Recommender Systems (RS)
An Overview

MEI

CONSTANTINO MARTINS, CATARINA FIGUEIREDO, DULCE MOTA AND FÁTIMA RODRIGUES

Disclaimer

- **Some of this material/slides are adapted from several:**
  - Presentations found on the internet;
  - Papers
  - Books;
  - Web sites
  - …

RS Problem/Goal

Which computer should I buy?

What is the best city to live for me and my family?

Which music should I by?

Which movie should I rent?

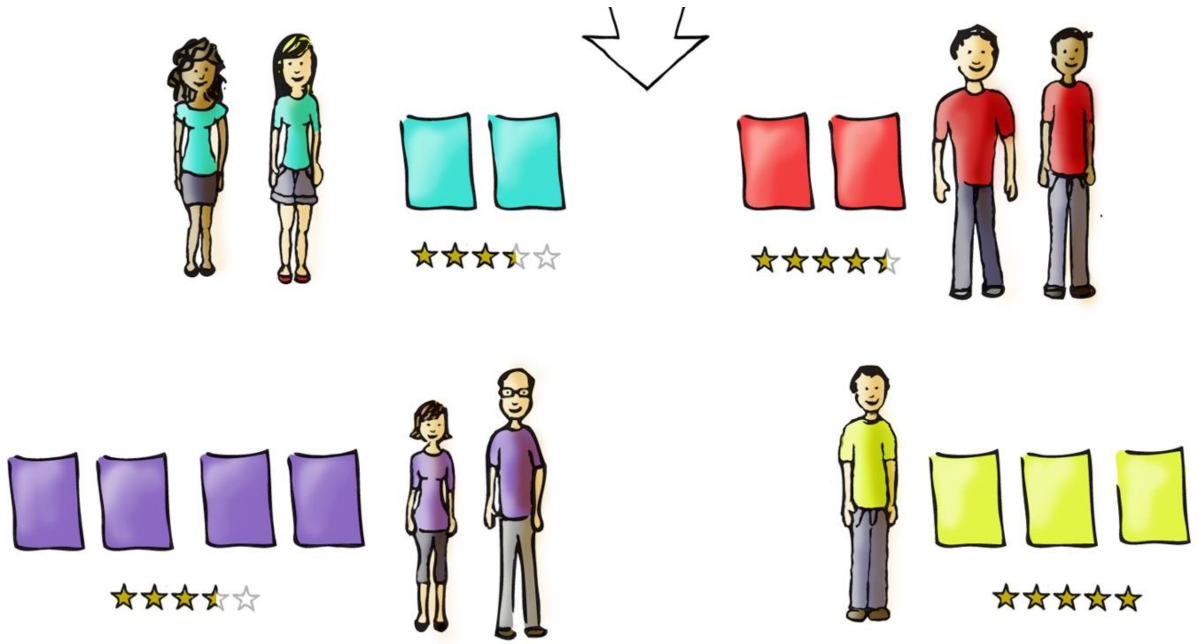
Which web social sites will I find interesting people?

Which course and university are the best for my future?

Basic principles that guide RS

"Tell me who you walk with, and I'll tell you who you are."

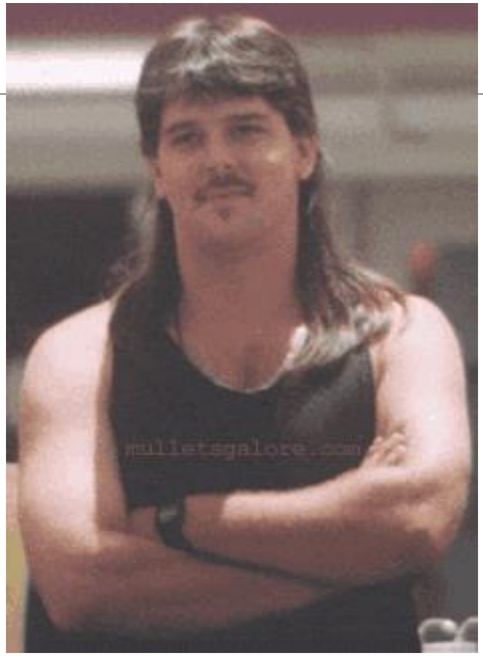
"What is relevant to me should also be relevant to someone with similar interests."



Basic principles that guide RS

“Tell me who you walk with, and I'll tell you who you are.

