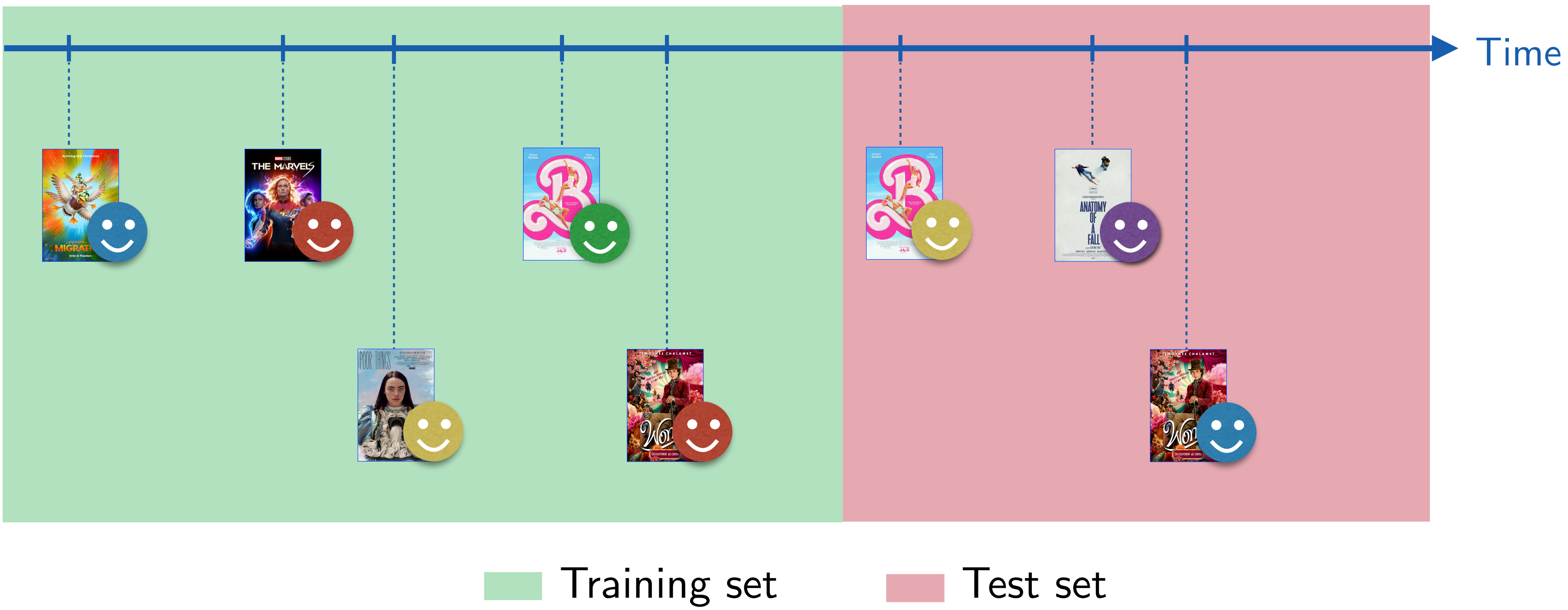
Offline evaluation

- Given-*n* split



Offline evaluation

- Chronological split



Evaluation criteria

- Accuracy
    - How accurate are the predictions of utility of items?

Evaluation criteria

- Accuracy
  - How accurate are the predictions of utility of items?
- Coverage
  - How many items from the catalog are the recommendations covering?

Evaluation criteria

- Accuracy
  - How accurate are the predictions of utility of items?
- Coverage
  - How many items from the catalog are the recommendations covering?
- Diversity
  - How diverse are the computed recommendations?

Evaluation criteria

- Accuracy
  - How accurate are the predictions of utility of items?
- Coverage
  - How many items from the catalog are the recommendations covering?
- Diversity
  - How diverse are the computed recommendations?
- Serendipity
  - How surprising are the relevant computed recommendations?

Evaluation criteria

- Accuracy
  - How accurate are the predictions of utility of items?
- Coverage
  - How many items from the catalog are the recommendations covering?
- Diversity
  - How diverse are the computed recommendations?
- Serendipity
  - How surprising are the relevant computed recommendations?
- Explainability
  - Are we able to explain and justify the recommendations?

Evaluation metrics for accuracy

Measuring the accuracy of the predicted ratings.

- Mean Absolute Error (MAE)

$$MAE = \frac{1}{|\mathcal{T}|} \sum_{(u,i)\in\mathcal{T}} |r_{ui} - \hat{r}_{ui}|$$

Evaluation metrics for accuracy

Measuring the accuracy of the predicted ratings.

- Mean Absolute Error (MAE)

$$MAE = \frac{1}{|\mathcal{T}|} \sum_{(u,i)\in\mathcal{T}} |r_{ui} - \hat{r}_{ui}|$$

$$MAE = 1.4$$

Evaluation metrics for accuracy

Measuring the accuracy of the predicted ratings.

- Root Mean Squarred Error (RMSE)

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{|\mathcal{T}|} \sum_{(u,i) \in \mathcal{T}} (r_{ui} - \hat{r}_{ui})^2}$$

Evaluation metrics for accuracy

Measuring the accuracy of the predicted ratings.

- Root Mean Squarred Error (RMSE)

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{|\mathcal{T}|} \sum_{(u,i)\in\mathcal{T}} (r_{ui} - \hat{r}_{ui})^2}$$

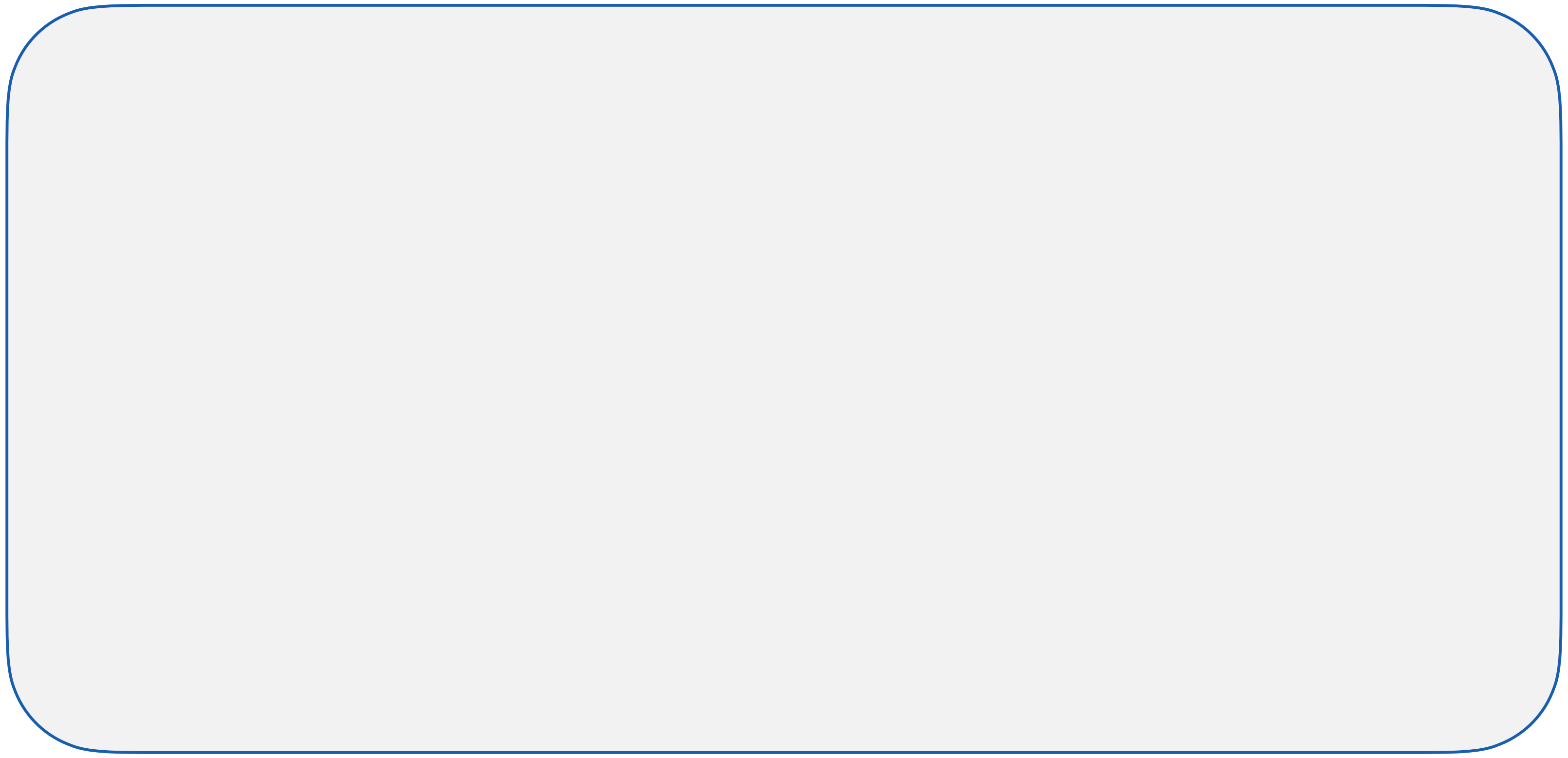
$$RMSE = 3$$

Evaluation metrics for accuracy

Measuring the accuracy of top-N recommendations

Evaluation metrics for accuracy

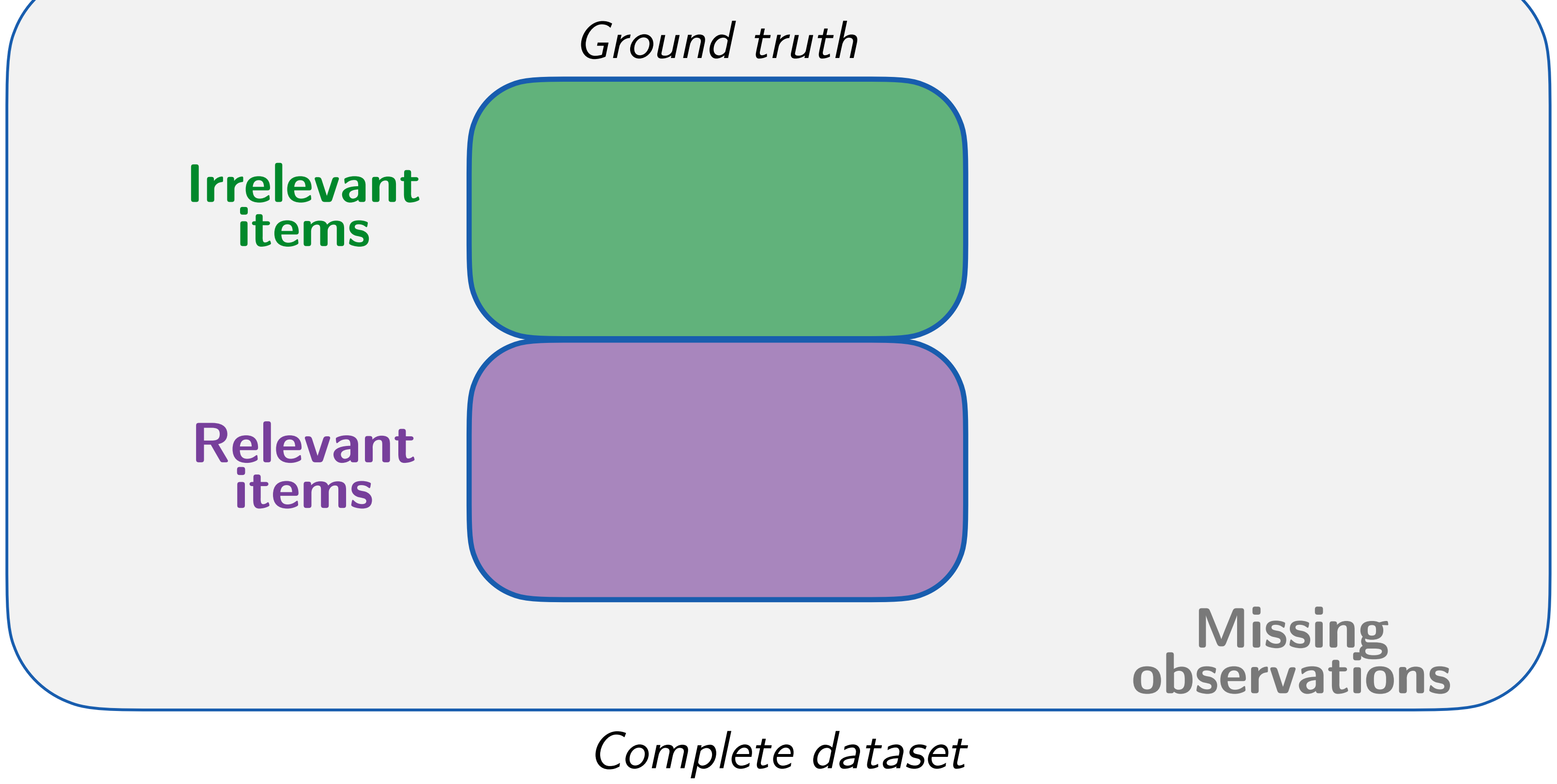
Measuring the accuracy of top-N recommendations



Complete dataset

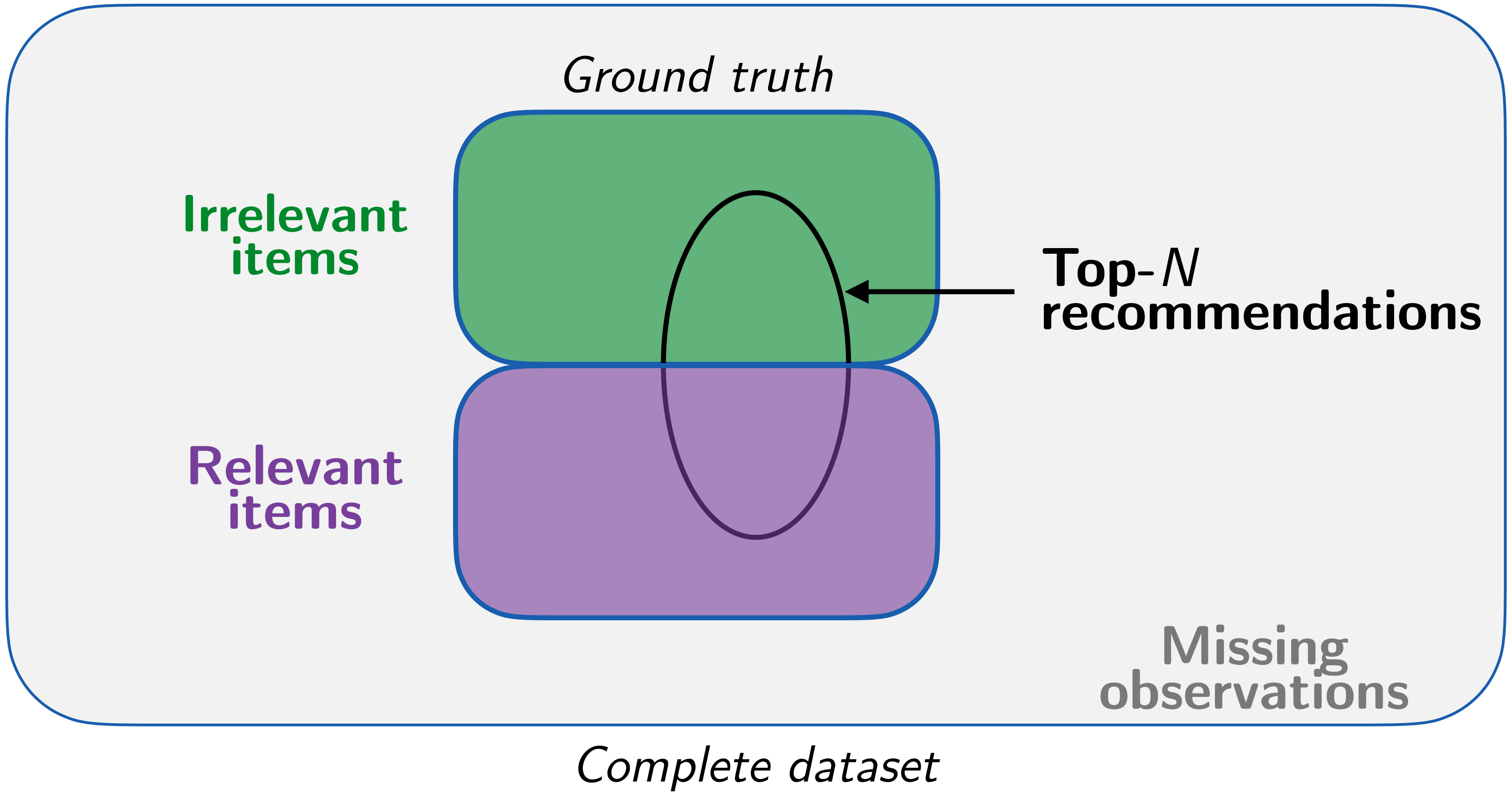
Evaluation metrics for accuracy

Measuring the accuracy of top-N recommendations



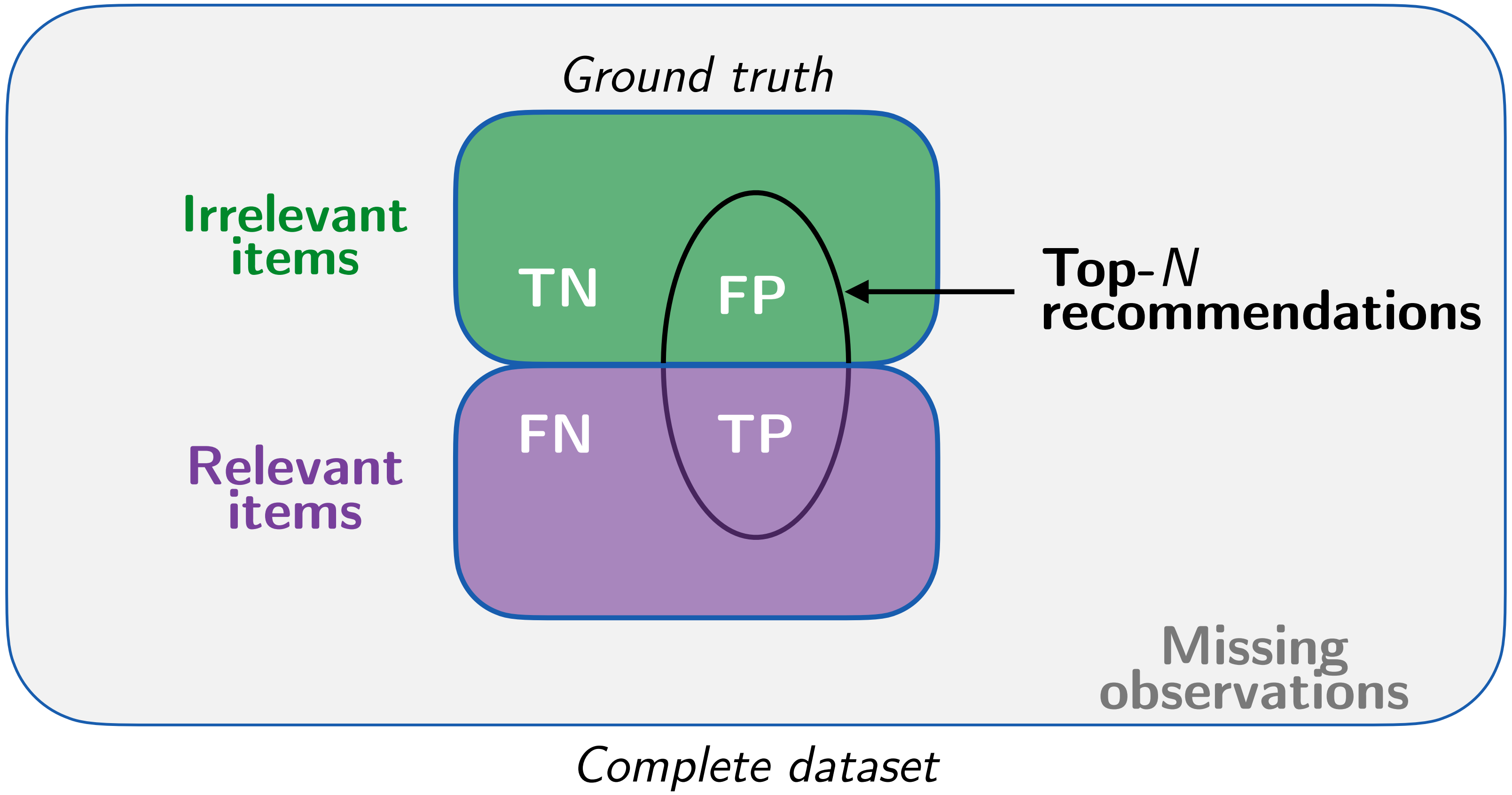
Evaluation metrics for accuracy

Measuring the accuracy of top-N recommendations



Evaluation metrics for accuracy

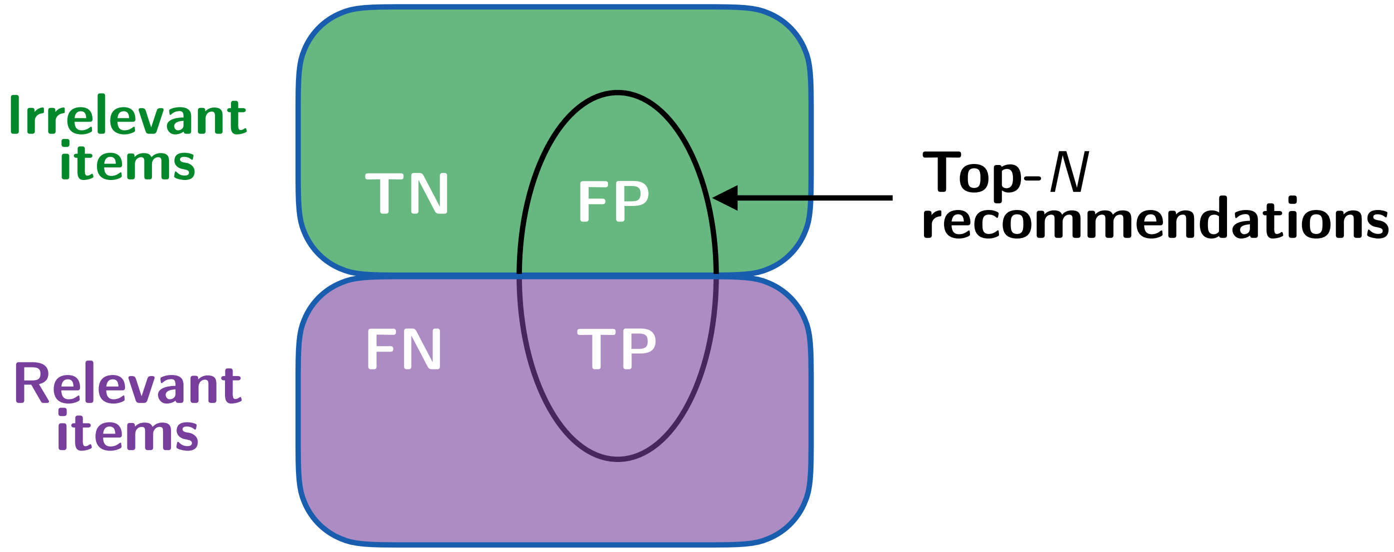
Measuring the accuracy of top-N recommendations



Evaluation metrics for accuracy

Measuring the accuracy of top-N recommendations

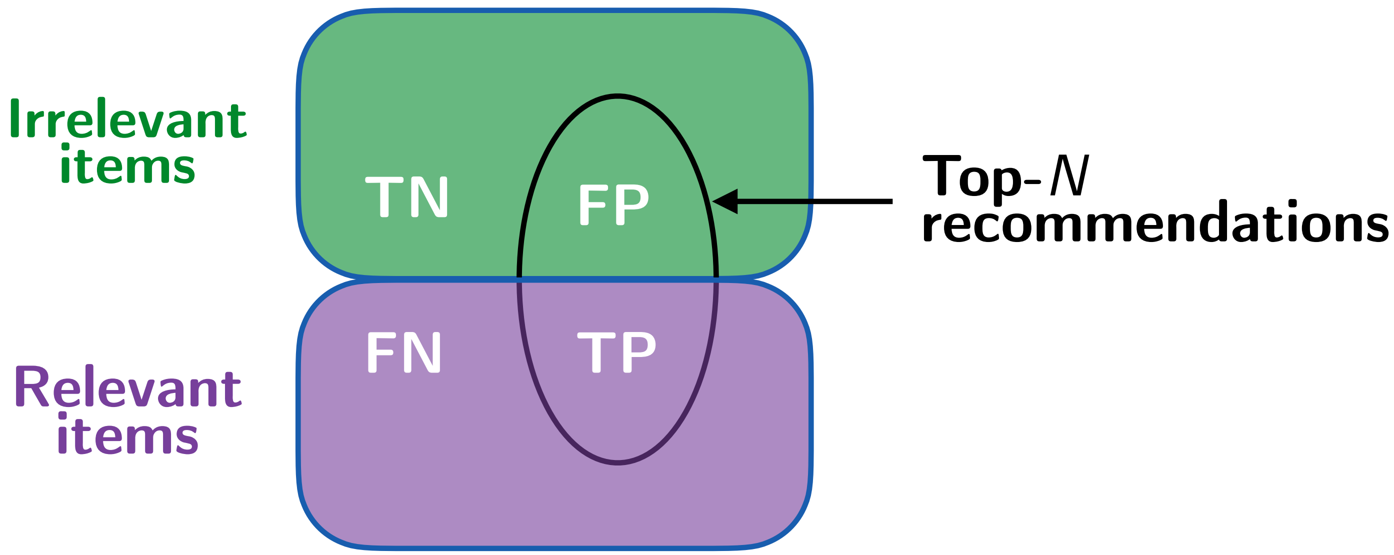
$$Recall_u@N = \frac{TP}{FN + TP} = \frac{\#relevant\ recommended\ items}{\#relevant\ items}$$