"What is relevant to me should also be relevant to someone with similar interests."



Customer A

- Buys Metalica music
- Buys Megadeth music
- Buys Moonspell music

Customer B

- Does search on Metalica and Megadeth
- Recommender system suggests Moonspell from data collected from customer A

RS Definitions?

- ”The goal of recommender systems is to suggest items or ideas that align with a user's specific needs or way of thinking.”
- “These systems aim is to provide personalized recommendations based on individual preferences.”
- “Recommender systems try to automate aspects of a completely different information discovery model where people try to find other people (or items) with similar tastes and then ask them to suggest new things.”

RS Definition?

- A RS can be defined as a collection of different techniques, used by systems to filter and organise their items, in order to select either the best or the most suitable ones for presentation, according to the user tastes (Kobsa, 1994)
- **The general term "item" is used to express what is recommended to the users**. (Thorat et al., 2015)
- RS try to model the relationship between users and items, representing the user items preferences. Suggest items that suit the needs of a customer and help him select and buy items from a broad range of choices, is the RS goal. With the growth of on-line services using RSs, such as Amazon, Yahoo! Music and Netflix, this booming machine learning subfield, spread in the late 1990s. (Takács et al., 2009)
- Recommender systems are systems that help users discover items they may like

RS - Overview

- Recommender systems have their roots in various scientific/research areas, such as:
  - Information retrieval,
  - information filtering, and
  - text classification
- Recommender systems apply methods from different fields, such as:
  - machine learning,
  - data mining, and
  - knowledge-based systems
- Addressed main topics:
  - Basic recommendation algorithms
  - Knowledge-based and hybrid approaches
  - Evaluation of recommender systems and their business value
  - ..

Non-personalized RS

- Non-personalized recommendation systems are systems that provide recommendations without taking into account individual preferences or specific user characteristics
- These systems typically use simpler and more straightforward approaches to suggest items or content to all users, without personalized adaptation
- Use the population behaviour of a whole in order to infer what the user might like
- Even if we had never visited a specific website, it's expected that the user receive recommendations Items recommended to you are the same as what's recommended to others

Examples of types of Non-personalized RS

- Popular/Trending: Recommends the most popular or trending items based on global metrics such as overall popularity, sales, or views
- New Arrivals: Suggests the most recently added items to the catalog, regardless of individual user preferences
- Random Recommendation: Provides randomly chosen recommendations from the entire set of available items
- Category-Based Recommendation: Recommends items based on general categories or characteristics without considering the user's interaction history
- Special Offers: Recommendations based on promotions, discounts, or special offers, without considering individual preferences

Non-personalized RS

- They are easier to implement and do not require a significant amount of individual user data
- They are useful in situations where individual preferences are unknown or challenging to obtain
- They typically do not provide the same level of personalized relevance as personalized systems that take into account each user's specific behavior and preferences

Does it make sense to combine non-personalized and collaborative recommendation systems (or another) in a hybrid model?

Main RS Approach

- Collaborative/Social-filtering: aggregation of consumers' preferences and recommendations to other users based on similarity in behavioral patterns
- Content-based Filtering: supervised machine learning used to induce a classifier to discriminate between interesting and uninteresting items for the user
- Knowledge-based system: knowledge about users and products used to reason what meets the user's requirements, using discrimination tree, decision support tools, case-based reasoning (CBR),…

Collaborative/Social-filtering

- CF (also called social-filtering) is one of the currently most used techniques and was greatly influenced by the Web 2.0 phenomena (”social web”)
- It relies on other users’ information to recommend the current user items and is the process of filtering or evaluating items using the opinions of other people
- The most similar users found will then be the source of new recommendation items, using the theory that, if a user is similar to others, then their tastes/necessities etc will also be similar