Evaluation metrics for accuracy

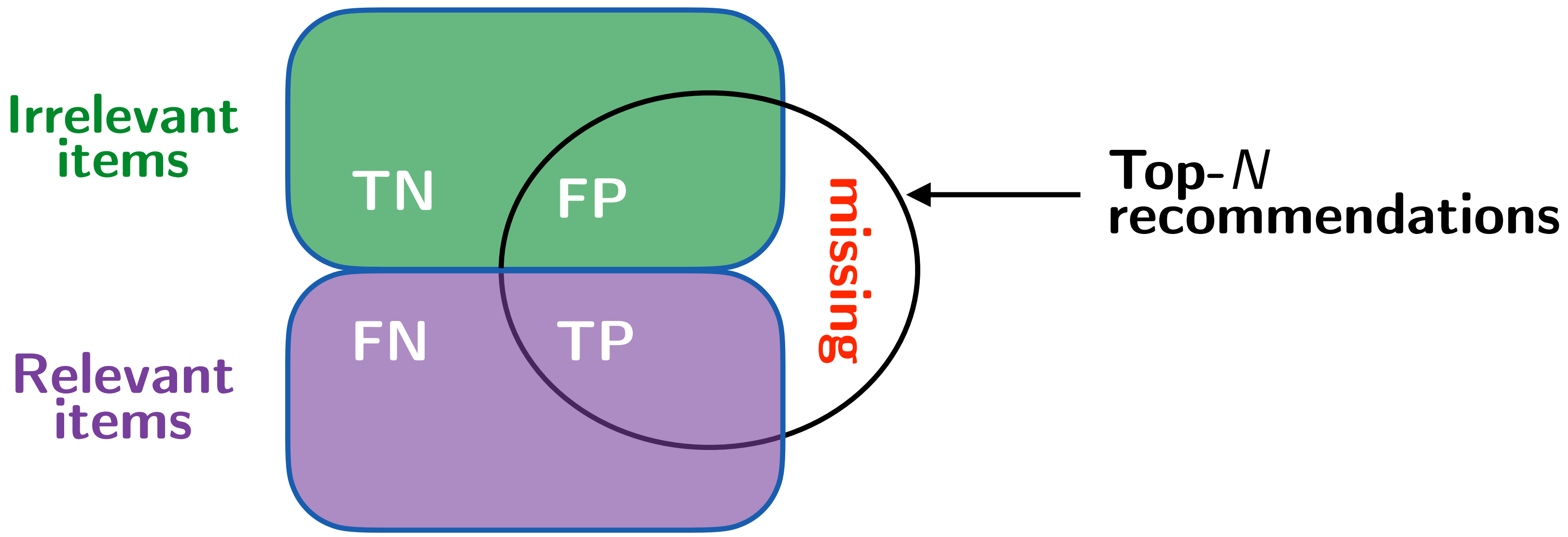
Measuring the accuracy of top-N recommendations

$$Precision_u@N = \frac{TP}{FP + TP} = \frac{\#relevant\ recommended\ items}{\#recommended\ items}$$



Evaluation metrics for accuracy

- Recall: *Missing Not As Random* assumption
  - Non-relevant items are more likely to be missing than relevant items
- Precision: *All Missing As Negative* assumption
  - All missing ratings are irrelevant



Evaluation metrics for ranking

Measuring the quality of ranking of top-N recommendations

- Discounted Cumulative Gain (DCG):

$$DCG_u@N = \sum_{i=1}^{N} \frac{rel_{ui}}{log_2(i+1)}$$

where rel_{ui} = *1* if item at rank i is relevant, *0* otherwise.

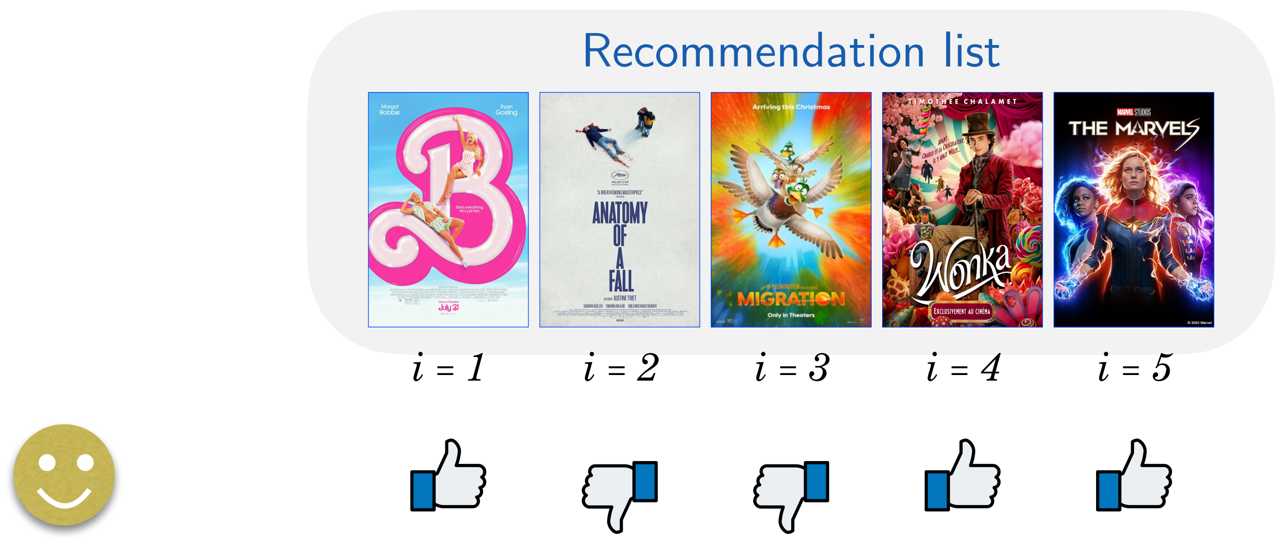
Evaluation metrics for ranking

Measuring the quality of ranking of top-N recommendations

- Discounted Cumulative Gain (DCG):

$$DCG_u@N = \sum_{i=1}^{N} \frac{rel_{ui}}{log_2(i+1)}$$

where rel_{ui} = *1* if item at rank i is relevant, *0* otherwise.



DCG_u $1 + 0 + 0 + 0.43 + 0.38$

Evaluation metrics for ranking

Measuring the quality of ranking of top-N recommendations

- Normalized Discounted Cumulative Gain (DCG):

$$NDCG@N = \frac{1}{|\mathcal{T}_u|} \sum_{u \in \mathcal{T}_u} \frac{DCG_u@N}{DCG_u^*@N}$$

where $DCG^*{}_u$ is the best possible DCG_u that can be obtained.

Evaluation metrics for ranking

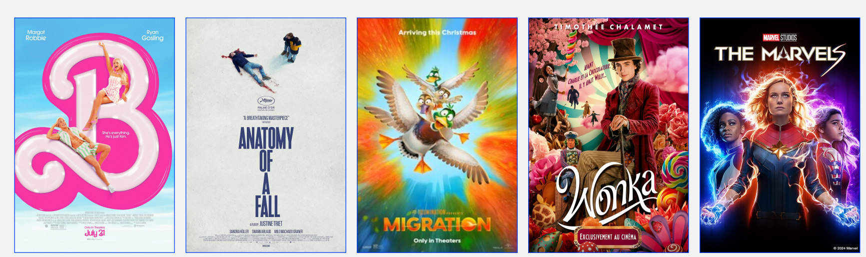
Measuring the quality of ranking of top-N recommendations

- Normalized Discounted Cumulative Gain (DCG):

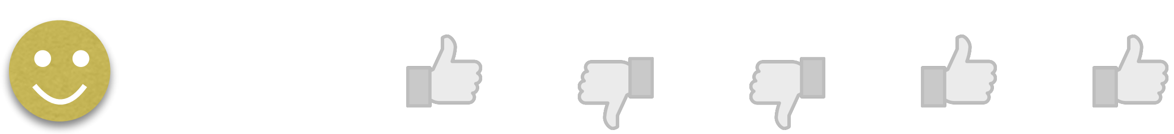
$$NDCG@N = \frac{1}{|\mathcal{T}_u|} \sum_{u \in \mathcal{T}_u} \frac{DCG_u@N}{DCG_u^*@N}$$

where $DCG^*{}_u$ is the best possible DCG_u that can be obtained.

Recommendation list



i = 1 i = 2 i = 3 i = 4 i = 5



$DCG^*{}_u$ $\quad 1 \; + 0.63 + \; 0.5 \; + 0.43 + 0.38$

List of other used metrics

- Mean Average Precision (MAP)
- Mean Reciprocal Rank (MRR)
- Area under the ROC Curve (AUC)
- Hit-Rank (HR)
- Average Reciprocal Hit-Rank (ARHR)

...

Non-personalized recommendations

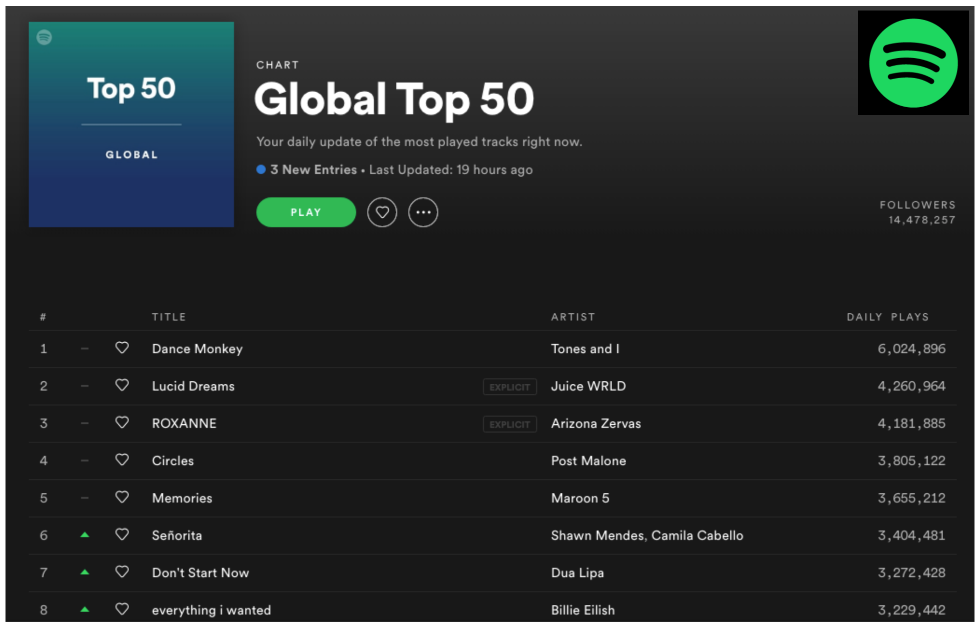
Non-personalized recommendations

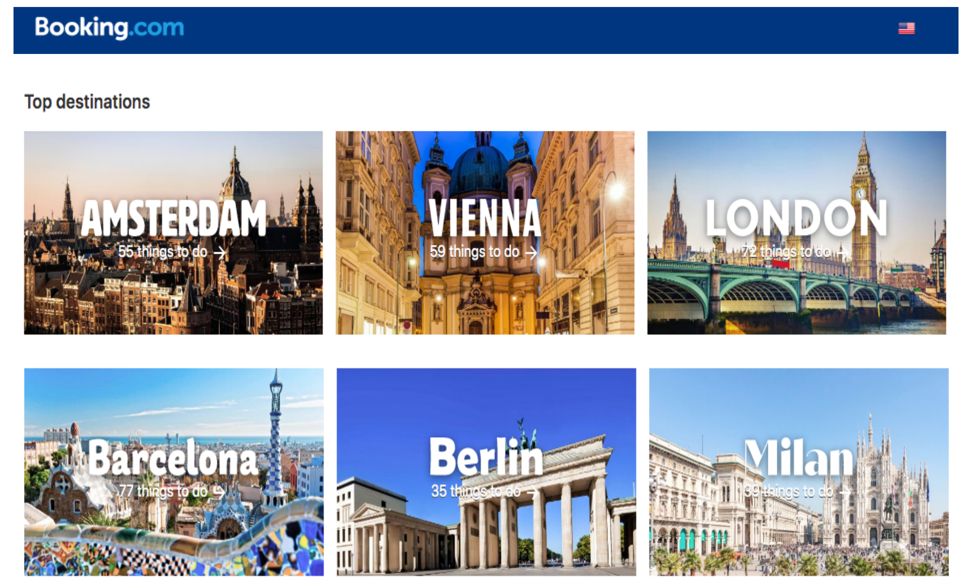
- Personalized recommendations
  - Tailor the recommendation list to a specific individual
  - Using information about the target user and similar users

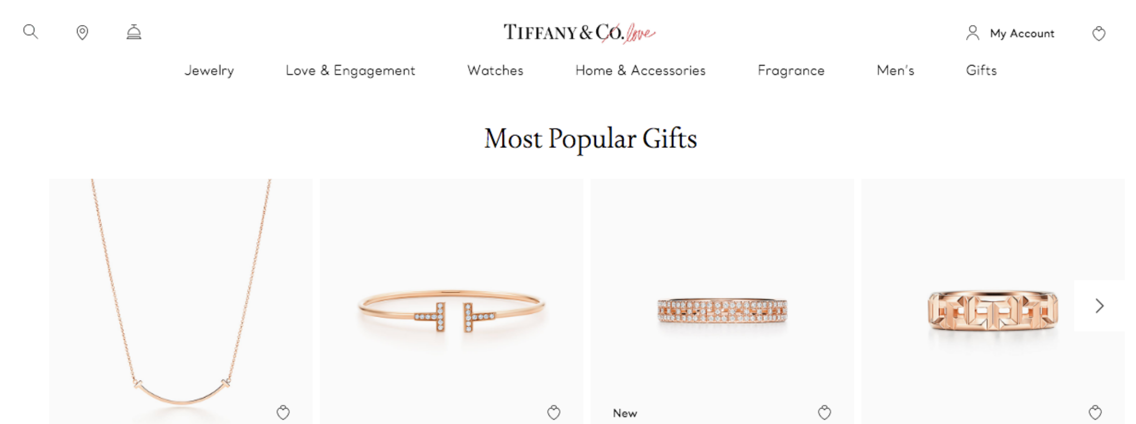
vs.

- Non-personalized recommendations
  - Do not require to know anything about the individual user
  - Using aggregated information about users

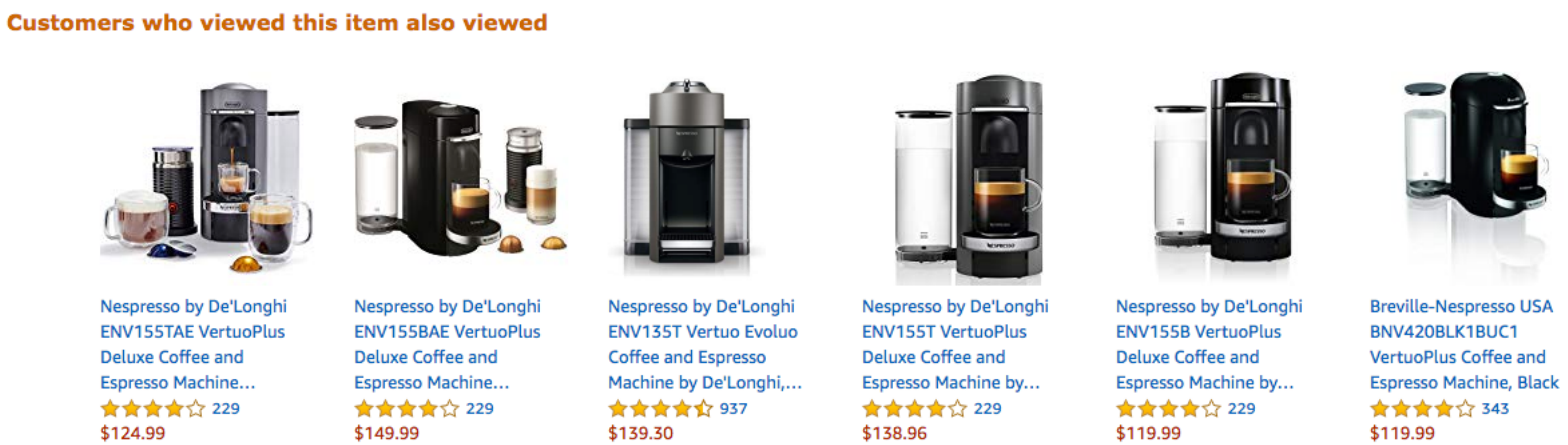
Most popular item recommendation







Association rules

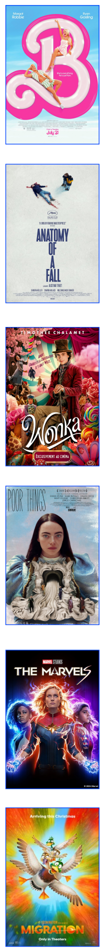


Association rules

1		1	1		
			1	1	1
	1	1	1		
1	1			1	1
	1		1		

Also watched...

Users who watched...



	1	1	1	1	1
1		1	2	1	1
1	1		2	0	0
1	2	2		1	1
1	1	0	1		2
1	1	0	1	2	

Confidence: $c(i \to j) = \dfrac{|\mathcal{U}_i \cap \mathcal{U}_j|}{|\mathcal{U}_i|}$

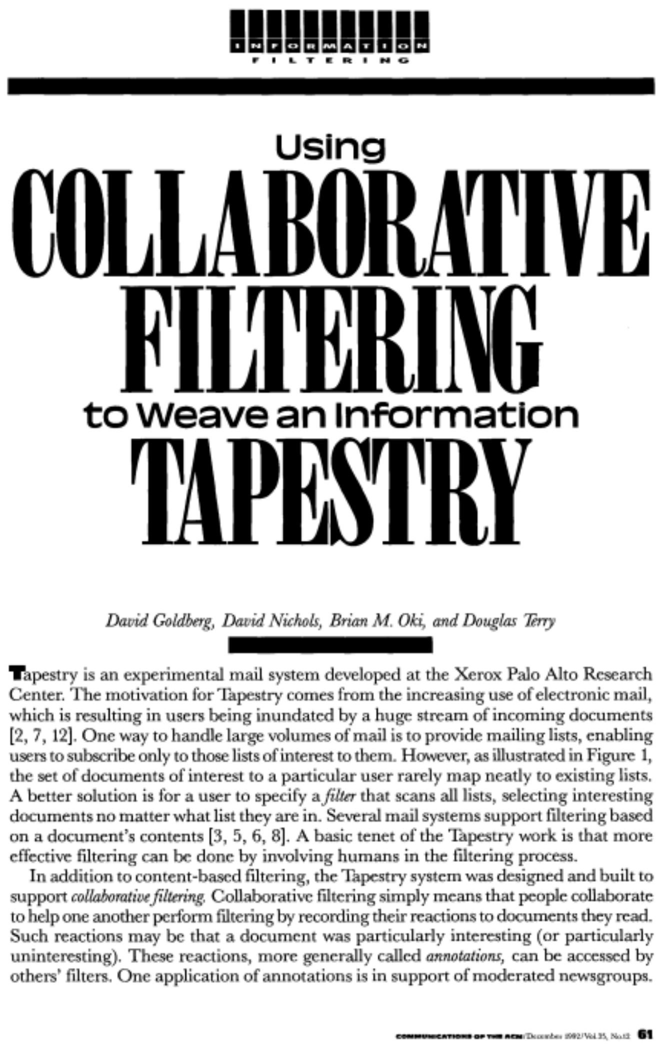
Support: $S(i \to j) = \dfrac{|\mathcal{U}_i \cap \mathcal{U}_j|}{|\mathcal{U}|}$

Towards personalized recommendations...

- Recommendations approaches
  - Collaborative Filtering approaches
  - Content-Based Filtering approaches
  - Hybrid approaches
  - Context-aware approaches

Collaborative Filtering

Origin of collaborative filtering



INFORMATION FILTERING

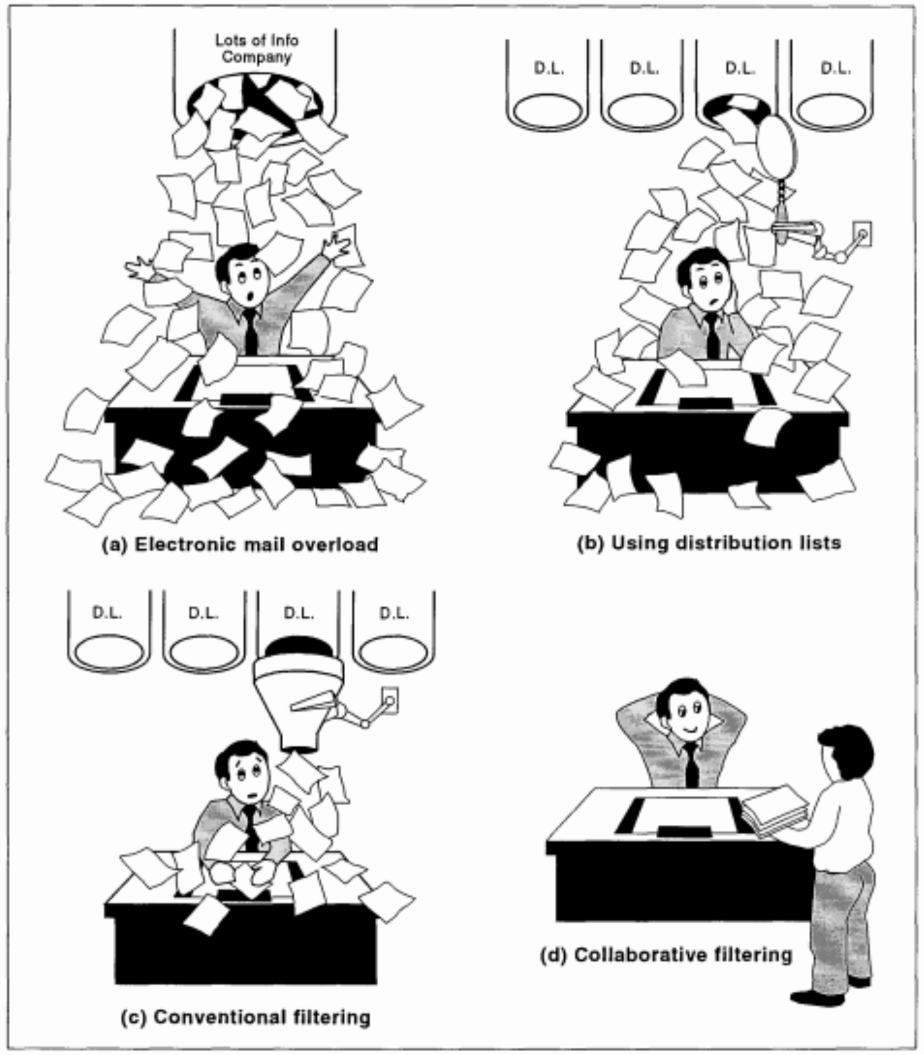
Using
COLLABORATIVE FILTERING
to Weave an Information
TAPESTRY

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

Tapestry is an experimental mail system developed at the Xerox Palo Alto Research Center. The motivation for Tapestry comes from the increasing use of electronic mail, which is resulting in users being inundated by a huge stream of incoming documents [2, 7, 12]. One way to handle large volumes of mail is to provide mailing lists, enabling users to subscribe only to those lists of interest to them. However, as illustrated in Figure 1, the set of documents of interest to a particular user rarely map neatly to existing lists. A better solution is for a user to specify a *filter* that scans all lists, selecting interesting documents no matter what list they are in. Several mail systems support filtering based on a document's contents [3, 5, 6, 8]. A basic tenet of the Tapestry work is that more effective filtering can be done by involving humans in the filtering process.

In addition to content-based filtering, the Tapestry system was designed and built to support *collaborative filtering*. Collaborative filtering simply means that people collaborate to help one another perform filtering by recording their reactions to documents they read. Such reactions may be that a document was particularly interesting (or particularly uninteresting). These reactions, more generally called *annotations*, can be accessed by others' filters. One application of annotations is in support of moderated newsgroups.

COMMUNICATIONS OF THE ACM/December 1992/Vol.35, No.12 61



(a) Electronic mail overload

(b) Using distribution lists

(c) Conventional filtering

(d) Collaborative filtering

Users *collaborate* to help each others *filter* the emails by annotating their usefulness and propagating the information to the others

Collaborative filtering

- General idea: Users who are similar with regards to the history of interactions are considered to be like-minded and are susceptible to share similar preferences in the future.
  - Recommendations are based on the user history and on the past behavior of similar users.

Collaborative filtering

- General idea: Users who are similar with regards to the history of interactions are considered to be like-minded and are susceptible to share similar preferences in the future.
  - Recommendations are based on the user history and on the past behavior of similar users.
  - No additional information about items is required!!

Collaborative filtering

- General idea: Users who are similar with regards to the history of interactions are considered to be like-minded and are susceptible to share similar preferences in the future.
  - Recommendations are based on the user history and on the past behavior of similar users.
  - No additional information about items is required!!

However,

Collaborative filtering

- General idea: Users who are similar with regards to the history of interactions are considered to be like-minded and are susceptible to share similar preferences in the future.
  - Recommendations are based on the user history and on the past behavior of similar users.
  - No additional information about items is required!!

However,

- Requires a large number of user interactions

Collaborative filtering

- General idea: Users who are similar with regards to the history of interactions are considered to be like-minded and are susceptible to share similar preferences in the future.
    - Recommendations are based on the user history and on the past behavior of similar users.
    - No additional information about items is required!!

However,

- Requires a large number of user interactions
- Requires products to be standardized

Collaborative filtering

- General idea: Users who are similar with regards to the history of interactions are considered to be like-minded and are susceptible to share similar preferences in the future.
  - Recommendations are based on the user history and on the past behavior of similar users.
  - No additional information about items is required!!

However,

- Requires a large number of user interactions
- Requires products to be standardized
- Assumes that prior behavior determines current behavior

Categories of collaborative filtering approaches

- Memory-based approaches
  - Directly use the recorded user interactions to compute recommendations
  - Can be *user-based* or *item-based*
  - Example: Neighborhood-based approach

Categories of collaborative filtering approaches

- Memory-based approaches
  - Directly use the recorded user interactions to compute recommendations
  - Can be *user-based* or *item-based*
  - Example: Neighborhood-based approach
- Model-based approaches
  - Learn a predictive model which is then used for recommendation
  - Example: Matrix factorization

Memory-based approaches

- General idea of neighborhood-based approaches:

  Similar users are interested in the same items, or

  Similar items interest the same users,

  assuming that user preferences remain stable over time.

Memory-based approaches

- General idea of neighborhood-based approaches:

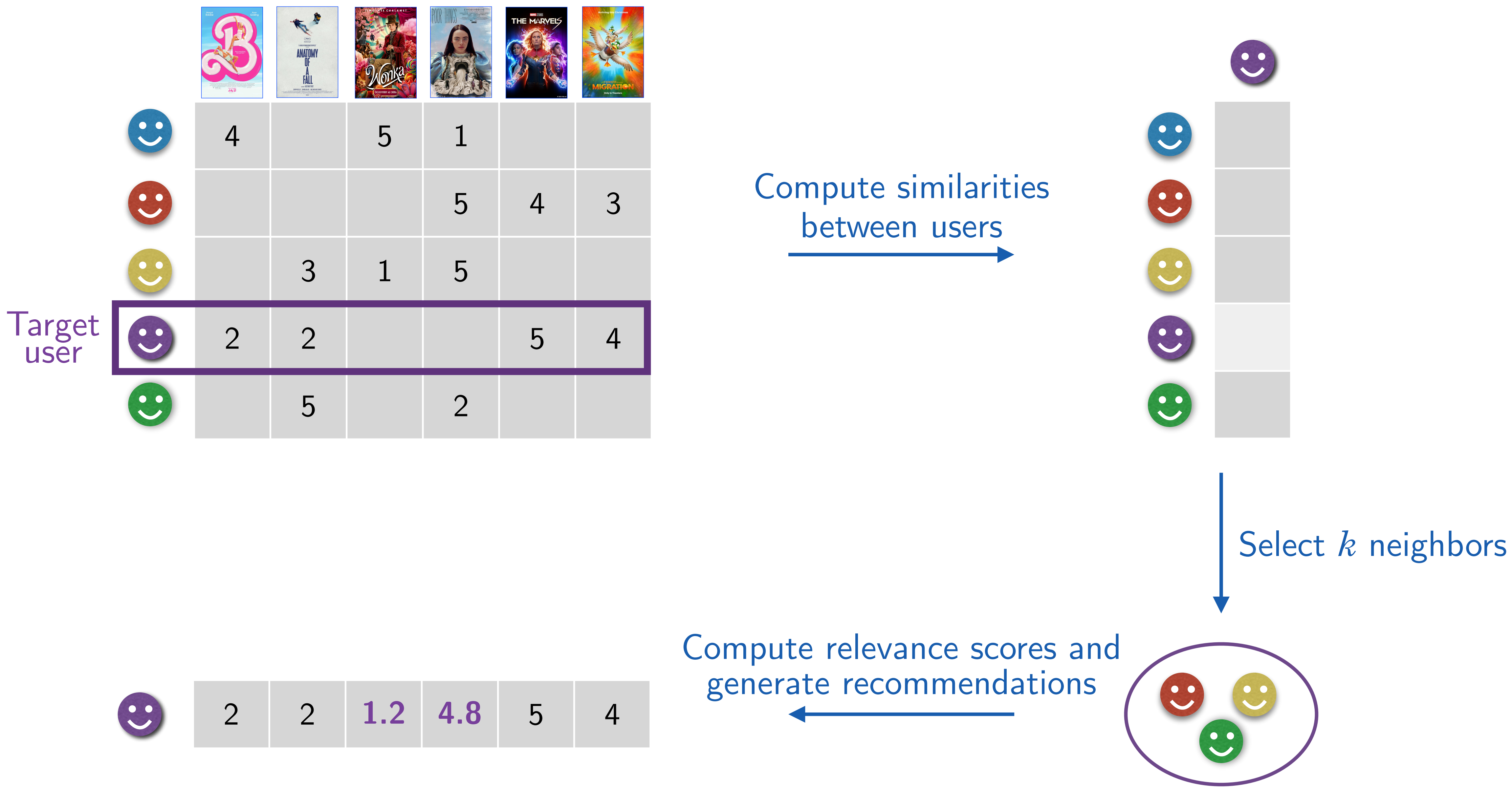
  Similar users are interested in the same items, or

  Similar items interest the same users,

  assuming that user preferences remain stable over time.

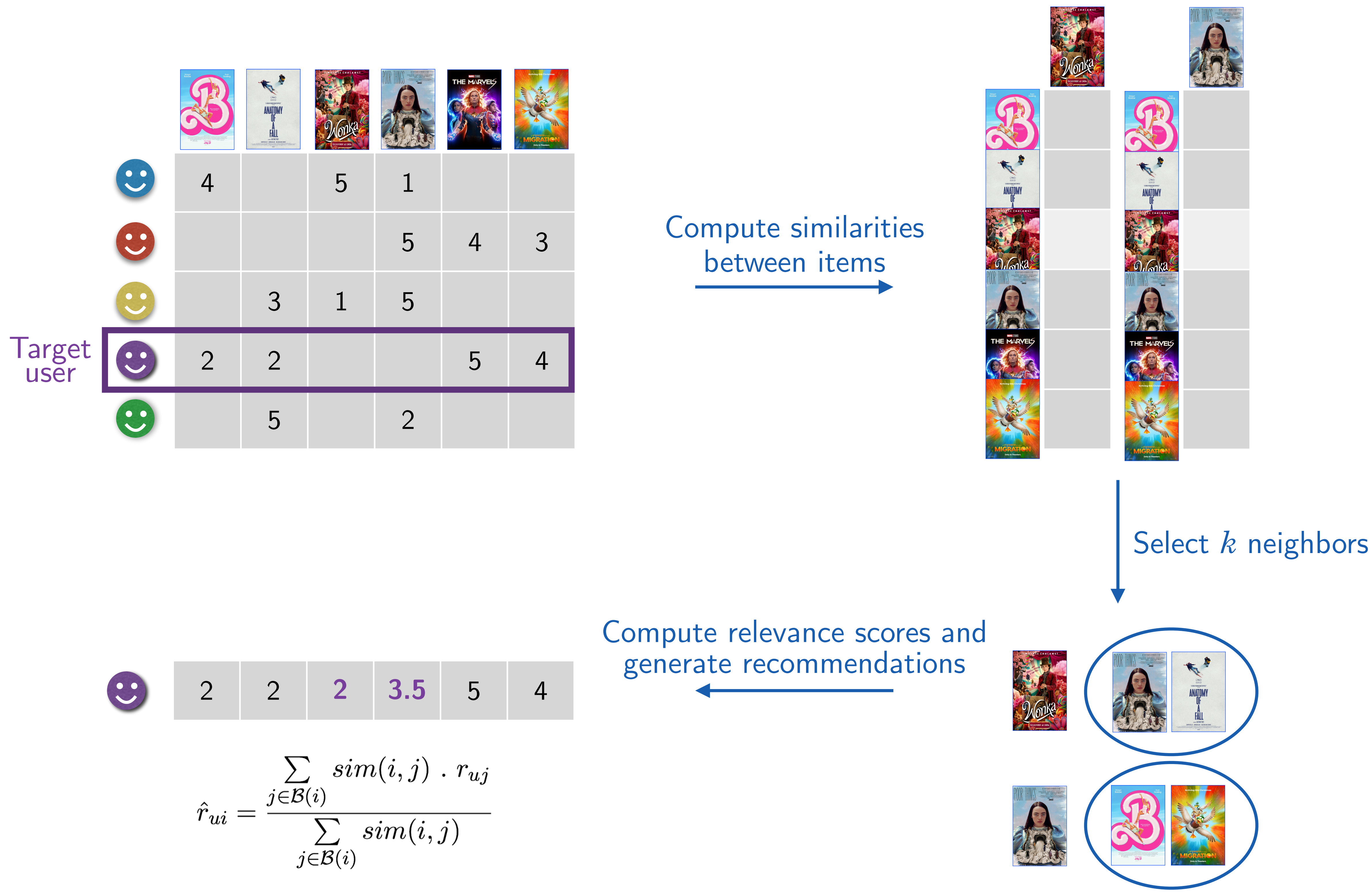
- Components of a k-Nearest Neighbor (kNN) approach:
  - Determine the neighbors of an entity using a similarity measure
  - Compute relevance scores and recommendations based on the neighbors and on previous user interactions

User-based neighborhood approach



$$\hat{r}_{ui} = \frac{\sum\limits_{v \in \mathcal{B}(u)} sim(u,v)\ .\ r_{vi}}{\sum\limits_{v \in \mathcal{B}(u)} sim(u,v)}$$

Item-based neighborhood approach



4		5	1		
			5	4	3
	3	1	5		
2	2			5	4
	5		2		

2	2	2	3.5	5	4

$$\hat{r}_{ui} = \frac{\sum_{j \in \mathcal{B}(i)} sim(i,j) \cdot r_{uj}}{\sum_{j \in \mathcal{B}(i)} sim(i,j)}$$

Similarity measures

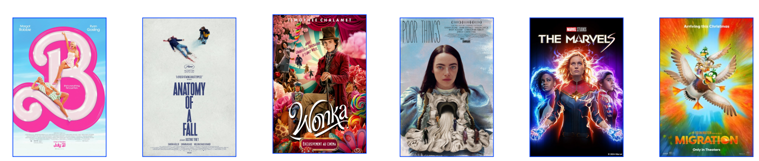
- Jaccard similarity coefficient

$$sim_{JS}(u,v) = \frac{|\mathcal{I}_u \cap \mathcal{I}_v|}{|\mathcal{I}_u \cup \mathcal{I}_v|}$$

Similarity measures

- Jaccard similarity coefficient

$$sim_{JS}(u,v) = \frac{|\mathcal{I}_u \cap \mathcal{I}_v|}{|\mathcal{I}_u \cup \mathcal{I}_v|}$$



							JS
	4		5	1			0.16
		5			4	3	0.75
		3	1	5			0.16
Target user	2	2			5	4	
	1	5		2			0.4

Similarity measures

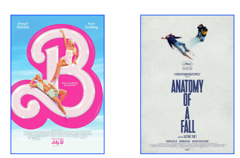
- Cosine similarity

$$sim_{CS}(u,v) = \frac{\mathbf{r}_u^\top \ . \ \mathbf{r}_v^\top}{\|\mathbf{r}_u^\top\| \ \|\mathbf{r}_v^\top\|}$$

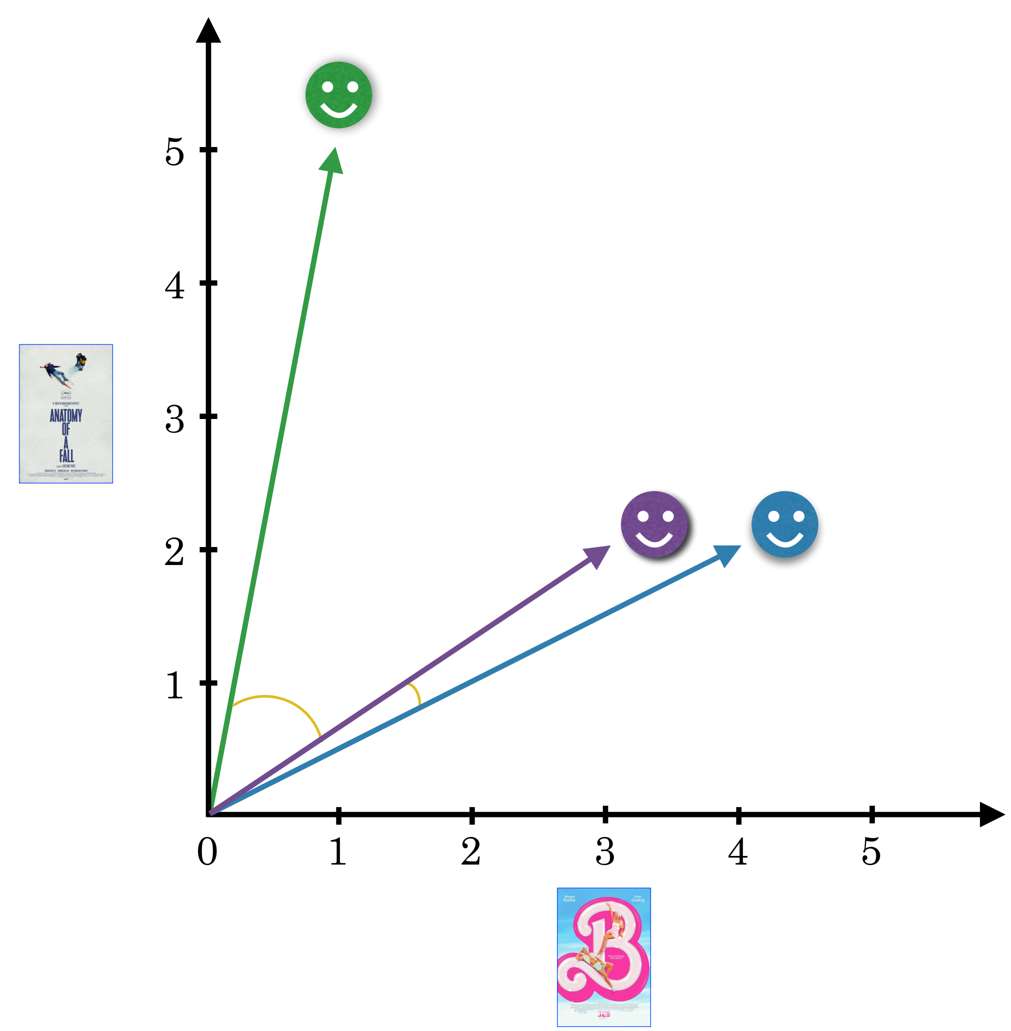
Similarity measures

- Cosine similarity

$$sim_{CS}(u,v) = \frac{\mathbf{r}_u^\top \cdot \mathbf{r}_v^\top}{\|\mathbf{r}_u^\top\| \, \|\mathbf{r}_v^\top\|}$$



4	2
	5
	3
3	2
1	5



Similarity measures

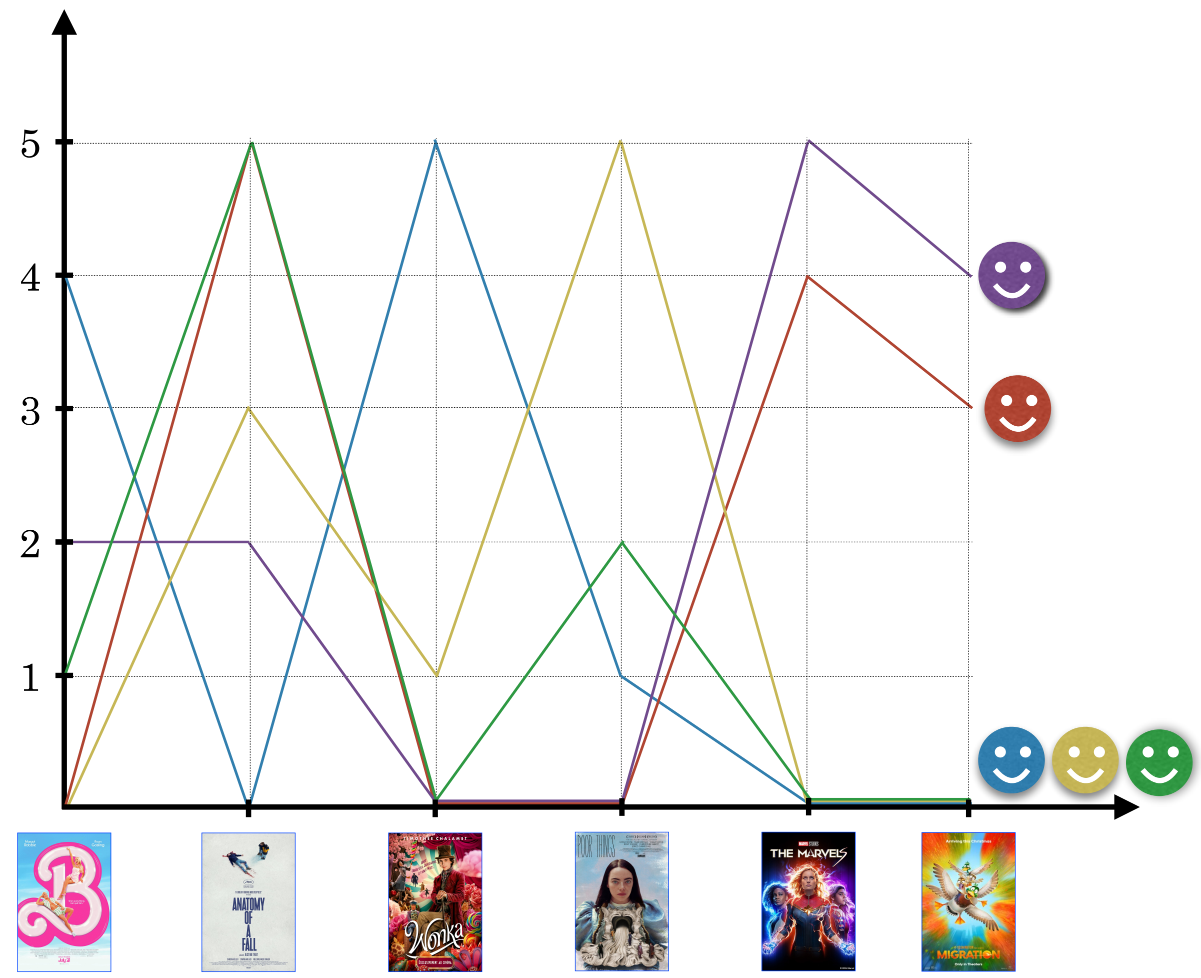
- Pearson correlation

$$sim_{PC}(u,v) = \frac{\sum\limits_{x \in \mathcal{I}_{uv}} (r_{ux} - \overline{r}_u)(r_{vx} - \overline{r}_v)}{\sqrt{\sum\limits_{x \in \mathcal{I}_{uv}} (r_{ux} - \overline{r}_u)^2} \sqrt{\sum\limits_{x \in \mathcal{I}_{uv}} (r_{vx} - \overline{r}_v)^2}}$$

Similarity measures

- Pearson correlation

$$sim_{PC}(u,v) = \frac{\sum_{x \in \mathcal{I}_{uv}} (r_{ux} - \overline{r}_u)(r_{vx} - \overline{r}_v)}{\sqrt{\sum_{x \in \mathcal{I}_{uv}} (r_{ux} - \overline{r}_u)^2} \sqrt{\sum_{x \in \mathcal{I}_{uv}} (r_{vx} - \overline{r}_v)^2}}$$



Memory-based approaches

- User-based *vs.* Item-based:

  Item relationships tend to be more stable than user relationships.

Memory-based approaches

- User-based *vs.* Item-based:

  Item relationships tend to be more stable than user relationships.

- *Offline* component: Compute similarities;

  Online component: Generating predictions and recommendations.

Memory-based approaches

- User-based *vs.* Item-based:

  Item relationships tend to be more stable than user relationships.
- *Offline* component: Compute similarities;

  Online component: Generating predictions and recommendations.
- Pros:
  - Simple and intuitive approach, with only one parameter (k)

Memory-based approaches

- User-based *vs.* Item-based:

  Item relationships tend to be more stable than user relationships.
- *Offline* component: Compute similarities;

  Online component: Generating predictions and recommendations.
- Pros:
  - Simple and intuitive approach, with only one parameter (k)
  - Recommendations are easy to explain

Memory-based approaches

- User-based *vs.* Item-based:

  Item relationships tend to be more stable than user relationships.
- *Offline* component: Compute similarities;

  Online component: Generating predictions and recommendations.
- Pros:
  - Simple and intuitive approach, with only one parameter (k)
  - Recommendations are easy to explain
- Cons:
  - Scalability issues (neighborhood computation)

Memory-based approaches

- User-based *vs.* Item-based:

  Item relationships tend to be more stable than user relationships.

- *Offline* component: Compute similarities;

  Online component: Generating predictions and recommendations.

- Pros:
  - Simple and intuitive approach, with only one parameter (k)
  - Recommendations are easy to explain

- Cons:
  - Scalability issues (neighborhood computation)
  - Lack of sensitivity to sparse data

Memory-based approaches

- User-based *vs.* Item-based:

  Item relationships tend to be more stable than user relationships.

- *Offline* component: Compute similarities;

  Online component: Generating predictions and recommendations.

- Pros:
  - Simple and intuitive approach, with only one parameter (k)
  - Recommendations are easy to explain

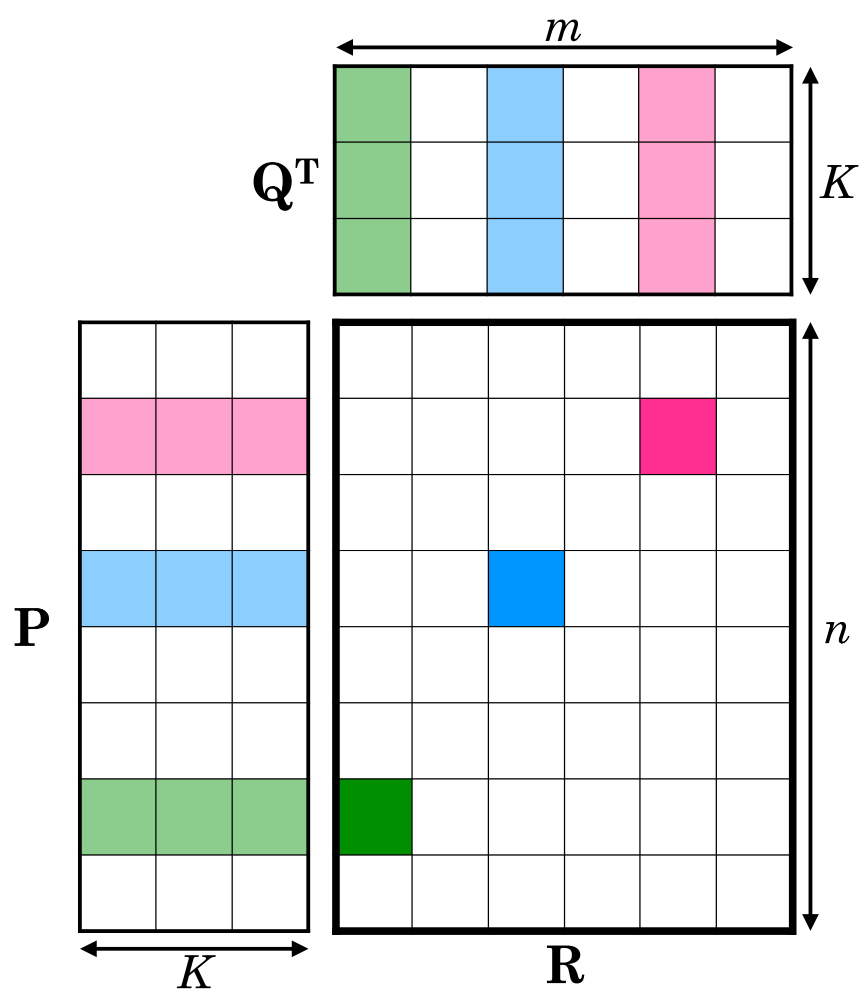
- Cons:
  - Scalability issues (neighborhood computation)
  - Lack of sensitivity to sparse data
  - Usually perform worse than model-based approaches

Model-based approaches

- Learn a predictive model representing users and items.
- Most popular technique: Matrix Factorization (MF)
  - Good accuracy and scalability

Matrix Factorization framework

- Represent users and items in a common space of latent factors of dimensionality K.
- Values of P and Q represent to which extent the user is attracted by or the item has a certain feature, respectively.

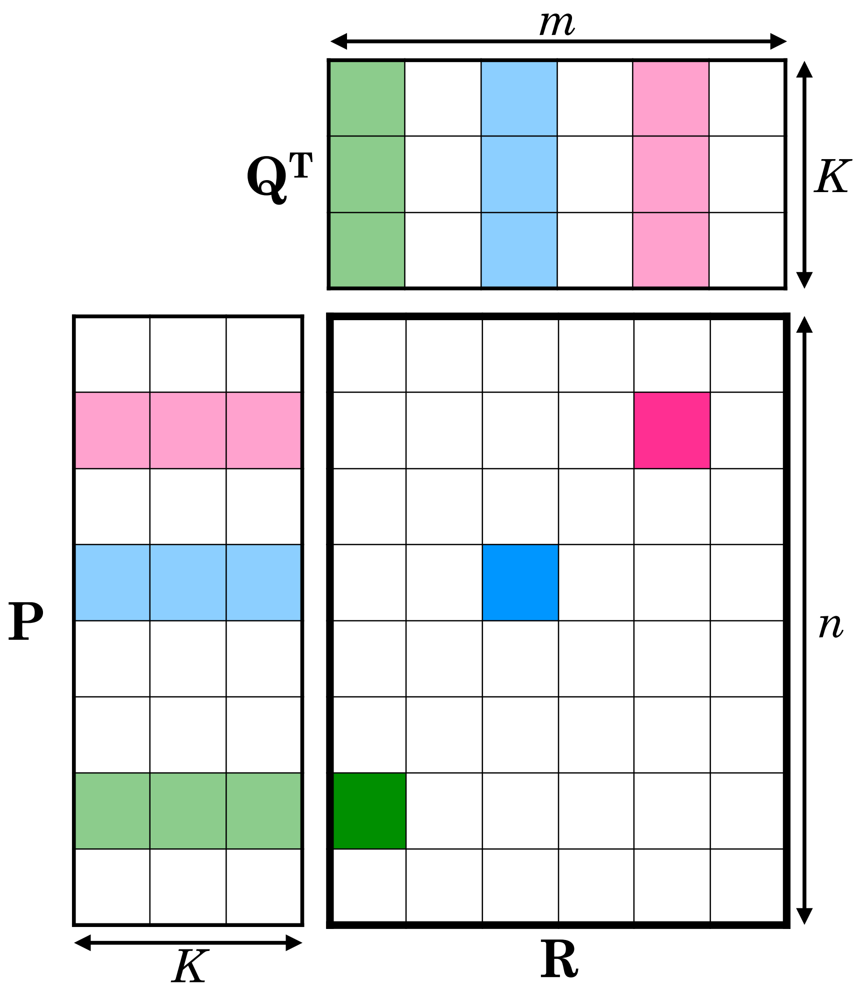


P : User latent matrix
Q : Item latent matrix
R : Feedback matrix

Matrix Factorization framework

- Represent users and items in a common space of latent factors of dimensionality K.
- Values of P and Q represent to which extent the user is attracted by or the item has a certain feature, respectively.
- Hypothesis of MF:

$$\hat{r}_{ui} = \mathbf{p}_u \mathbf{q}_i^\top$$



P : User latent matrix
Q : Item latent matrix
R : Feedback matrix

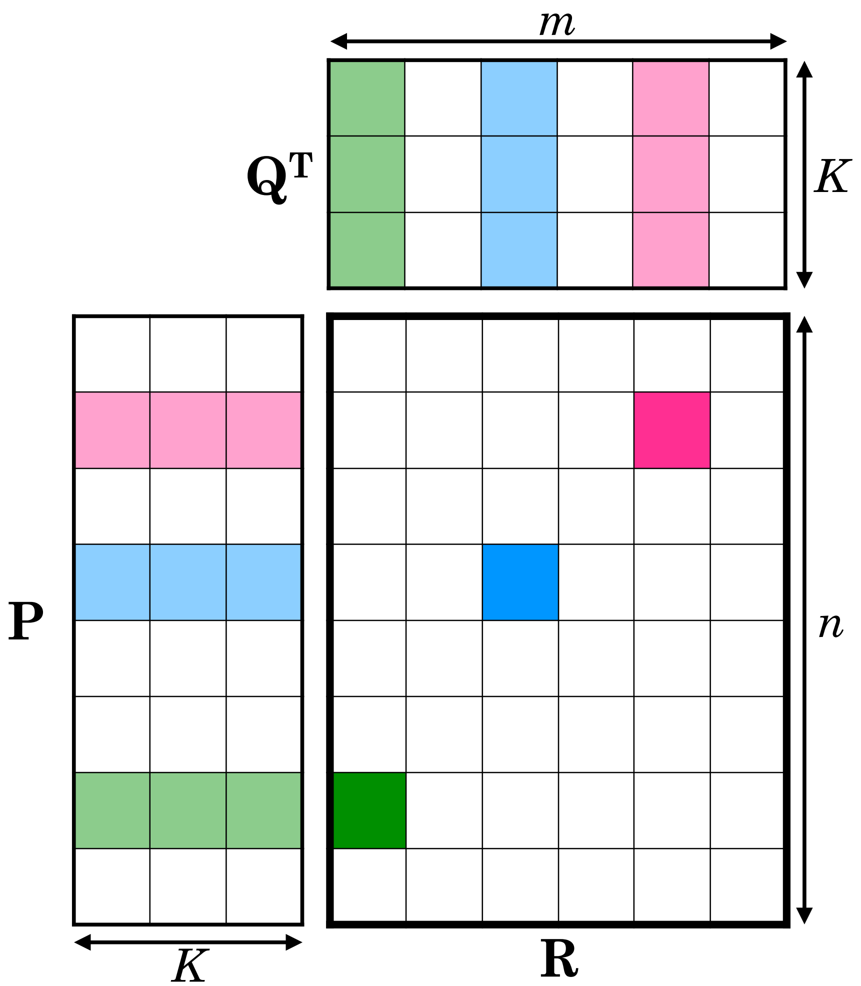
Matrix Factorization framework

- Represent users and items in a common space of latent factors of dimensionality K.
- Values of P and Q represent to which extent the user is attracted by or the item has a certain feature, respectively.
- Hypothesis of MF:

$$\hat{r}_{ui} = \mathbf{p}_u \mathbf{q}_i^\top$$

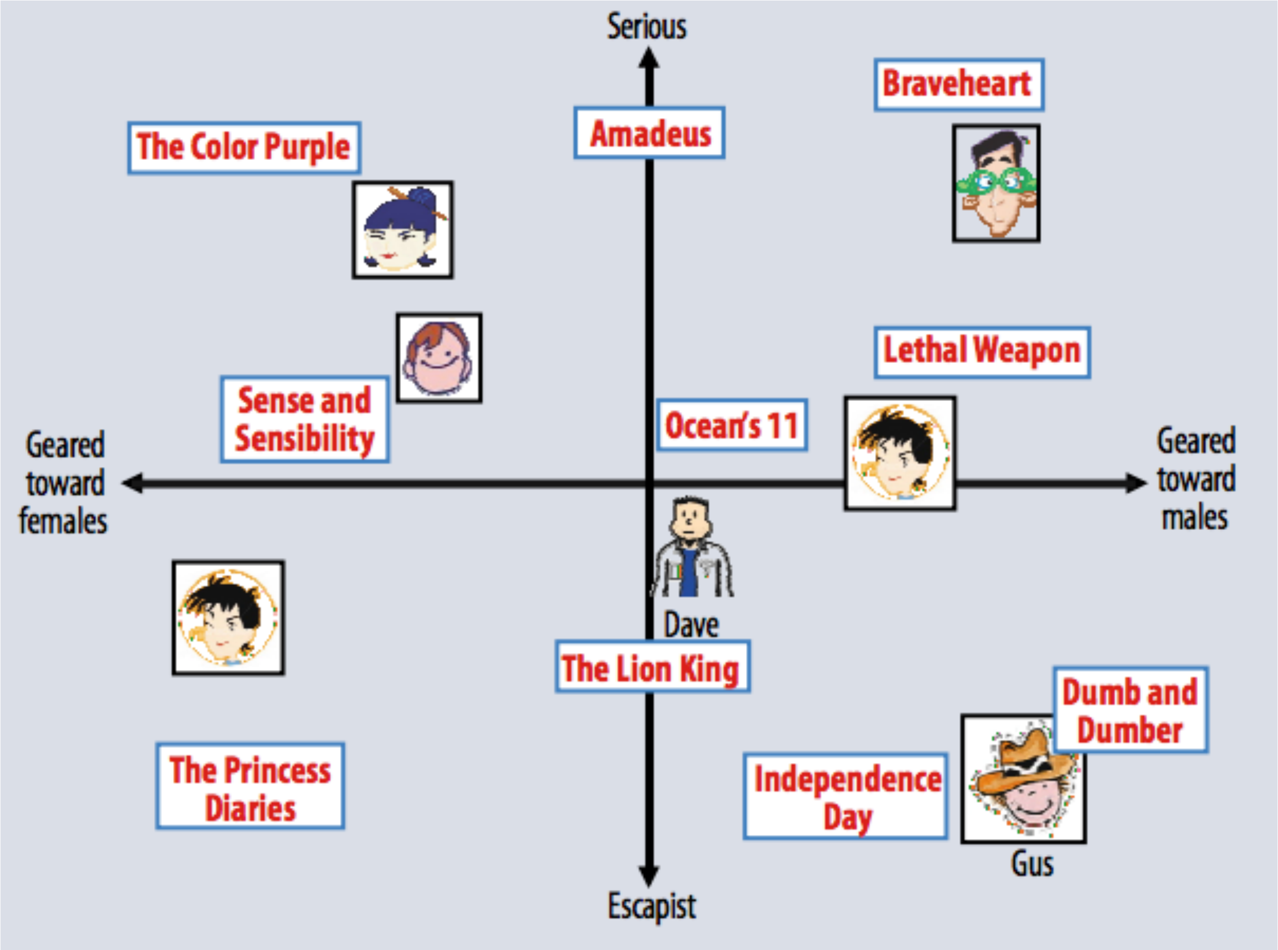
Learning task. Determine P and Q given R.

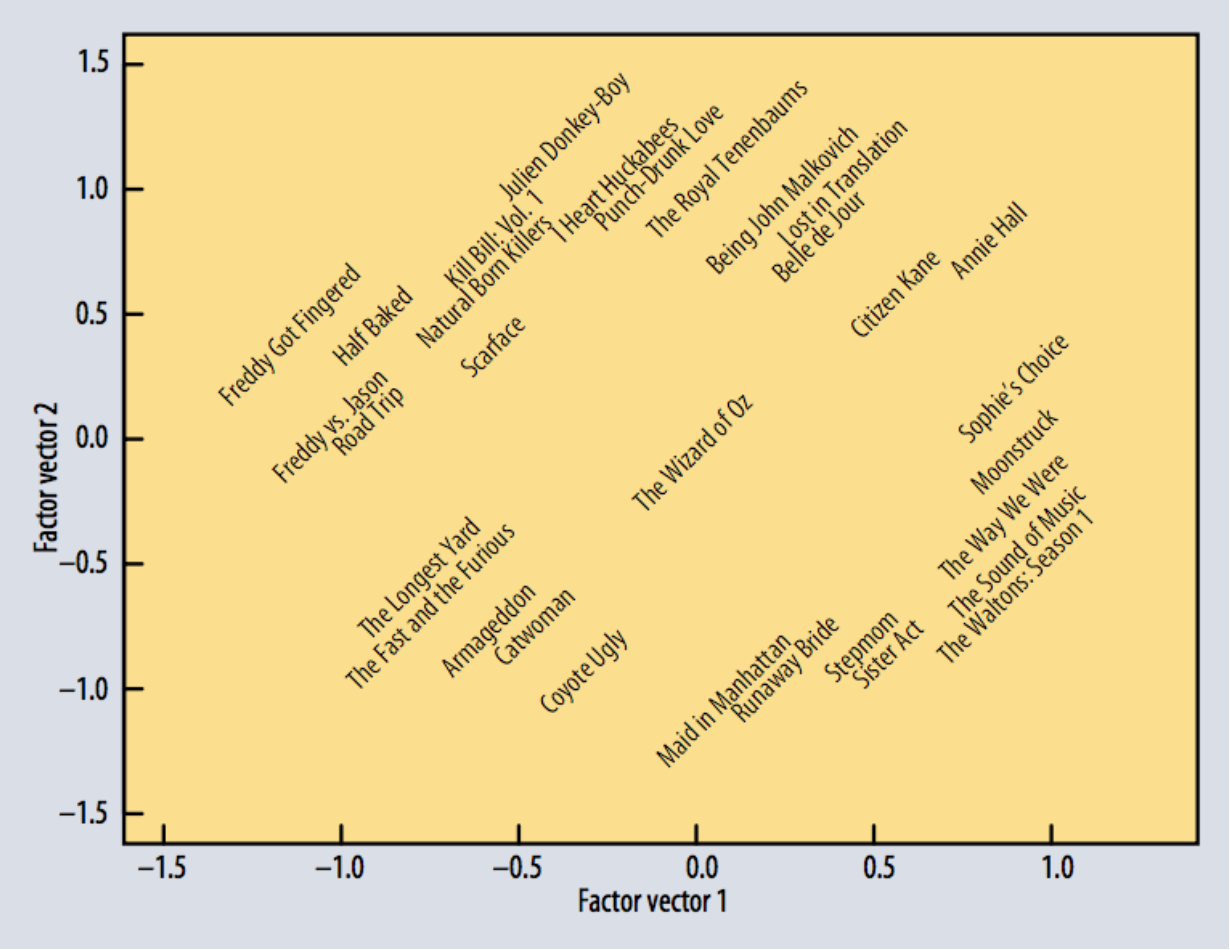
- Several possible approaches, for the application of RS.



P : User latent matrix
Q : Item latent matrix
R : Feedback matrix

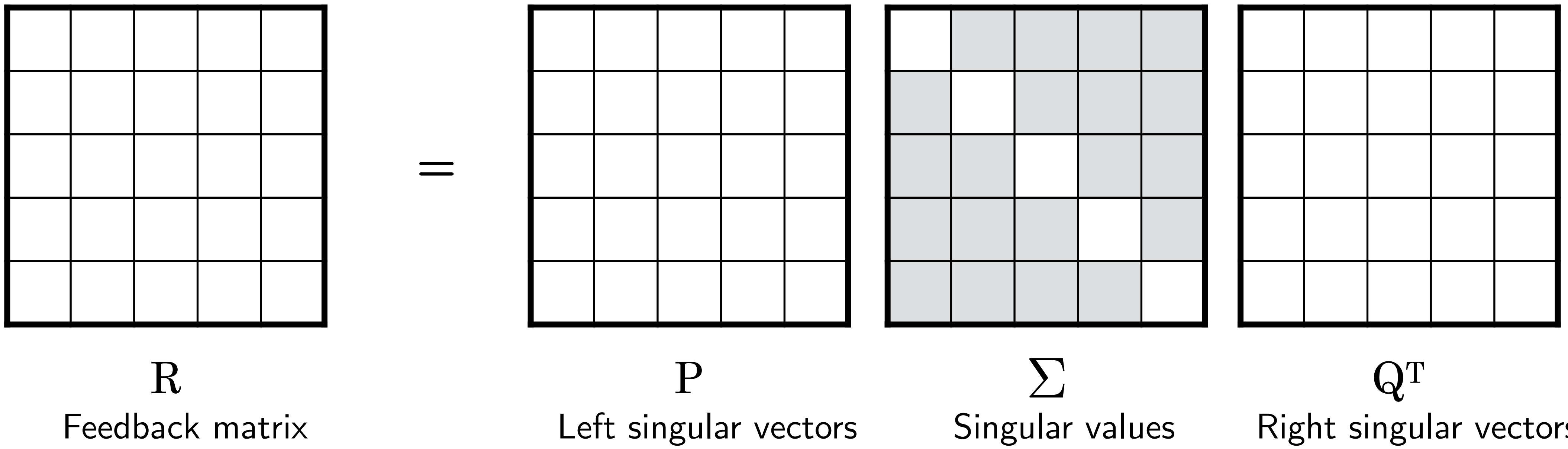
Matrix Factorization framework





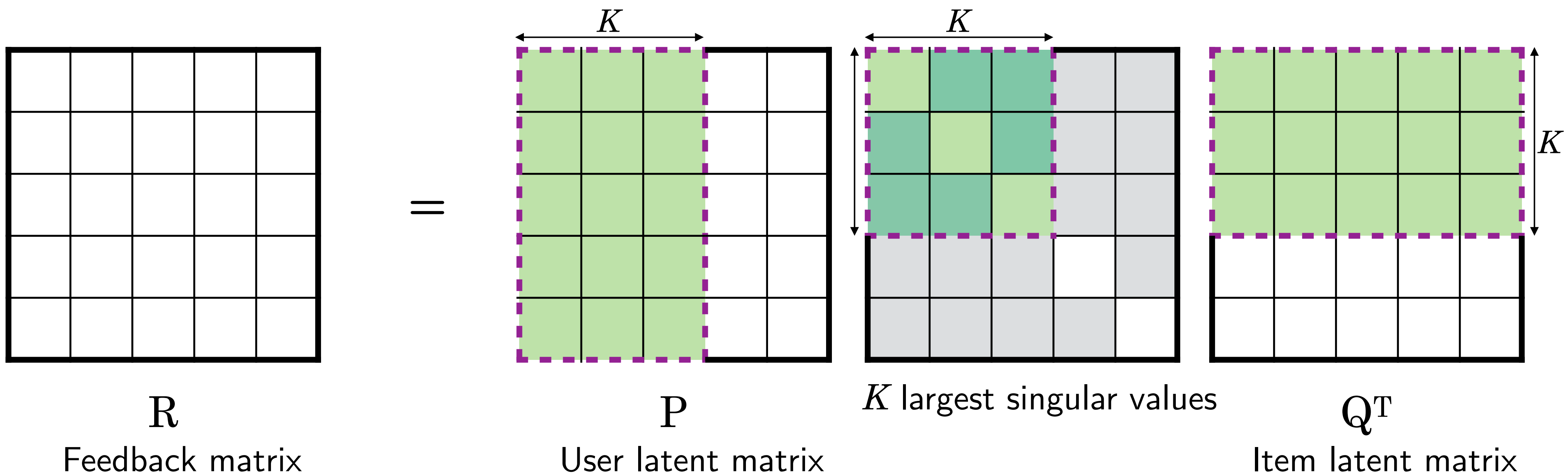
Yehuda Koren, Robert Bell, and Chris Volinsky. *Matrix Factorization Techniques For Recommender Systems*. IEEE Computer, (8), 2009.

Singular Value Decomposition



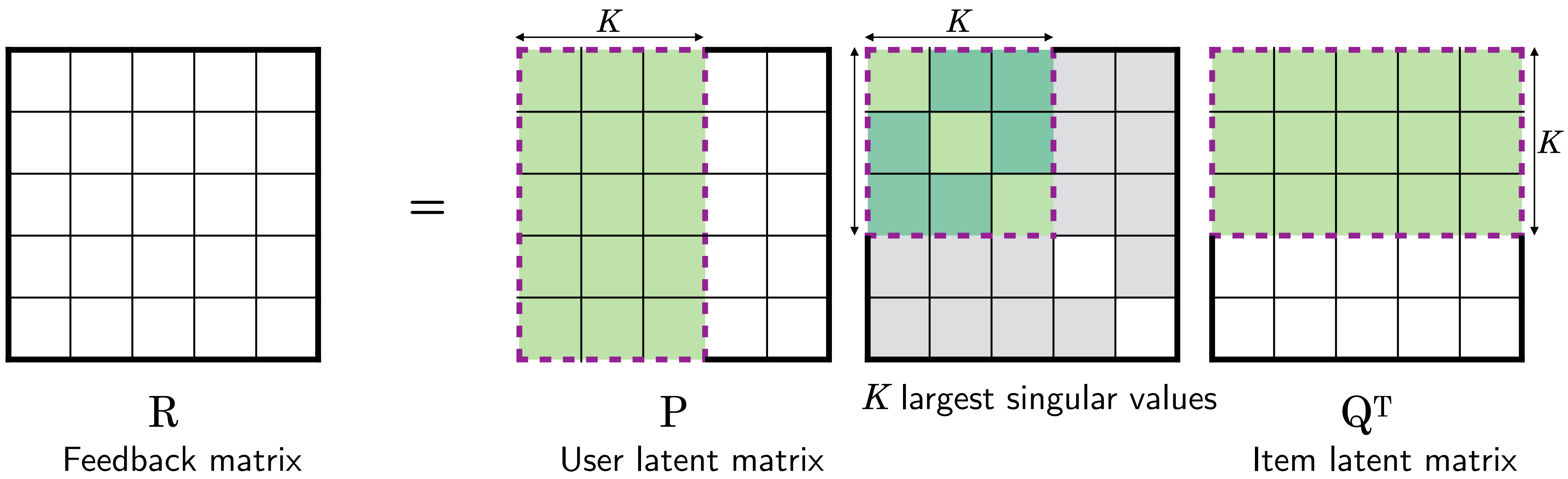
$$\mathbf{R} = \mathbf{P}\mathbf{\Sigma}\mathbf{Q}^\top = \sum_{s=1}^{d} \sigma_s \mathbf{p}_s \mathbf{q}_s^\top$$

Singular Value Decomposition



Rank-K approximation of R: $$\hat{\mathbf{R}} = \sum_{s=1}^{K} \sigma_s \mathbf{p}_s \mathbf{q}_s^\top$$

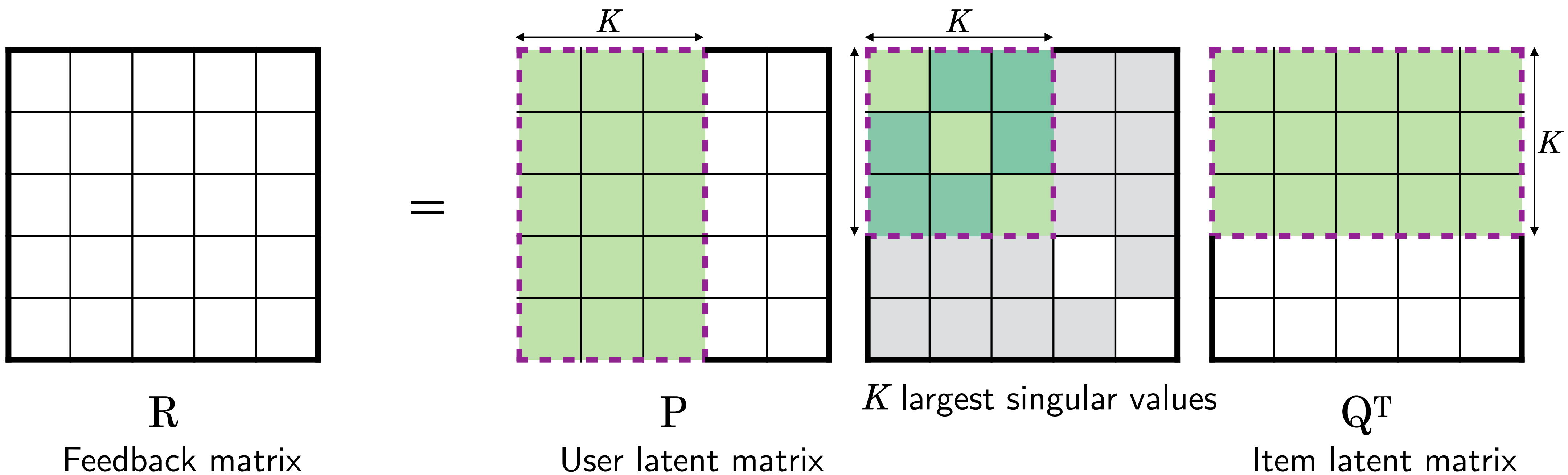
Singular Value Decomposition



Rank-K approximation of R: $\hat{\mathbf{R}} = \sum_{s=1}^{K} \sigma_s \mathbf{p}_s \mathbf{q}_s^\top$

- Applying SVD for collaborative filtering: Sparsity issue
  - Filling R with default values, *e.g.*, average value per user or item

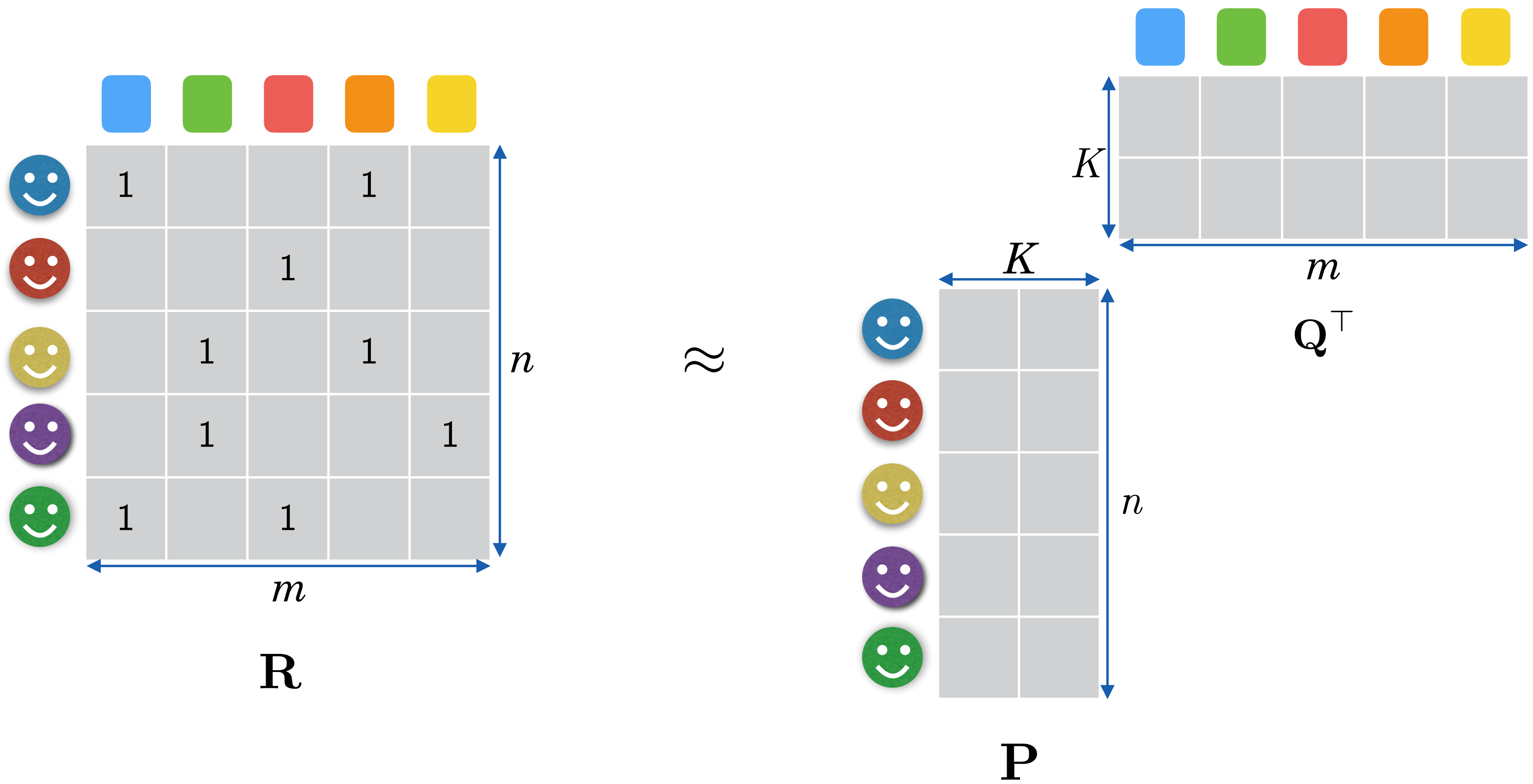
Singular Value Decomposition



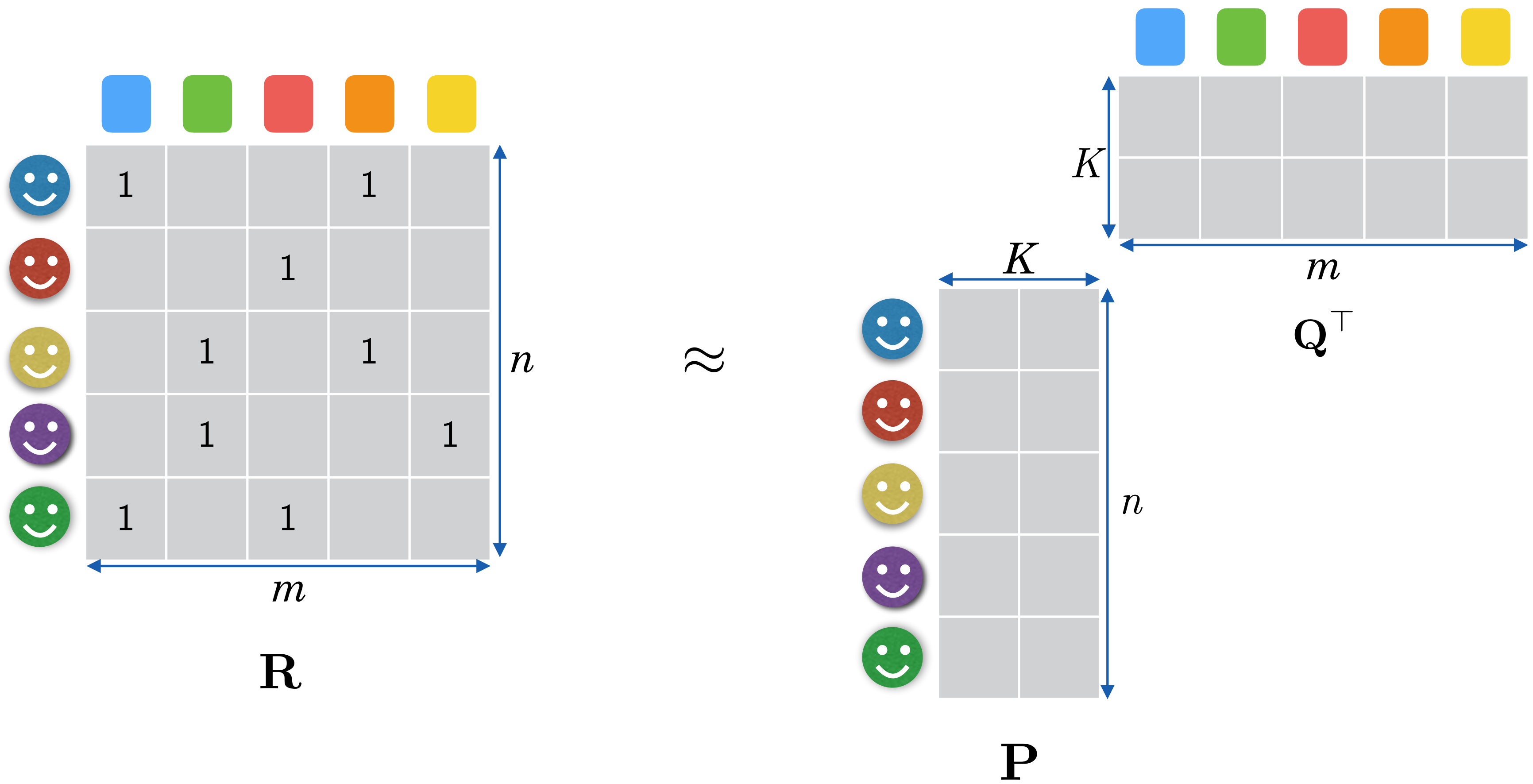
Rank-K approximation of R: $\hat{\mathbf{R}} = \sum_{s=1}^{K} \sigma_s \mathbf{p}_s \mathbf{q}_s^\top$

- Applying SVD for collaborative filtering: Sparsity issue
  - Filling R with default values, *e.g.*, average value per user or item
    - Inaccurate and very expensive

Minimizing the squared loss



Minimizing the squared loss

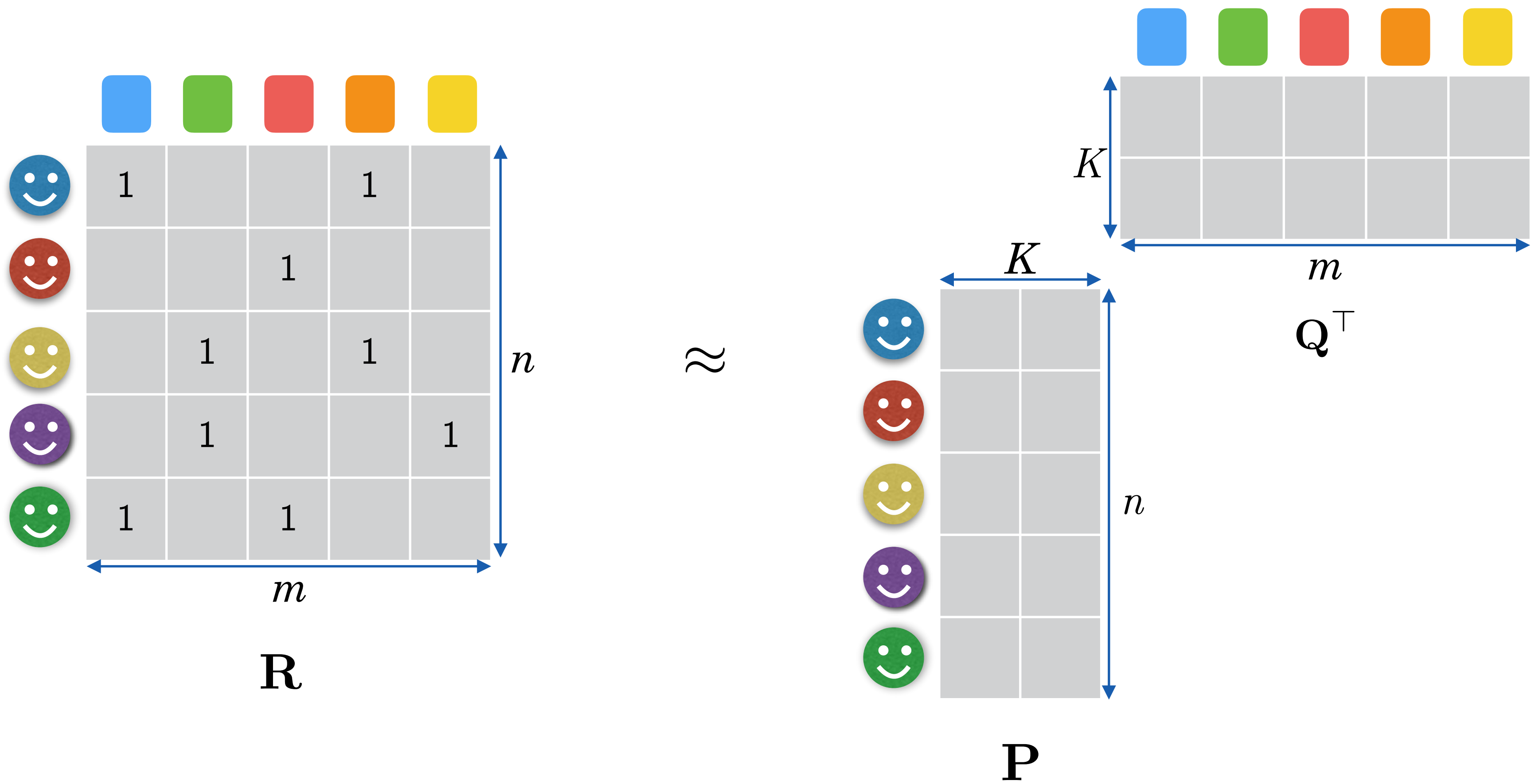


$$\min_{\mathbf{P},\mathbf{Q}} \boxed{\|\mathbf{R} - \mathbf{P}\mathbf{Q}^\top\|_F^2} + \boxed{\lambda(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2)}$$

Least-squares - fitting ability

Regularization - prevent overfitting

Minimizing the squared loss



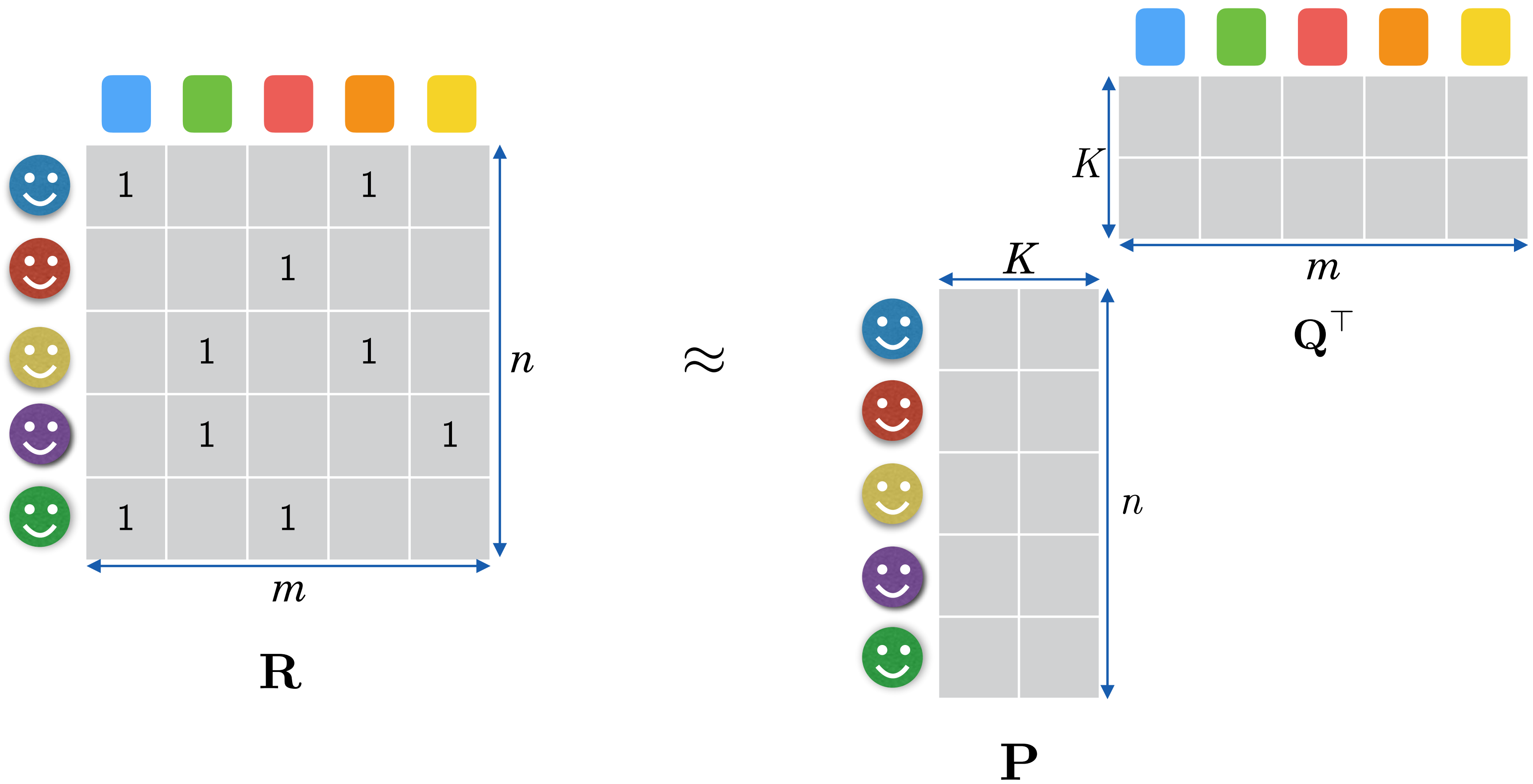
$$\min_{\mathbf{P},\mathbf{Q}} \boxed{\left\|\mathbf{R} - \mathbf{P}\mathbf{Q}^\top\right\|_F^2} + \boxed{\lambda(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2)}$$

Least-squares - fitting ability

Regularization - prevent overfitting

$$\min_{\mathbf{p}_*,\mathbf{q}_*} \sum_{(u,i)\in\mathcal{D}} (r_{ui} - \mathbf{p}_u\mathbf{q}_i^\top)^2 + \lambda(\sum_{u=1}^{n}\|\mathbf{p}_u\|^2 + \sum_{i=1}^{m}\|\mathbf{q}_i\|^2)))$$

Minimizing the squared loss



$$\min_{\mathbf{P},\mathbf{Q}} \underbrace{\|\mathbf{R} - \mathbf{P}\mathbf{Q}^\top\|_F^2 + \lambda(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2)}_{} \rightarrow \mathcal{L}(\mathbf{P},\mathbf{Q})$$

$$\min_{\mathbf{p}_*,\mathbf{q}_*} \sum_{(u,i)\in\mathcal{D}} (r_{ui} - \mathbf{p}_u\mathbf{q}_i^\top)^2 + \lambda(\sum_{u=1}^{n}\|\mathbf{p}_u\|^2 + \sum_{i=1}^{m}\|\mathbf{q}_i\|^2)))$$

Learning the parameters

Stochastic Gradient Descent

- Performs a parameter update for each observation (u, i)
- Faster than gradient descent
- Requires fixing the value of the learning rate η

Learning the parameters

Stochastic Gradient Descent

- Performs a parameter update for each observation (u, i)
- Faster than gradient descent
- Requires fixing the value of the learning rate η

Parameter updates:

$$\mathbf{p}_u \leftarrow \mathbf{p}_u - \eta \frac{\partial \mathcal{L}(\mathbf{P}, \mathbf{Q})}{\partial \mathbf{p}_u} \;, \quad \mathbf{q}_i \leftarrow \mathbf{q}_i - \eta \frac{\partial \mathcal{L}(\mathbf{P}, \mathbf{Q})}{\partial \mathbf{q}_i}$$

For each factor l:

$$\frac{\partial \mathcal{L}(\mathbf{P}, \mathbf{Q})}{\partial p_{ul}} = 2q_{il}(\mathbf{p}_u \mathbf{q}_i^\top - r_{ui}) \;+\; 2\lambda p_{ul}$$

$$\frac{\partial \mathcal{L}(\mathbf{P}, \mathbf{Q})}{\partial q_{il}} = 2p_{ul}\;(\mathbf{p}_u \mathbf{q}_i^\top - r_{ui}) \;+\; 2\lambda q_{il}$$

Learning the parameters

Algorithm 2 Stochastic Gradient Descent (SGD)

Input: Rating matrix $\mathbf{R}$, set of observed ratings $\mathcal{D}$, stopping criterion ϵ, rank K, learning rate η, and regularization parameter λ

Output: Optimal user feature matrix $\mathbf{P}$, optimal item feature matrix $\mathbf{Q}$

```
Initialize P and Q, e.g., randomly
while not converged, e.g., ||R − PQ^⊤|| > ε, do
    for each (u, i) ∈ D do
        e_ui = p_u q_i^⊤ − r_ui
        p_u ← p_u − 2η(e_ui q_i + λ p_u)
        q_i ← q_i − 2η(e_ui p_u + λ q_i)
    end for
end while
```

Initialize $\mathbf{P}$ and $\mathbf{Q}$, e.g., randomly

while not converged, e.g., $\left\|\mathbf{R} - \mathbf{P}\mathbf{Q}^\top\right\| > \epsilon$, **do**

  for each $(u, i) \in \mathcal{D}$ **do**

    $e_{ui} = \mathbf{p}_u \mathbf{q}_i^\top - r_{ui}$

    $\mathbf{p}_u \leftarrow \mathbf{p}_u - 2\eta(e_{ui}\mathbf{q}_i + \lambda \mathbf{p}_u)$

    $\mathbf{q}_i \leftarrow \mathbf{q}_i - 2\eta(e_{ui}\mathbf{p}_u + \lambda \mathbf{q}_i)$

  end for

end while

Learning the parameters

Alternating Least Squares

- Iterate over the parameters, fix them, and solve the optimization problem
- Well-designed for parallelization

Learning the parameters

Algorithm 3 Alternating Least Squares (ALS)

Input: Rating matrix $\mathbf{R}$ and regularization parameter λ

Output: Optimal user feature matrix $\mathbf{P}$, optimal item feature matrix $\mathbf{Q}$

```
Initialize P and Q, e.g., randomly
while not converged do
    for u ∈ 1, ..., m do                                          ▷ update users
        p_u ← ( Σ_{i∈I_u} q_i q_i^⊤ + λI )^{-1} Σ_{i∈I_u} r_ui q_i
    end for
    for i ∈ 1, ..., n do                                          ▷ update items
        q_i ← ( Σ_{u∈U_i} p_u p_u^⊤ + λI )^{-1} Σ_{u∈U_i} r_ui p_u
    end for
end while
```

$$\mathbf{p}_u \leftarrow \Big(\sum_{i\in\mathcal{I}_u} \mathbf{q}_i\mathbf{q}_i^\top + \lambda\mathbf{I}\Big)^{-1} \sum_{i\in\mathcal{I}_u} r_{ui}\mathbf{q}_i$$

$$\mathbf{q}_i \leftarrow \Big(\sum_{u\in\mathcal{U}_i} \mathbf{p}_u\mathbf{p}_u^\top + \lambda\mathbf{I}\Big)^{-1} \sum_{u\in\mathcal{U}_i} r_{ui}\mathbf{p}_u$$

Beyond optimizing the squared loss function

Rating prediction problem

- Task: *Matrix completion*, *i.e.*, accurately predict missing ratings
  - Performance metric: RMSE
  - Objective function: Squared loss function

Beyond optimizing the squared loss function

Rating prediction problem

- Task: *Matrix completion*, *i.e.*, accurately predict missing ratings
  - Performance metric: RMSE
  - Objective function: Squared loss function

With the evolution of the recommendation problem, several objective functions were explored.

- Point-wise functions, *e.g.*, squared loss, $\ell^{point}(r_{ui}, \hat{r}_{ui})$
  - Focus on the accuracy of individual preferences

Beyond optimizing the squared loss function

Rating prediction problem

- Task: *Matrix completion*, *i.e.*, accurately predict missing ratings
  - Performance metric: RMSE
  - Objective function: Squared loss function

With the evolution of the recommendation problem, several objective functions were explored.

- Point-wise functions, *e.g.*, squared loss, $\ell^{point}(r_{ui}, \hat{r}_{ui})$
  - Focus on the accuracy of individual preferences
- Pair-wise functions, $\ell^{pair}(r_{uij}, \hat{r}_{uij})$
  - Consider the relative ranking of predictions for pairs of items

Beyond optimizing the squared loss function

Rating prediction problem

- Task: *Matrix completion*, *i.e.*, accurately predict missing ratings
  - Performance metric: RMSE
  - Objective function: Squared loss function

With the evolution of the recommendation problem, several objective functions were explored.

- Point-wise functions, *e.g.*, squared loss, $\ell^{point}(r_{ui}, \hat{r}_{ui})$
  - Focus on the accuracy of individual preferences
- Pair-wise functions, $\ell^{pair}(r_{uij}, \hat{r}_{uij})$
  - Consider the relative ranking of predictions for pairs of items
- List-wise functions
  - Reflect the distance between the complete recommended list and the reference one

Dealing with implicit feedback

Implicit feedback is often *positive-only*

- One-Class Collaborative Filtering

- Transform the set of implicit observations by integrating additional negative feedback,
- Adapting the objective function

Augmenting implicit feedback datasets

- All Missing As Negative: Convert implicit feedback to binary ratings
  - Observed interactions take the value of 1 in the rating matrix and unobserved interactions the value of 0.
    - The model will be strongly biased towards negative feedback and will tend to always predict 0.

Augmenting implicit feedback datasets

- All Missing As Negative: Convert implicit feedback to binary ratings
  - Observed interactions take the value of 1 in the rating matrix and unobserved interactions the value of 0.
    - The model will be strongly biased towards negative feedback and will tend to always predict 0.
- All Missing As Unknown: Ignore unobserved interactions
    - The model will be strongly biased towards positive feedback and this would result in a trivial solution.

Augmenting implicit feedback datasets

- All Missing As Negative: Convert implicit feedback to binary ratings
  - Observed interactions take the value of 1 in the rating matrix and unobserved interactions the value of 0.
    - The model will be strongly biased towards negative feedback and will tend to always predict 0.
- All Missing As Unknown: Ignore unobserved interactions
    - The model will be strongly biased towards positive feedback and this would result in a trivial solution.
- In-between solution: Discriminate negative items from the unobserved ones
  - Randomly sampling negative items, or
  - Using weighting techniques.

Defining the objective function

- Focus on the ranking quality of items instead of the value of predicted ratings
- Two seminal MF approaches to deal with implicit feedback:
  - Weighted Regularized MF (WRMF)
    - Relies on a weighted point-wise function
  - Bayesian Personalized Ranking (BPR)
    - Relies on a pair-wise function

Weighted Regularized Matrix Factorization

- The weight indicates the confidence we have in the corresponding observation
  - If user u interacted with i, we are confident that u likes i
  - Otherwise, we can make assumptions with different levels of confidence

$$\mathcal{L}(\mathbf{P}, \mathbf{Q}) = \sum_{(u,i) \in \mathcal{D}} c_{ui}(r_{ui} - \mathbf{p}_u \mathbf{q}_i^\top)^2 + \lambda\Omega(\mathbf{P}) + \lambda\Omega(\mathbf{Q})$$

Yifan Hu, Yehuda Koren, and Chris Volinsky. *Collaborative Filtering for Implicit Feedback Datasets.* In Proc. IEEE ICDM, 2008.

Weighted Regularized Matrix Factorization

- The weight indicates the confidence we have in the corresponding observation
  - If user u interacted with i, we are confident that u likes i
  - Otherwise, we can make assumptions with different levels of confidence

$$\mathcal{L}(\mathbf{P},\mathbf{Q}) = \sum_{(u,i)\in\mathcal{D}} c_{ui}(r_{ui} - \mathbf{p}_u\mathbf{q}_i^\top)^2 + \lambda\Omega(\mathbf{P}) + \lambda\Omega(\mathbf{Q})$$

  - Definition of c_{ui}:
    - Constant values for positive and negative observations
    - Consider additional parameters

Yifan Hu, Yehuda Koren, and Chris Volinsky. *Collaborative Filtering for Implicit Feedback Datasets.* In Proc. IEEE ICDM, 2008.

Weighted Regularized Matrix Factorization

- The weight indicates the confidence we have in the corresponding observation
  - If user u interacted with i, we are confident that u likes i
  - Otherwise, we can make assumptions with different levels of confidence

$$\mathcal{L}(\mathbf{P}, \mathbf{Q}) = \sum_{(u,i) \in \mathcal{D}} c_{ui}(r_{ui} - \mathbf{p}_u \mathbf{q}_i^\top)^2 + \lambda\Omega(\mathbf{P}) + \lambda\Omega(\mathbf{Q})$$

  - Definition of c_{ui}:
    - Constant values for positive and negative observations
    - Consider additional parameters

Alternating Least Squares (ALS) is used for optimization in cases where c_{ui} is constant for unobserved items.

Yifan Hu, Yehuda Koren, and Chris Volinsky. *Collaborative Filtering for Implicit Feedback Datasets.* In Proc. IEEE ICDM, 2008.

Bayesian Personalized Ranking

1		1	1		
			1	1	1
	1	1	1		
1	1			1	1
	1		1		

$i >_u j$

j \ i					
		?	-	-	?	?
	?		-	-	?	?
	+	+		?	+	+
	+	+	?		+	+
	?	?	-	-		?
	?	?	-	-	?	

Pairwise preferences for user

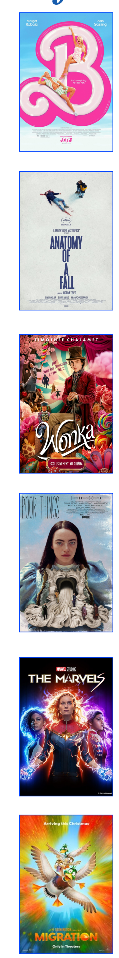
Bayesian Personalized Ranking

1		1	1		
			1	1	1
	1	1	1		
1	1			1	1
	1		1		

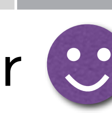
$i >_u j$

i

j



	?	-	-	?	?
?		-	-	?	?
+	+		?	+	+
+	+	?		+	+
?	?	-	-		?
?	?	-	-	?	

Pairwise preferences for user 

$$\mathcal{D}_{bpr} = \{(u,i,j) | i \in \mathcal{I}_u \wedge j \in \mathcal{I} \setminus \mathcal{I}_u\}$$

Bayesian Personalized Ranking

- The goal is to provide the user with a personalized total ranking $>_u$ of all items

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. *BPR: Bayesian Personalized Ranking from Implicit Feedback.* In Proc. UAI, 2009.

Bayesian Personalized Ranking

- The goal is to provide the user with a personalized total ranking $>_u$ of all items
- Maximize $p(\Theta| >_u) \propto p(>_u |\Theta)\, p(\Theta)$

  with $p(i >_u j|\Theta) := \sigma(\hat{r}_{uij}(\Theta))$

  $\hat{r}_{uij} := \hat{r}_{ui} - \hat{r}_{uj}$

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. *BPR: Bayesian Personalized Ranking from Implicit Feedback.* In Proc. UAI, 2009.

Bayesian Personalized Ranking

- The goal is to provide the user with a personalized total ranking $>_u$ of all items
- Maximize $\quad p(\Theta| >_u) \propto p(>_u |\Theta)\, p(\Theta)$

  with $\quad p(i >_u j|\Theta) := \sigma(\hat{r}_{uij}(\Theta))$

  $$\hat{r}_{uij} := \hat{r}_{ui} - \hat{r}_{uj}$$

- Criterion to optimize:

$$\begin{aligned}\text{BPR-OPT} &:= \ln p(\Theta \,|>_u) \\ &= \ln p(>_u|\, \Theta) p(\Theta) \\ &= \sum_{(u,i,j) \in D_{bpr}} \ln \sigma(\hat{r}_{uij}) - \lambda_\Theta \|\Theta\|^2\end{aligned}$$

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. *BPR: Bayesian Personalized Ranking from Implicit Feedback.* In Proc. UAI, 2009.

Bayesian Personalized Ranking

- The goal is to provide the user with a personalized total ranking $>_u$ of all items
- Maximize $p(\Theta| >_u) \propto p(>_u |\Theta)\, p(\Theta)$

  with $p(i >_u j|\Theta) := \sigma(\hat{r}_{uij}(\Theta))$

  $\hat{r}_{uij} := \hat{r}_{ui} - \hat{r}_{uj}$

- Criterion to optimize:

$$\begin{aligned}\text{BPR-O}\textsc{pt} &:= \ln p(\Theta \,|{>_u}) \\ &= \ln p({>_u}|\,\Theta)p(\Theta) \\ &= \sum_{(u,i,j)\in \mathcal{D}_{bpr}} \ln \sigma(\hat{r}_{uij}) - \lambda_\Theta \left\|\Theta\right\|^2\end{aligned}$$

  - Optimization solved using SGD
  - Triples are uniformly sampled from D_{bpr}

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. *BPR: Bayesian Personalized Ranking from Implicit Feedback*. In Proc. UAI, 2009.

Collaborative filtering recap

- Memory-based approaches
  - User-based neighborhood approach
  - Item-based neighborhood approach
  - Similarity measures

Collaborative filtering recap

- Memory-based approaches
  - User-based neighborhood approach
  - Item-based neighborhood approach
  - Similarity measures
- Model-based approaches
  - Matrix Factorization framework
  - SVD
  - Minimizing the squared loss
  - Learning the parameters: SGD, ALS
  - Dealing with implicit feedback: WRMF, BPR

Content-Based Filtering

Content-based filtering for recommendation

- Use content information about users and items to generate recommendations, rather than other users' interactions and preferences.

Content-based filtering for recommendation

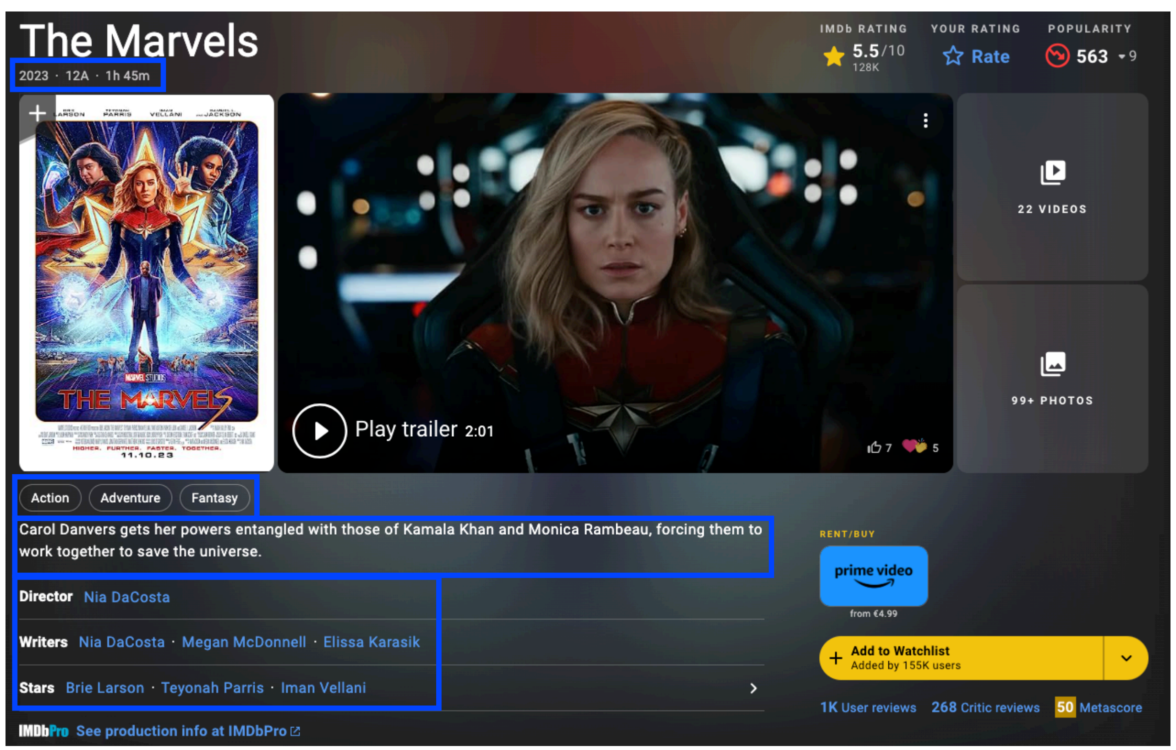
- Use content information about users and items to generate recommendations, rather than other users' interactions and preferences.
- Recommend items that are similar to the ones the user previously interacted with.

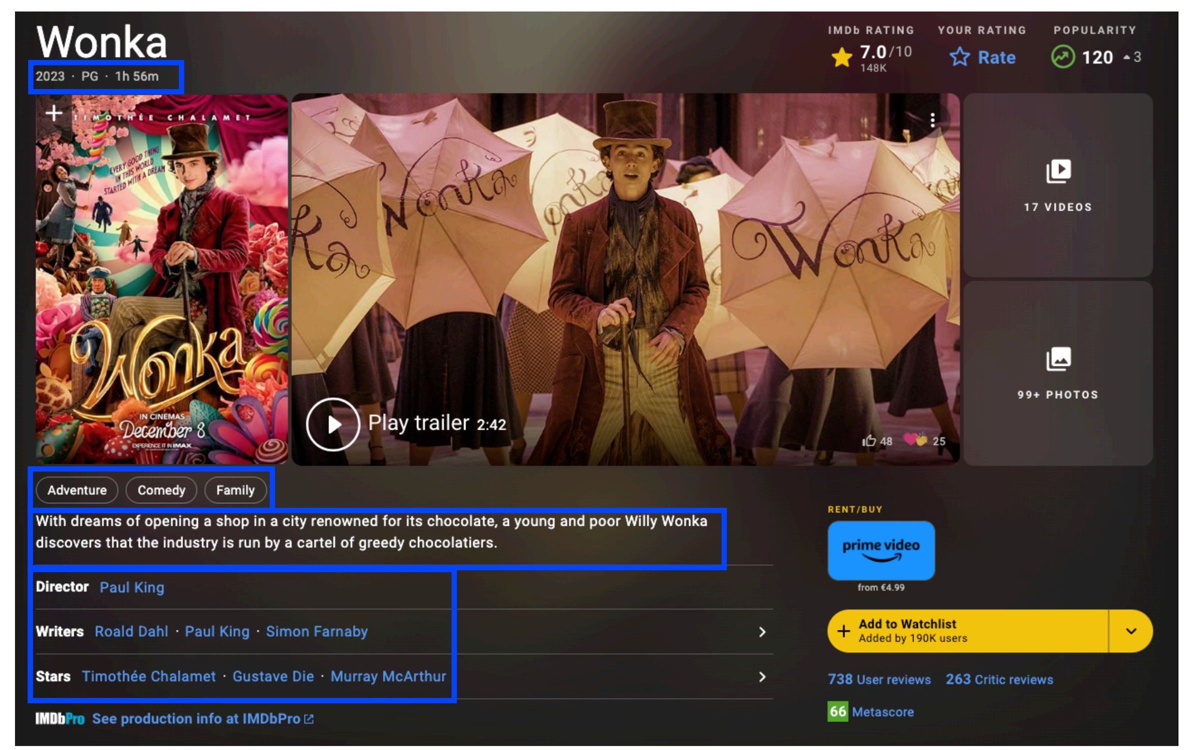
Content-based filtering for recommendation

- Use content information about users and items to generate recommendations, rather than other users' interactions and preferences.
- Recommend items that are similar to the ones the user previously interacted with.
- Develop a content-based RS:
  - Content analyzer. Extract information about the content of items
  - User Profiler. Build user profiles
  - Item retriever. Filter items according to the similarity between profiles

Content of an item

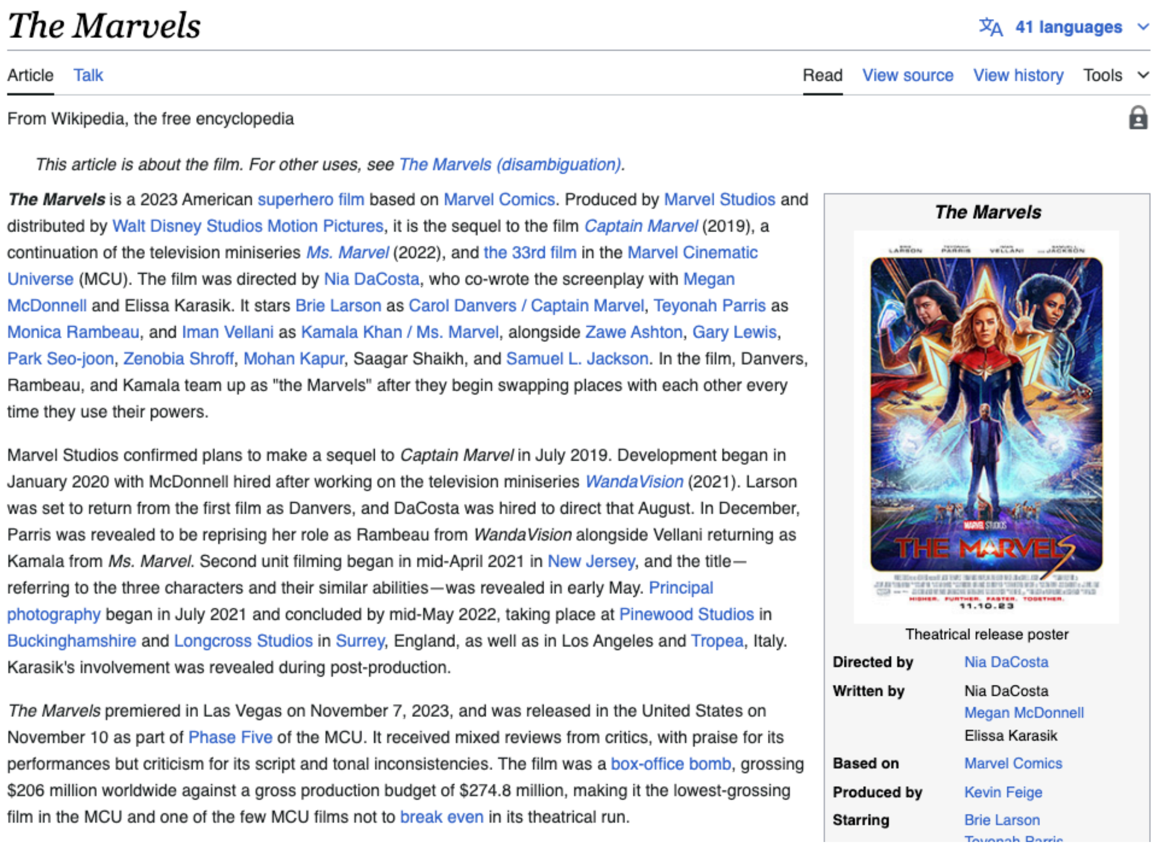
- Explicit features or characteristics of the item





Content of an item

- Explicit features or characteristics of the item
- Textual content: Title, description, tags
  - Exploit NLP techniques to extract content



The Marvels

41 languages

Article Talk Read View source View history Tools

From Wikipedia, the free encyclopedia

This article is about the film. For other uses, see The Marvels (disambiguation).

The Marvels is a 2023 American superhero film based on Marvel Comics. Produced by Marvel Studios and distributed by Walt Disney Studios Motion Pictures, it is the sequel to the film *Captain Marvel* (2019), a continuation of the television miniseries *Ms. Marvel* (2022), and the 33rd film in the Marvel Cinematic Universe (MCU). The film was directed by Nia DaCosta, who co-wrote the screenplay with Megan McDonnell and Elissa Karasik. It stars Brie Larson as Carol Danvers / Captain Marvel, Teyonah Parris as Monica Rambeau, and Iman Vellani as Kamala Khan / Ms. Marvel, alongside Zawe Ashton, Gary Lewis, Park Seo-joon, Zenobia Shroff, Mohan Kapur, Saagar Shaikh, and Samuel L. Jackson. In the film, Danvers, Rambeau, and Kamala team up as "the Marvels" after they begin swapping places with each other every time they use their powers.

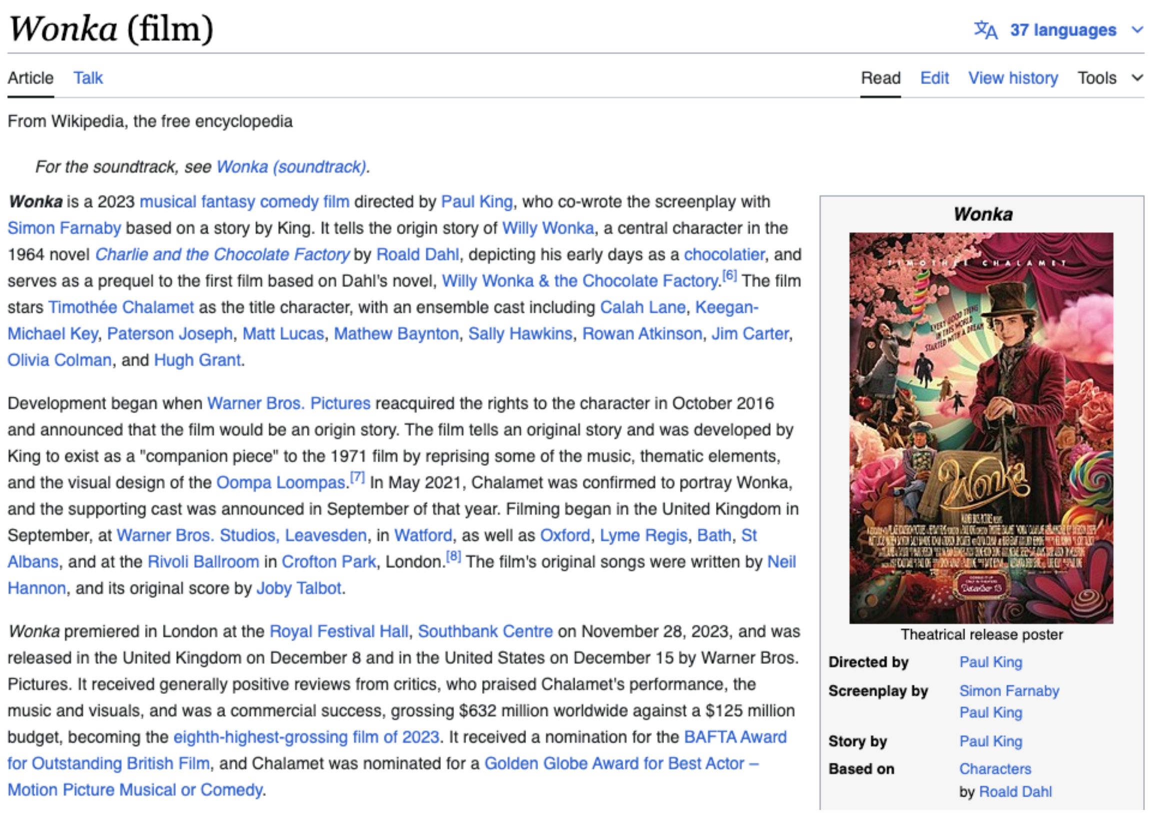
Marvel Studios confirmed plans to make a sequel to *Captain Marvel* in July 2019. Development began in January 2020 with McDonnell hired after working on the television miniseries *WandaVision* (2021). Larson was set to return from the first film as Danvers, and DaCosta was hired to direct that August. In December, Parris was revealed to be reprising her role as Rambeau from *WandaVision* alongside Vellani returning as Kamala from *Ms. Marvel*. Second unit filming began in mid-April 2021 in New Jersey, and the title—referring to the three characters and their similar abilities—was revealed in early May. Principal photography began in July 2021 and concluded by mid-May 2022, taking place at Pinewood Studios in Buckinghamshire and Longcross Studios in Surrey, England, as well as in Los Angeles and Tropea, Italy. Karasik's involvement was revealed during post-production.

The Marvels premiered in Las Vegas on November 7, 2023, and was released in the United States on November 10 as part of Phase Five of the MCU. It received mixed reviews from critics, with praise for its performances but criticism for its script and tonal inconsistencies. The film was a box-office bomb, grossing $206 million worldwide against a gross production budget of $274.8 million, making it the lowest-grossing film in the MCU and one of the few MCU films not to break even in its theatrical run.

The Marvels

Theatrical release poster

Directed by	Nia DaCosta
Written by	Nia DaCosta Megan McDonnell Elissa Karasik
Based on	Marvel Comics
Produced by	Kevin Feige
Starring	Brie Larson



Wonka (film)

37 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

For the soundtrack, see Wonka (soundtrack).

Wonka is a 2023 musical fantasy comedy film directed by Paul King, who co-wrote the screenplay with Simon Farnaby based on a story by King. It tells the origin story of Willy Wonka, a central character in the 1964 novel *Charlie and the Chocolate Factory* by Roald Dahl, depicting his early days as a chocolatier, and serves as a prequel to the first film based on Dahl's novel, Willy Wonka & the Chocolate Factory.[6] The film stars Timothée Chalamet as the title character, with an ensemble cast including Calah Lane, Keegan-Michael Key, Paterson Joseph, Matt Lucas, Mathew Baynton, Sally Hawkins, Rowan Atkinson, Jim Carter, Olivia Colman, and Hugh Grant.

Development began when Warner Bros. Pictures reacquired the rights to the character in October 2016 and announced that the film would be an origin story. The film tells an original story and was developed by King to exist as a "companion piece" to the 1971 film by reprising some of the music, thematic elements, and the visual design of the Oompa Loompas.[7] In May 2021, Chalamet was confirmed to portray Wonka, and the supporting cast was announced in September of that year. Filming began in the United Kingdom in September, at Warner Bros. Studios, Leavesden, in Watford, as well as Oxford, Lyme Regis, Bath, St Albans, and at the Rivoli Ballroom in Crofton Park, London.[8] The film's original songs were written by Neil Hannon, and its original score by Joby Talbot.

Wonka premiered in London at the Royal Festival Hall, Southbank Centre on November 28, 2023, and was released in the United Kingdom on December 8 and in the United States on December 15 by Warner Bros. Pictures. It received generally positive reviews from critics, who praised Chalamet's performance, the music and visuals, and was a commercial success, grossing $632 million worldwide against a $125 million budget, becoming the eighth-highest-grossing film of 2023. It received a nomination for the BAFTA Award for Outstanding British Film, and Chalamet was nominated for a Golden Globe Award for Best Actor – Motion Picture Musical or Comedy.

Wonka

Theatrical release poster

Directed by	Paul King
Screenplay by	Simon Farnaby Paul King
Story by	Paul King
Based on	Characters by Roald Dahl

Content of an item

- Explicit features or characteristics of the item
- Textual content: Title, description, tags
  - Exploit NLP techniques to extract content
- Features extracted from the signal (image, audio)

Content-based recommendations

- Pros:

- Cons:

Content-based recommendations

- Pros:
  - Item cold-start
    - Able to recommend new and unpopular items
- Cons:

Content-based recommendations

- Pros:
  - Item cold-start
    - Able to recommend new and unpopular items
  - Explainability
    - Able to provide the set of content features leading to an item being recommended
- Cons:

Content-based recommendations

- Pros:
  - Item cold-start
    - Able to recommend new and unpopular items
  - Explainability
    - Able to provide the set of content features leading to an item being recommended
- Cons:
  - Content availability
    - Performance tied to the quality and availability of content

Content-based recommendations

- Pros:
  - Item cold-start
    - Able to recommend new and unpopular items
  - Explainability
    - Able to provide the set of content features leading to an item being recommended
- Cons:
  - Content availability
    - Performance tied to the quality and availability of content
  - Overspecialization
    - Unable to provide novel or serendipitous recommendations

Content-based recommendations

- Pros:
  - Item cold-start
    - Able to recommend new and unpopular items
  - Explainability
    - Able to provide the set of content features leading to an item being recommended
- Cons:
  - Content availability
    - Performance tied to the quality and availability of content
  - Overspecialization
    - Unable to provide novel or serendipitous recommendations
  - User cold-start
    - Require gathering enough user interactions

Example of a content-based filtering approach

- Develop a content-based RS:
  - Content analyzer. Extract information about the content of items
  - User Profiler. Build user profiles
  - Item retriever. Filter items according to the similarity between profiles

Example of a content-based filtering approach

- Develop a content-based RS:
  - Content analyzer. Extract information about the content of items
    - Build item profiles
  - User Profiler. Build user profiles
  - Item retriever. Filter items according to the similarity between profiles

Example of a content-based filtering approach

- Develop a content-based RS:
  - Content analyzer. Extract information about the content of items
    - Build item profiles
  - User Profiler. Build user profiles
    - Build user profiles
  - Item retriever. Filter items according to the similarity between profiles

Example of a content-based filtering approach

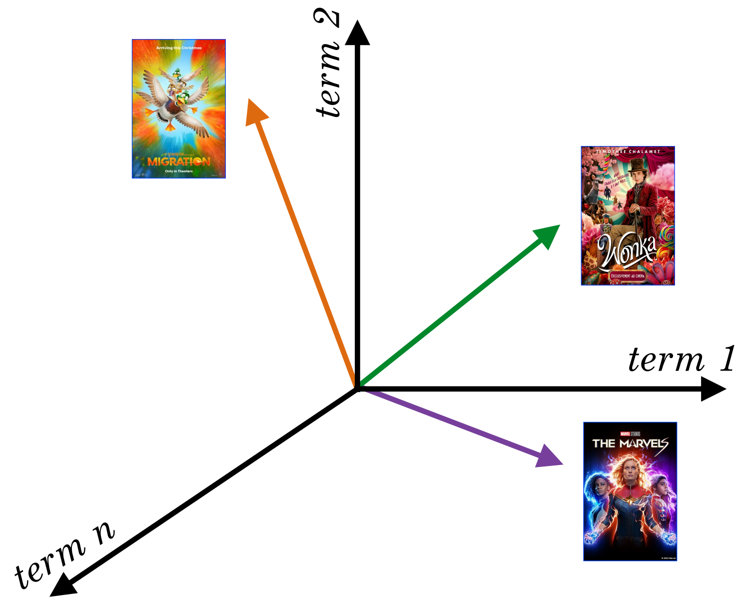
- Develop a content-based RS:
  - Content analyzer. Extract information about the content of items
    - Build item profiles
  - User Profiler. Build user profiles
    - Build user profiles
  - Item retriever. Filter items according to the similarity between profiles
    - Compute similarities between user and items

Building item profiles

- Exploit textual descriptions of items extracted from different sources
- Representation based on Vector Space Model (VSM)
  - Spatial representation of text documents

Building item profiles

- Exploit textual descriptions of items extracted from different sources
- Representation based on Vector Space Model (VSM)
  - Spatial representation of text documents



- Each dimension corresponds to a feature (word from the vocabulary)
  - Extracted using NLP techniques (normalization, tokenization, stemming)
- Each value corresponds to the term weight
  - Defined using *TF-IDF*

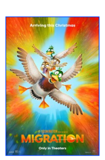
Building item profiles

Weighting terms is done with TF-IDF, the product of *Term Frequency (TF)* and *Inverse Document Frequency (IDF)*:

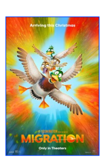
$$TF_{w,d_i} = \frac{f_{w,d_i}}{\max_{w' \in d_i} f_{w',d_i}}, \quad IDF_{w,d_i} = \log \frac{|\mathcal{I}|}{|\{i \in \mathcal{I} : w \in d_i\}|}$$

where f_{w,d_i} is the number of occurrences of the word w in document d_i.

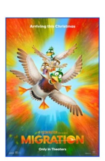
Building item profiles

	Barbie is a 2023 fantasy comedy film directed by Greta Gerwig from a screenplay she wrote with Noah Baumbach. It is the first live-action Barbie film after numerous animated films and specials.
	Anatomy of a Fall is a 2023 French legal drama film, directed by Justine Trier from a screeplay she wrote with Arthur Harari. It stars Sandra Hüller trying to prove her innocence in her husband's death.
	Wonka is a 2023 musical fantasy film directed by Paul King, who co-wrote the screenplay with Simon Farnaby based on a story by King.
	Poor Things is a 2023 film directed by Yorgos Lanthimos and written by Tony McNamara. It is a co-production between Ireland, the United Kingdom, and the United States.
	The Marvels is a 2023 American superhero film based on Marvel Comics. Produced by Marvel Studios and distributed by Walt Disney Studios Motion Pictures, it is the sequel to the film Captain Marvel.
	Migration is a 2023 American animated adventure comedy film produced by Universal Pictures and Illumination, and distributed by Universal.

Building item profiles

		TF(comedy)	IDF(comedy)	TF-IDF
	Barbie is a 2023 fantasy comedy film directed by Greta Gerwig from a screenplay she wrote with Noah Baumbach. It is the first live-action Barbie film after numerous animated films and specials.	1	$\log(6\ /\ 2)$	0.47
	Anatomy of a Fall is a 2023 French legal drama film, directed by Justine Trier from a screeplay she wrote with Arthur Harari. It stars Sandra Hüller trying to prove her innocence in her husband's death.	0	$\log(6\ /\ 2)$	0
	Wonka is a 2023 musical fantasy film directed by Paul King, who co-wrote the screenplay with Simon Farnaby based on a story by King.	0	$\log(6\ /\ 2)$	0
	Poor Things is a 2023 film directed by Yorgos Lanthimos and written by Tony McNamara. It is a co-production between Ireland, the United Kingdom, and the United States.	0	$\log(6\ /\ 2)$	0
	The Marvels is a 2023 American superhero film based on Marvel Comics. Produced by Marvel Studios and distributed by Walt Disney Studios Motion Pictures, it is the sequel to the film Captain Marvel.	0	$\log(6\ /\ 2)$	0
	Migration is a 2023 American animated adventure comedy film produced by Universal Pictures and Illumination, and distributed by Universal.	1	$\log(6\ /\ 2)$	0.47

Building item profiles

		TF(film)	IDF(film)	TF-IDF
	Barbie is a 2023 fantasy comedy film directed by Greta Gerwig from a screenplay she wrote with Noah Baumbach. It is the first live-action Barbie film after numerous animated films and specials.	3	$\log(6\ /\ 6)$	0
	Anatomy of a Fall is a 2023 French legal drama film, directed by Justine Trier from a screeplay she wrote with Arthur Harari. It stars Sandra Hüller trying to prove her innocence in her husband's death.	1	$\log(6\ /\ 6)$	0
	Wonka is a 2023 musical fantasy film directed by Paul King, who co-wrote the screenplay with Simon Farnaby based on a story by King.	1	$\log(6\ /\ 6)$	0
	Poor Things is a 2023 film directed by Yorgos Lanthimos and written by Tony McNamara. It is a co-production between Ireland, the United Kingdom, and the United States.	1	$\log(6\ /\ 6)$	0
	The Marvels is a 2023 American superhero film based on Marvel Comics. Produced by Marvel Studios and distributed by Walt Disney Studios Motion Pictures, it is the sequel to the film Captain Marvel.	2	$\log(6\ /\ 6)$	0
	Migration is a 2023 American animated adventure comedy film produced by Universal Pictures and Illumination, and distributed by Universal.	1	$\log(6\ /\ 6)$	0

Building user profiles and measuring similarities

- User profiles indicate the user interest in the various item dimensions.
  - They are created in the same space where items are represented.
  - An estimate can be obtained by aggregating the profiles of items rated.

Building user profiles and measuring similarities

- User profiles indicate the user interest in the various item dimensions.
  - They are created in the same space where items are represented.
  - An estimate can be obtained by aggregating the profiles of items rated.
- Predicting the user's interest in an item is done by computing the similarities between user and item profiles.
  - Using cosine similarity, for example.

Building user profiles and measuring similarities

- User profiles indicate the user interest in the various item dimensions.
  - They are created in the same space where items are represented.
  - An estimate can be obtained by aggregating the profiles of items rated.
- Predicting the user's interest in an item is done by computing the similarities between user and item profiles.
  - Using cosine similarity, for example.
- Other techniques can be used for content-based recommendation.
  - Bayesian classifiers, clustering, decision trees, neural networks.

Hybrid approaches

Hybrid approaches

- Collaborative filtering approaches
  - Leverage other users' interactions
  - Suffer from the cold-start problem

Hybrid approaches

- Collaborative filtering approaches
  - Leverage other users' interactions
  - Suffer from the cold-start problem
- Content-based filtering approaches
  - Leverage the content of items
  - Suffer from overspecialization

Hybrid approaches

- Collaborative filtering approaches
  - Leverage other users' interactions
  - Suffer from the cold-start problem
- Content-based filtering approaches
  - Leverage the content of items
  - Suffer from overspecialization
- Hybrid approaches try to get the best of both worlds
  - Leverage various sources of input at the same time
  - Use different recommendation approaches in one framework

Hybrid approaches

- Collaborative filtering approaches
  - Leverage other users' interactions
  - Suffer from the cold-start problem
- Content-based filtering approaches
  - Leverage the content of items
  - Suffer from overspecialization
- Hybrid approaches try to get the best of both worlds
  - Leverage various sources of input at the same time
  - Use different recommendation approaches in one framework

The best results are often achieved when different recommendation algorithms are combined in a single model.

Hybridization methods

- Weighted
- Switching
- Mixed
- Feature combination
- Cascade
- Feature augmentation

Robin Burke. *Hybrid Recommender Systems: Survey and Experiments.* UMUAI, 12(4), 2002

Weighted

- The score of a recommended item is computed by combining the scores from different recommendation techniques. Examples:
  - Linear combination of scores;
  - The output of each recommendation technique is treated as a set of votes, combined in a consensus scheme.

Robin Burke. *Hybrid Recommender Systems: Survey and Experiments.* UMUAI, 12(4), 2002

Weighted

- The score of a recommended item is computed by combining the scores from different recommendation techniques. Examples:
  - Linear combination of scores;
  - The output of each recommendation technique is treated as a set of votes, combined in a consensus scheme.
- Pros:
  - Straightforward way to do hybridization

Robin Burke. *Hybrid Recommender Systems: Survey and Experiments.* UMUAI, 12(4), 2002

Weighted

- The score of a recommended item is computed by combining the scores from different recommendation techniques. Examples:
  - Linear combination of scores;
  - The output of each recommendation technique is treated as a set of votes, combined in a consensus scheme.
- Pros:
  - Straightforward way to do hybridization
- Cons:
  - Assumption: The relative value of the different techniques is more or less uniform across the space of possible items
    - Generally not true; a collaborative recommender will be weaker for items with a small number of ratings

Robin Burke. *Hybrid Recommender Systems: Survey and Experiments.* UMUAI, 12(4), 2002

Switching

- The RS switches between different recommendation techniques.

  Example:
  - Use first a content-based recommendation method to generate recommendations;
  - If the content-based method cannot provide recommendations with sufficient confidence, switch to a collaborative filtering method;
  - Both methods suffer from the user cold-start problem.

Robin Burke. *Hybrid Recommender Systems: Survey and Experiments.* UMUAI, 12(4), 2002

Switching

- The RS switches between different recommendation techniques.

  Example:
  - Use first a content-based recommendation method to generate recommendations;
  - If the content-based method cannot provide recommendations with sufficient confidence, switch to a collaborative filtering method;
  - Both methods suffer from the user cold-start problem.
- Pros:
  - Benefit from the strengths of the constituent recommenders

Robin Burke. *Hybrid Recommender Systems: Survey and Experiments.* UMUAI, 12(4), 2002

Switching

- The RS switches between different recommendation techniques.

  Example:
  - Use first a content-based recommendation method to generate recommendations;
  - If the content-based method cannot provide recommendations with sufficient confidence, switch to a collaborative filtering method;
  - Both methods suffer from the user cold-start problem.
- Pros:
  - Benefit from the strengths of the constituent recommenders
- Cons:
  - Introduce additional complexity related to the definition of the switching criterion

Robin Burke. *Hybrid Recommender Systems: Survey and Experiments.* UMUAI, 12(4), 2002

Mixed

- Recommendations from more than one technique are presented together.

Robin Burke. *Hybrid Recommender Systems: Survey and Experiments.* UMUAI, 12(4), 2002

Feature combination

- Features from different recommendation data sources are fed to a single recommendation approach.

Robin Burke. *Hybrid Recommender Systems: Survey and Experiments.* UMUAI, 12(4), 2002

Feature combination

- Features from different recommendation data sources are fed to a single recommendation approach.

  Examples:

  - Treat collaborative information, *i.e.*, user interactions, as additional features associated with each observation and use content-based techniques on the augmented dataset;
  - Treat content features as additional features in the collaborative setting, *i.e.*, as additional feedback from virtual users.

Robin Burke. *Hybrid Recommender Systems: Survey and Experiments.* UMUAI, 12(4), 2002

Cascade

- One recommendation technique is used first to produce a coarse ranking of items and another technique refines the recommendations from the candidate set.

Robin Burke. *Hybrid Recommender Systems: Survey and Experiments.* UMUAI, 12(4), 2002

Cascade

- One recommendation technique is used first to produce a coarse ranking of items and another technique refines the recommendations from the candidate set.
- The system may decide to avoid using the second, lower-priority, technique on items that are already well-differentiated by the first.

Robin Burke. *Hybrid Recommender Systems: Survey and Experiments*. UMUAI, 12(4), 2002

Cascade

- One recommendation technique is used first to produce a coarse ranking of items and another technique refines the recommendations from the candidate set.
- The system may decide to avoid using the second, lower-priority, technique on items that are already well-differentiated by the first.
  - More efficient than other hybrid approaches that require the application of all of its techniques on all the items.

Robin Burke. *Hybrid Recommender Systems: Survey and Experiments.* UMUAI, 12(4), 2002

Cascade

- One recommendation technique is used first to produce a coarse ranking of items and another technique refines the recommendations from the candidate set.
- The system may decide to avoid using the second, lower-priority, technique on items that are already well-differentiated by the first.
  - More efficient than other hybrid approaches that require the application of all of its techniques on all the items.
  - Tolerant of noise induced by the low-priority technique that can only refine recommendations.

Robin Burke. *Hybrid Recommender Systems: Survey and Experiments.* UMUAI, 12(4), 2002

Cascade

- One recommendation technique is used first to produce a coarse ranking of items and another technique refines the recommendations from the candidate set.
- The system may decide to avoid using the second, lower-priority, technique on items that are already well-differentiated by the first.
  - More efficient than other hybrid approaches that require the application of all of its techniques on all the items.
  - Tolerant of noise induced by the low-priority technique that can only refine recommendations.
  - Requires a meaningful ordering of techniques.

Robin Burke. *Hybrid Recommender Systems: Survey and Experiments*. UMUAI, 12(4), 2002

Feature augmentation

- One technique is used to predict a rating or classify an item, and that information is incorporated into another recommendation technique.

Robin Burke. *Hybrid Recommender Systems: Survey and Experiments*. UMUAI, 12(4), 2002

Feature augmentation

- One technique is used to predict a rating or classify an item, and that information is incorporated into another recommendation technique.

  Example:

  - Make content-based recommendations of items based on textual data;
  - Include in the textual data "related titles" and "related authors" information from Amazon, generated based on its internal collaborative system.

Robin Burke. *Hybrid Recommender Systems: Survey and Experiments.* UMUAI, 12(4), 2002

Feature augmentation

- One technique is used to predict a rating or classify an item, and that information is incorporated into another recommendation technique.

  Example:
  - Make content-based recommendations of items based on textual data;
  - Include in the textual data "related titles" and "related authors" information from Amazon, generated based on its internal collaborative system.

- Similar to the feature combination method:
  - In feature augmentation, the output is used for another RS;
  - In feature combination, the representations used by two systems are combined.

Robin Burke. *Hybrid Recommender Systems: Survey and Experiments.* UMUAI, 12(4), 2002

Hybridization methods

- Weighted
- Switching
- Mixed
- Feature combination
- Cascade
- Feature augmentation

Robin Burke. *Hybrid Recommender Systems: Survey and Experiments*. UMUAI, 12(4), 2002

Context-aware recommendation

Contextual information

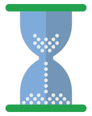
- User choices are not solely guided by a fixed set of preferences related to item features, but strongly depends on their context, their current needs, and the situation they are in.

Contextual information

- User choices are not solely guided by a fixed set of preferences related to item features, but strongly depends on their context, their current needs, and the situation they are in.

Example: Recommendation in the tourism domain

Location



Time

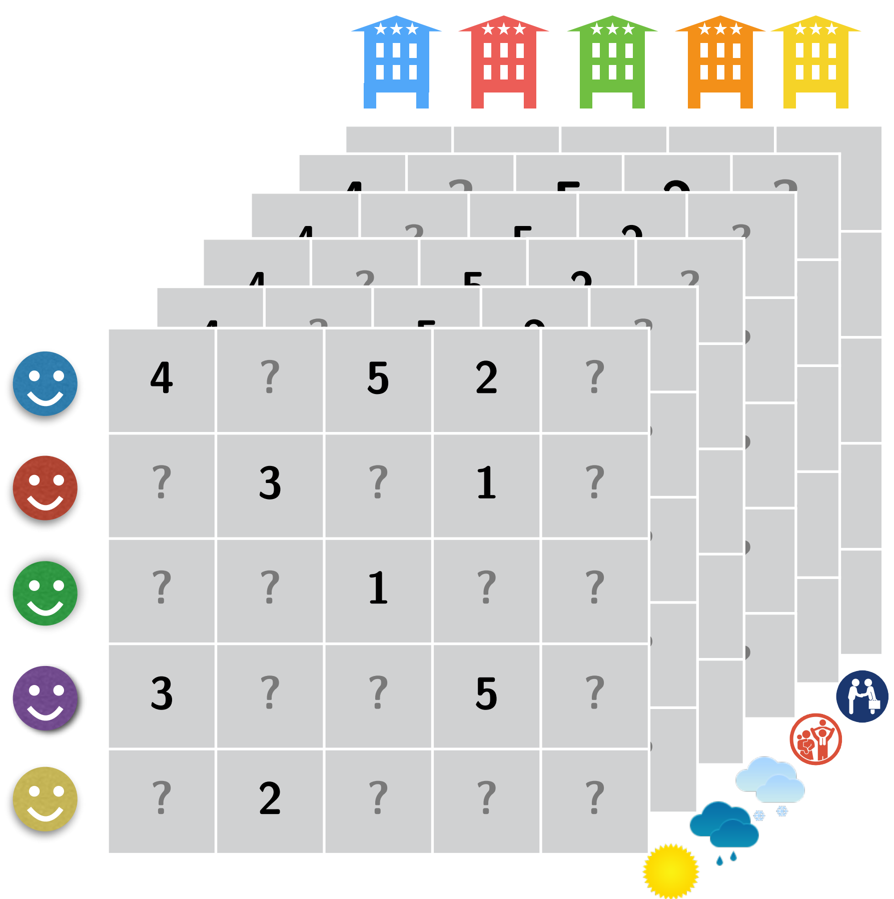
Trips' intent

Events

Weather

Context-aware recommendations

- Additional dimensions besides the user and item ones are considered.
- The representation space of feedback becomes multidimensional.
- The utility function is defined on the space $\mathcal{U} \times \mathcal{I} \times \mathcal{C}$ where $\mathcal{C}$ is the set of relevant contextual factors.



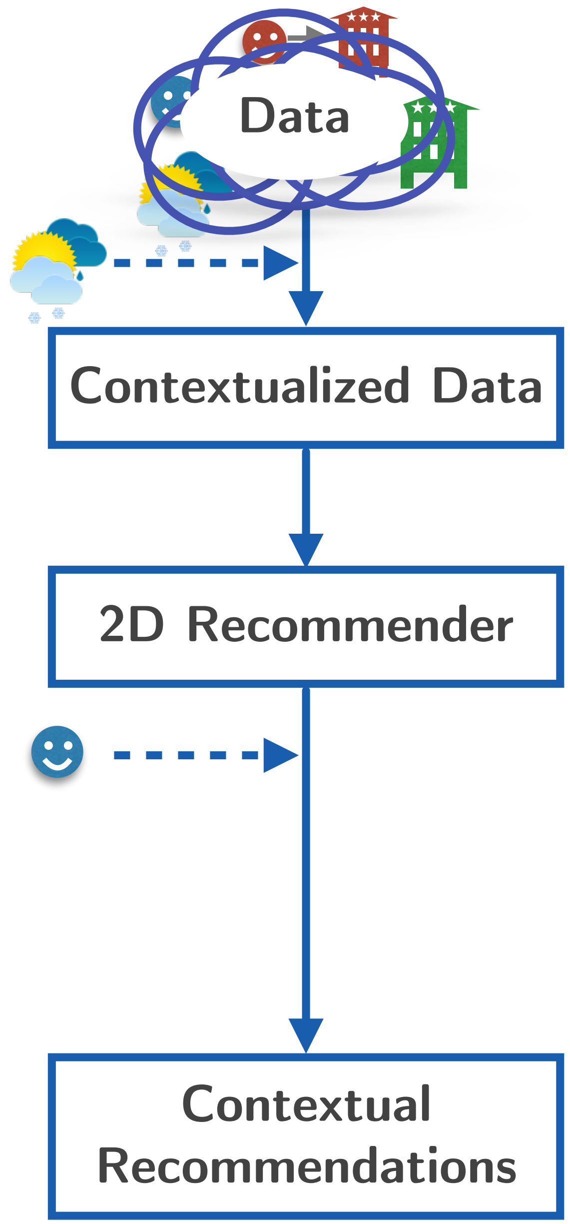
Paradigms for context-aware recommendation

- Contextual pre-filtering
  - Context is used for input data selection
- Contextual post-filtering
  - Context is used to filter recommendations
- Contextual modeling
  - Context is directly incorporated into the model

Gediminas Adomavicius and Alexander Tuzhilin. *Context-Aware Recommender System.* In Recommender Systems Handbook. 2015.

Contextual pre-filtering

- Context drives the selection of data given as input to the recommendation method.



Contextual pre-filtering

- Context drives the selection of data given as input to the recommendation method.

  Examples:

  - User splitting approach
    - Users are split into several sub-profiles, each of them representing the user behavior in a particular context.

User	Item	Rating	Season	Location	Companion
u_1	i_1	5	Summer	France	Family
u_1	i_1	1	Winter	France	Friend
u_1	i_1	2	Summer	Canada	Partner

User	Item	Rating
u_{11}	i_1	5
u_{12}	i_1	1
u_{13}	i_1	2

Contextual pre-filtering

- Context drives the selection of data given as input to the recommendation method.

  Examples:

  - Item splitting approach
    - Items are split into several fictitious items, according to the contexts in which they were selected.

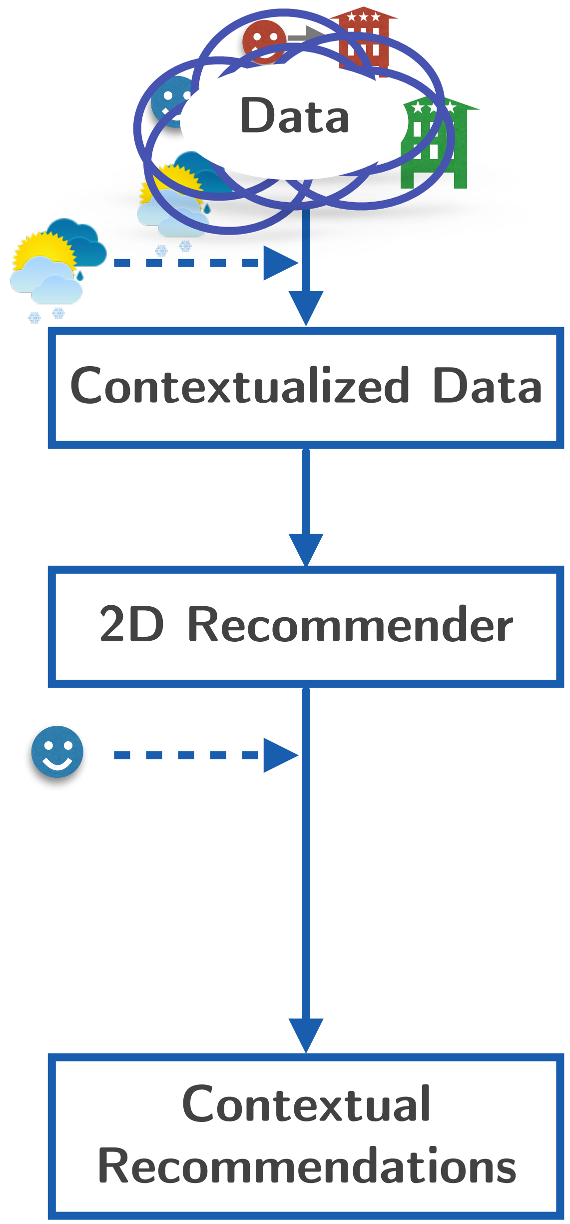
User	Item	Rating	Season	Location	Companion
u_1	i_1	5	Summer	France	Family
u_1	i_1	1	Winter	France	Friend
u_1	i_1	2	Summer	Canada	Partner

➡

User	Item	Rating
u_1	i_{11}	5
u_1	i_{12}	1
u_1	i_{13}	2

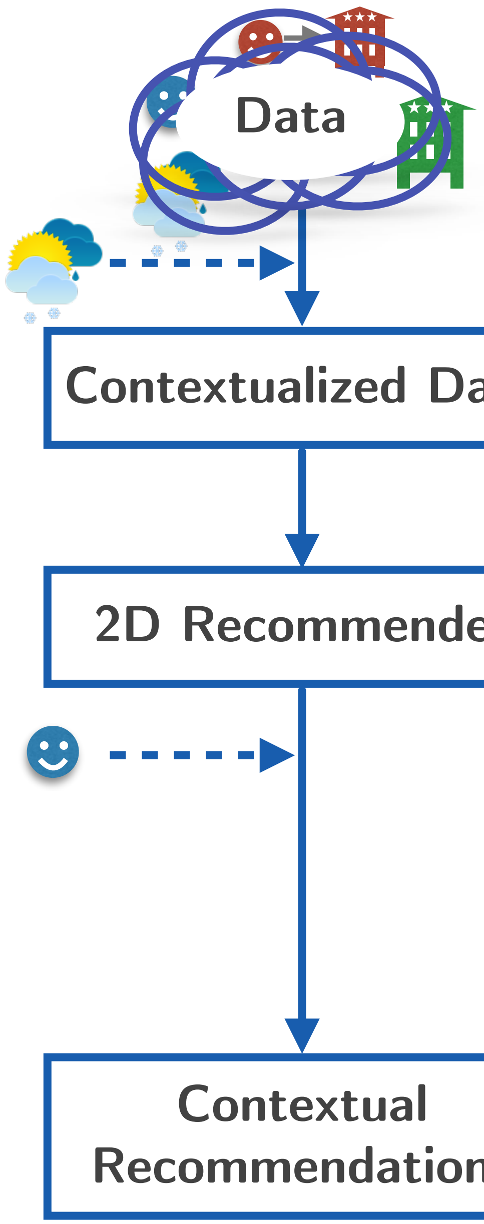
Contextual pre-filtering

- Context drives the selection of data given as input to the recommendation method.
- Pros: The contextual information is incorporated into user and item dimensions.



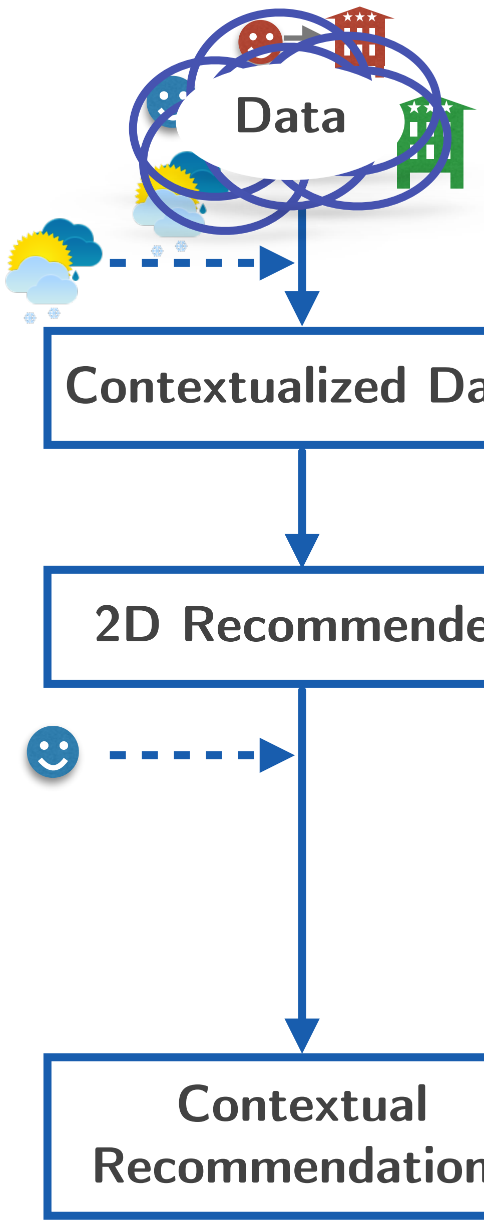
Contextual pre-filtering

- Context drives the selection of data given as input to the recommendation method.
- Pros: The contextual information is incorporated into user and item dimensions.
- Cons: Creating new users and items increases sparsity.



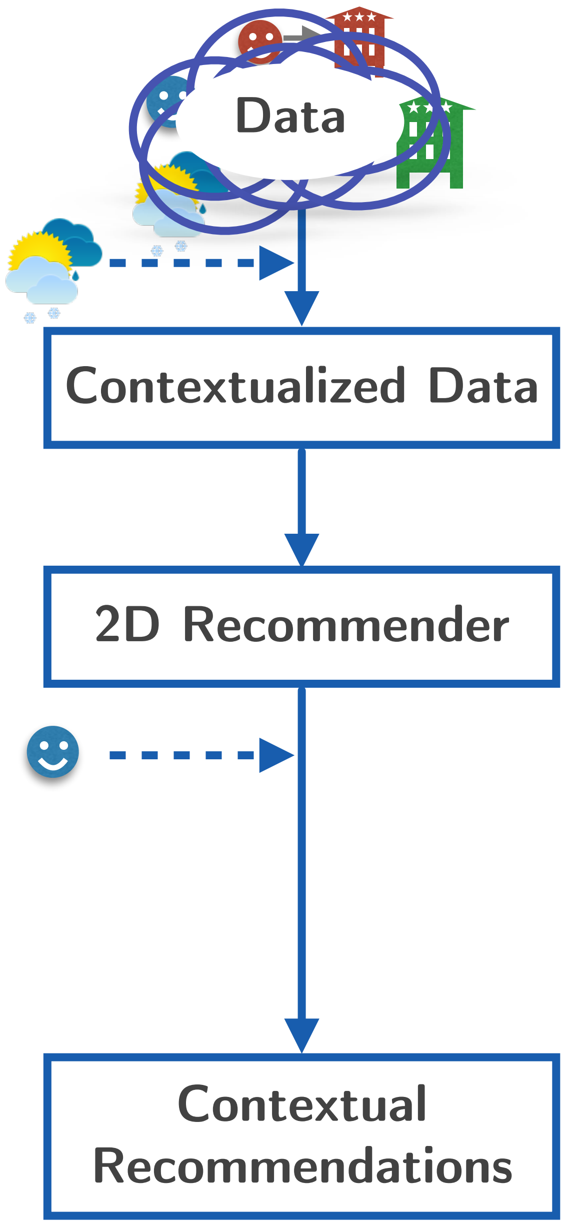
Contextual pre-filtering

- Context drives the selection of data given as input to the recommendation method.
- Pros: The contextual information is incorporated into user and item dimensions.
- Cons: Creating new users and items increases sparsity.
  - The exact context may be too narrow, and certain of its aspects may be insignificant, with only a few observations to learn from.



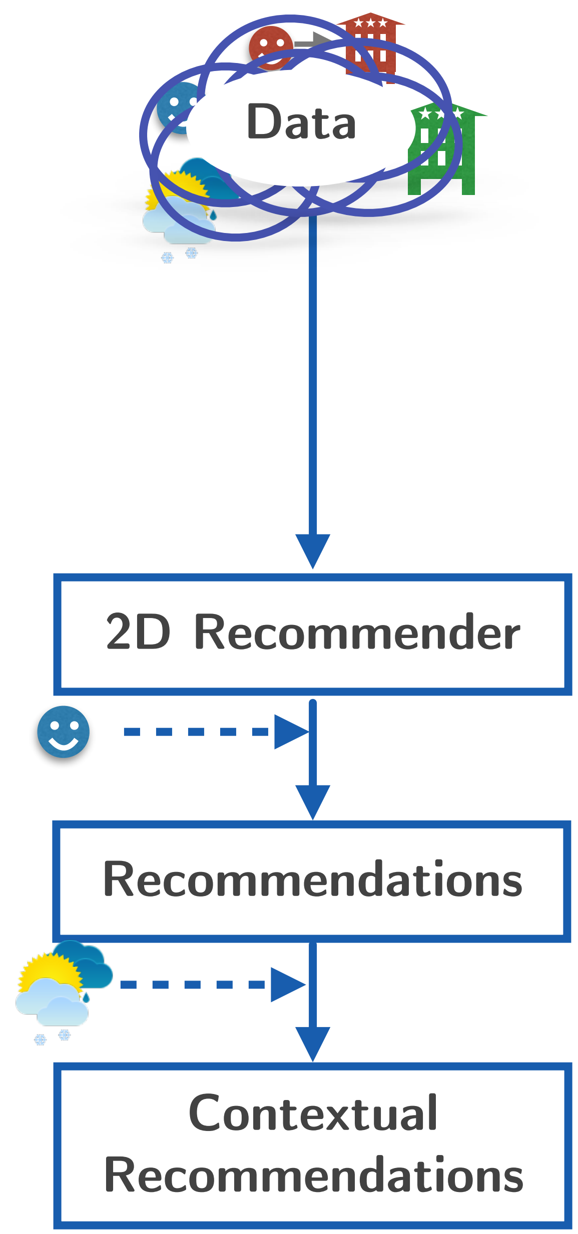
Contextual pre-filtering

- Context drives the selection of data given as input to the recommendation method.
- Pros: The contextual information is incorporated into user and item dimensions.
- Cons: Creating new users and items increases sparsity.
  - The exact context may be too narrow, and certain of its aspects may be insignificant, with only a few observations to learn from.
    - Generalize context, using for example higher levels concepts in context hierarchies.



Contextual post-filtering

- Context is used to refine the recommendation list, generating by a recommendation approach ignoring the context.

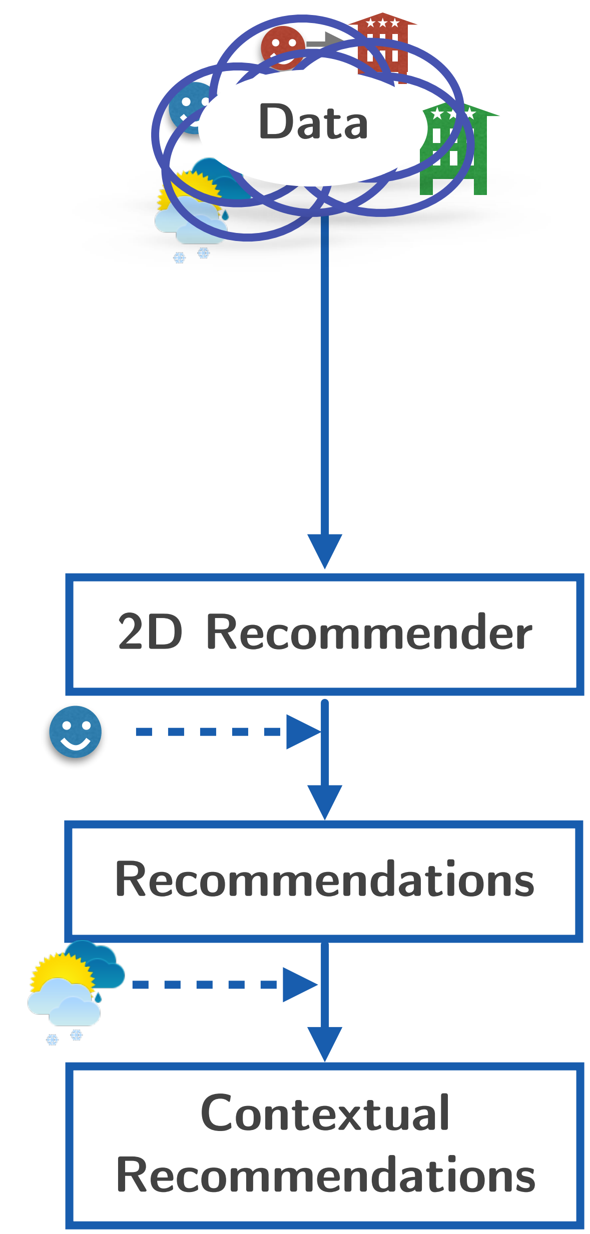


Contextual post-filtering

- Context is used to refine the recommendation list, generating by a recommendation approach ignoring the context.

  Examples:

  - Remove recommendations that are irrelevant to the context;
  - Adjust the ranking of items using the context.



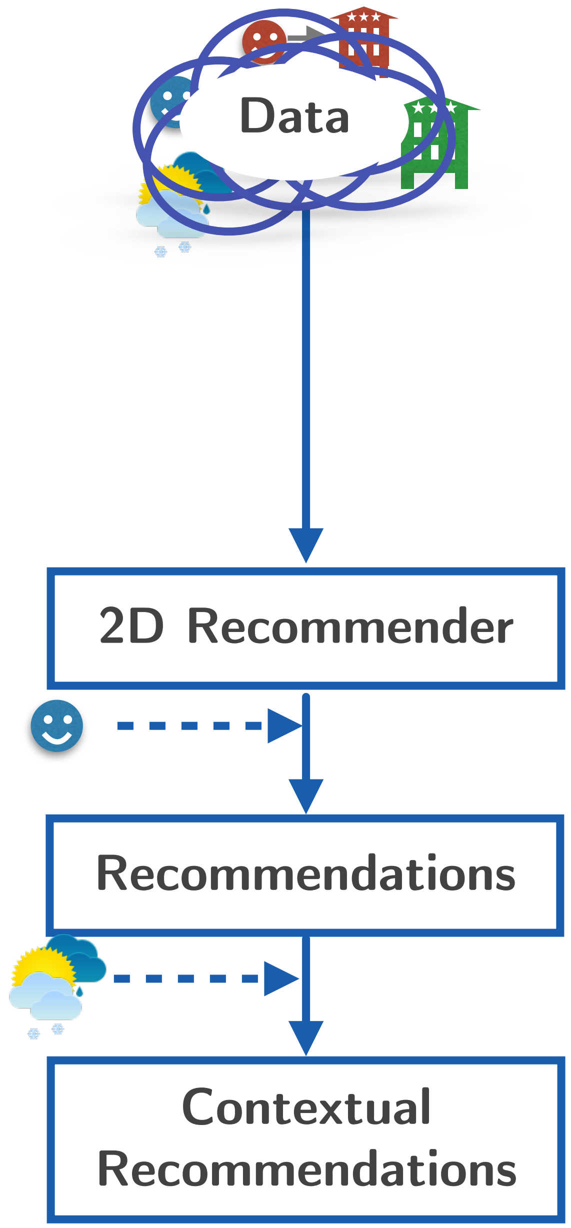
Contextual post-filtering

- Context is used to refine the recommendation list, generating by a recommendation approach ignoring the context.

  Examples:

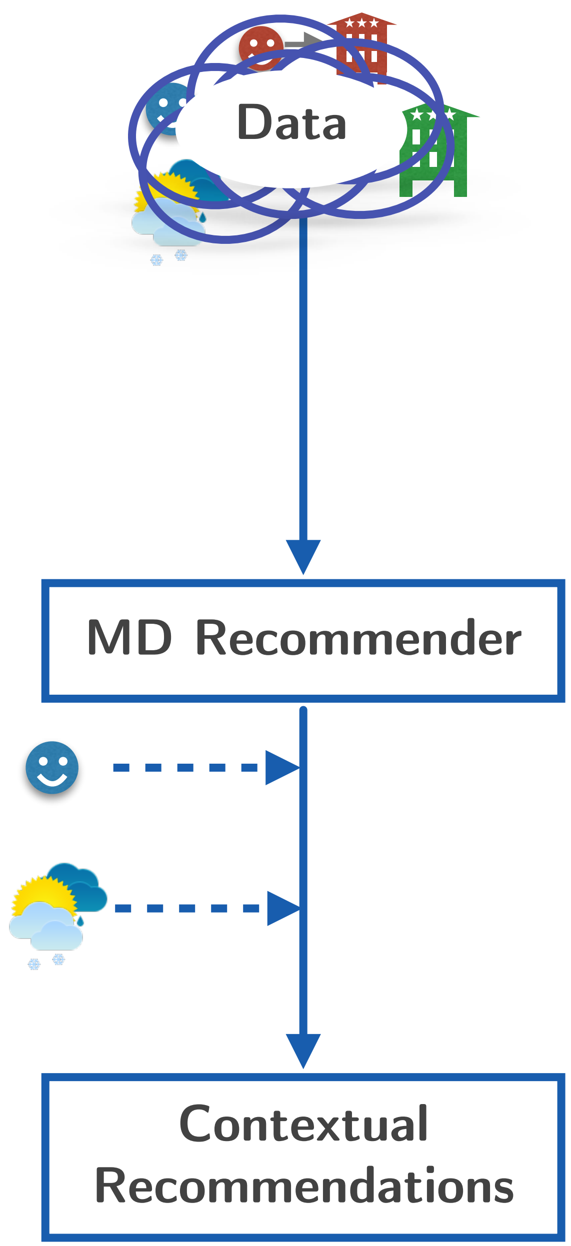
  - Remove recommendations that are irrelevant to the context;
  - Adjust the ranking of items using the context.

- Pros of contextual pre-filtering and post-filtering:
  - Ability to leverage traditional recommendation algorithms.



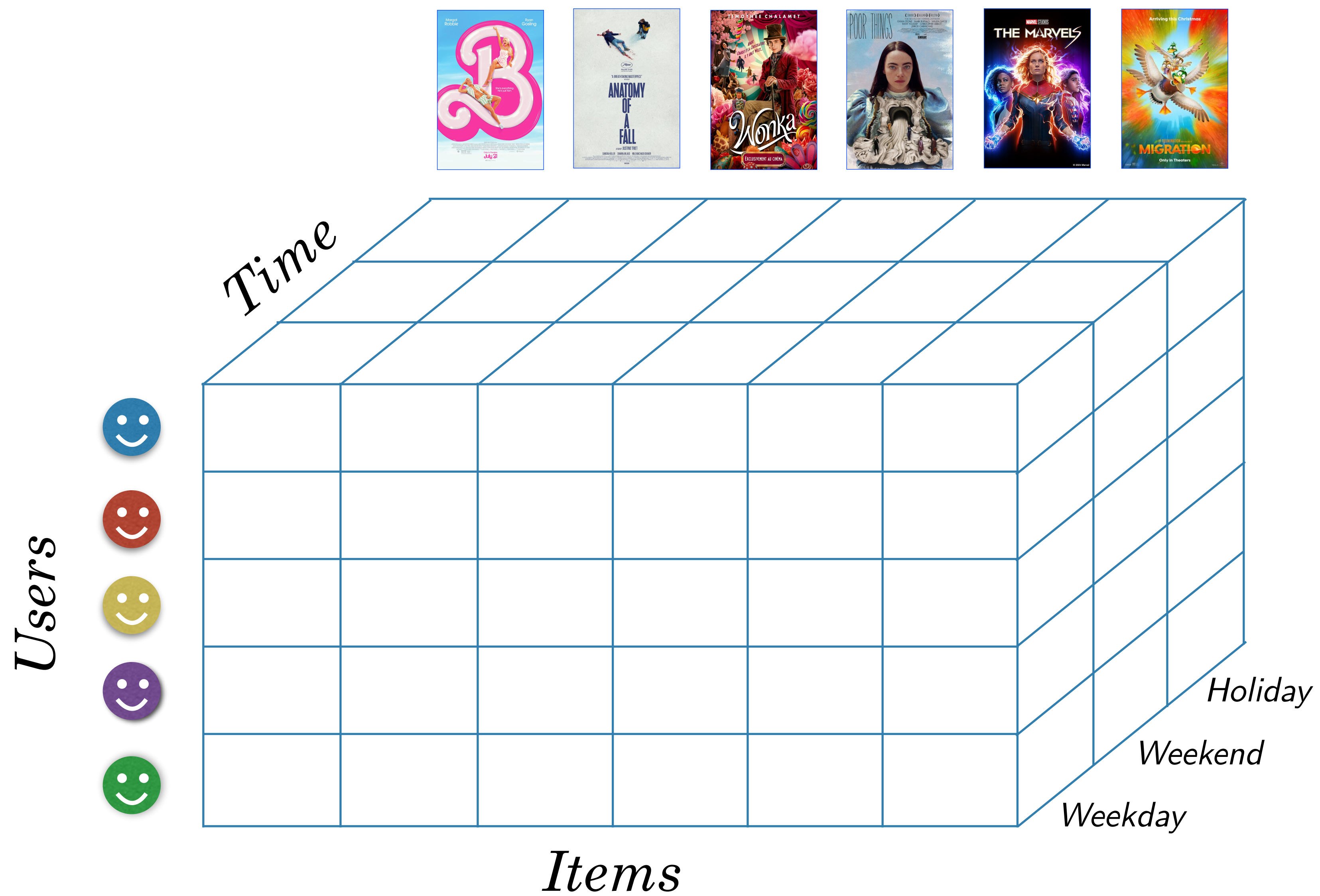
Contextual modeling

- Context is used directly in the model learning phase.
- Contextual variables are added as additional dimensions in the feature space, alongside the *user* and *item* dimensions.

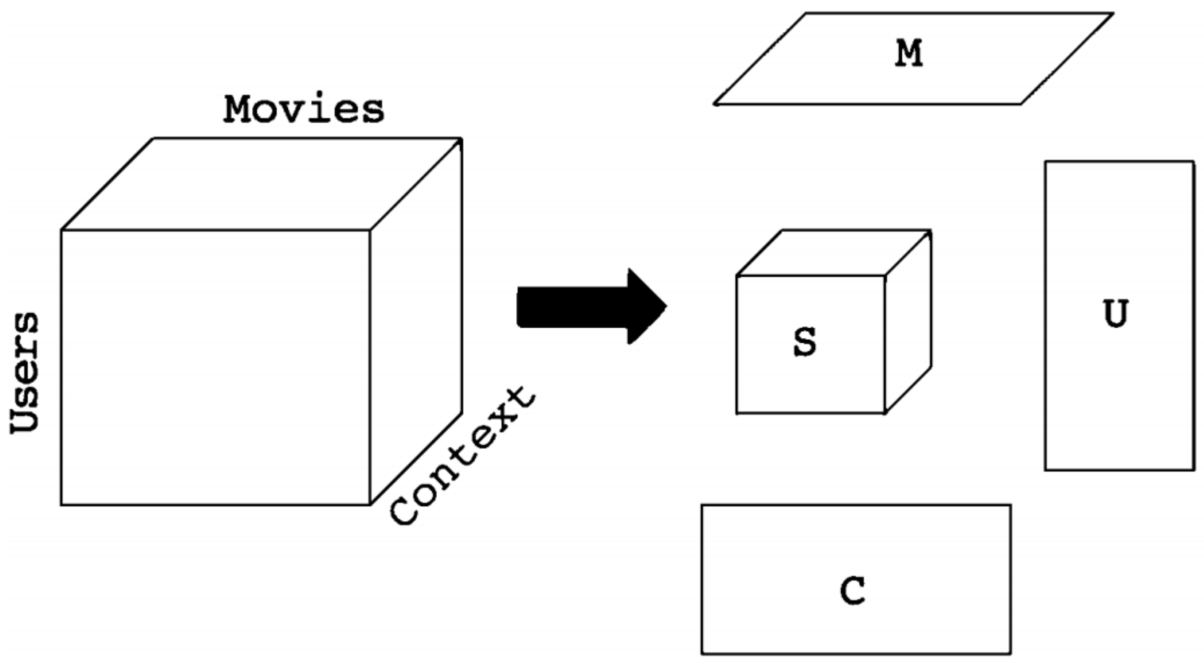


Two-dimensional model

N-dimensional model



Tensor factorization



$$F_{ijk} = S \times_U U_{i*} \times_M M_{j*} \times_C C_{k*}$$

Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. *Multiverse Recommendation: N-dimensional Tensor Factorization for Context-aware Collaborative Filtering*. In Proc. RecSys, 2010.

Factorization Machines

Feature vector **x** | Target y

	User				Movie					Other Movies rated					Time	Last Movie rated					Target y	
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...	Time	TI	NH	SW	ST	...		
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	$y^{(1)}$
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3	$y^{(2)}$
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1	$y^{(2)}$
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4	$y^{(3)}$
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5	$y^{(4)}$
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1	$y^{(5)}$
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5	$y^{(6)}$

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i \, x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle \, x_i \, x_j$$

Steffen Rendle. *Factorization Machines.* In Proc. ICDM, 2010.

Conclusion