Collaborative/Social-filtering

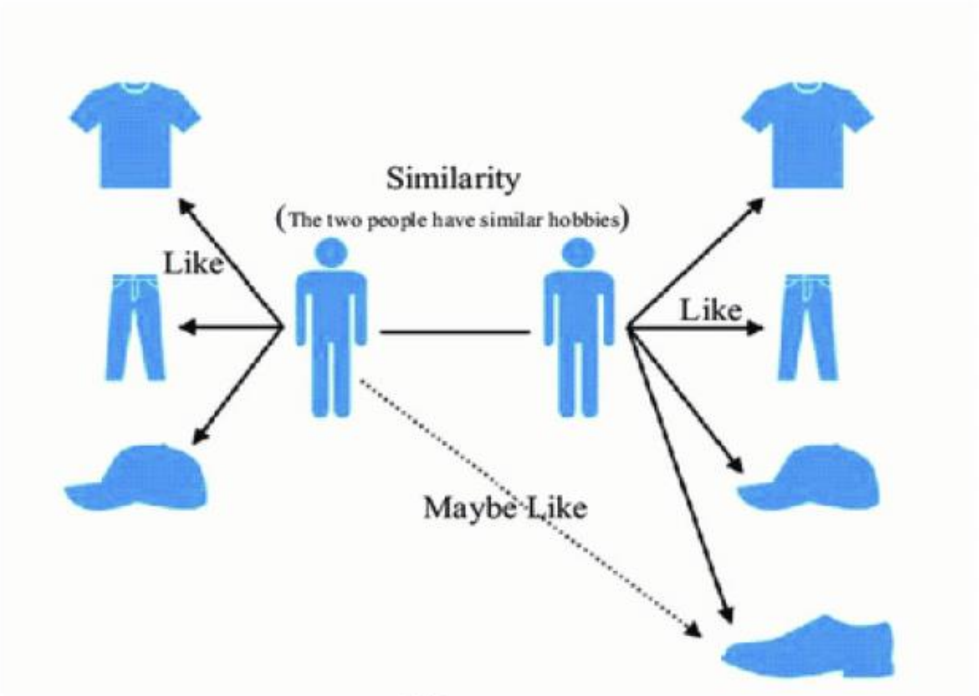
- Other information fundamental for this technique are the top-chosen items, user reviews or ratings on several types of items, among others
- An example of this technique's procedure might be: the system looks for users with the same personality characteristics as the current user, like gender, age, preferred genre, among others, and then searches for items viewed by those users, that the current user has not see

Collaborative/Social-filtering

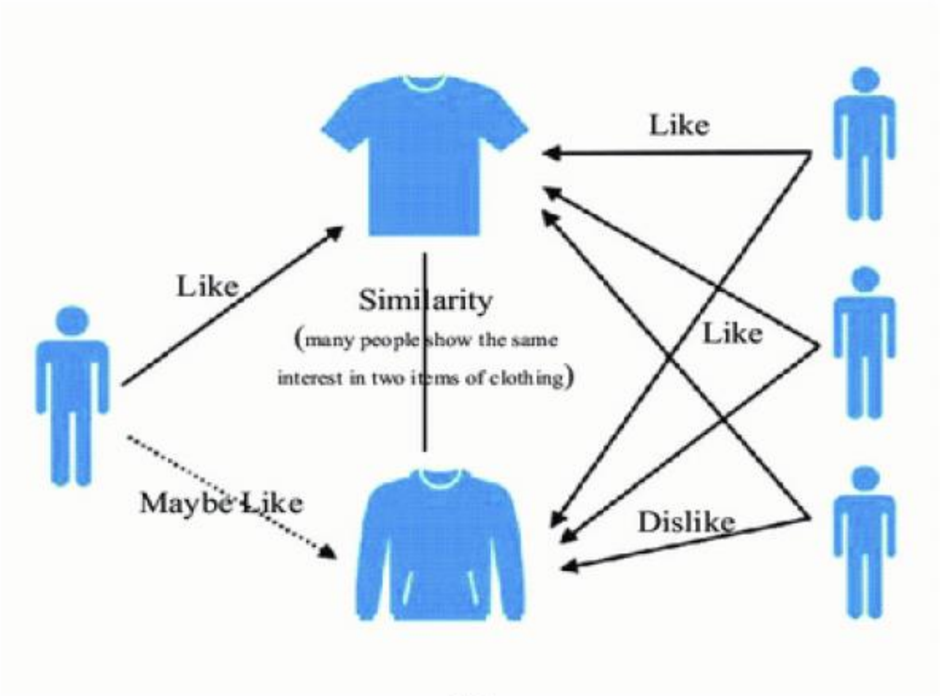
- When the system prioritizes users with similar preferences, it falls under the subtype known as **User-based Collaborative Filtering**. In this case, the similarity function can consider various user attributes such as age, gender, profession, hobbies, and more. This approach aims to identify users with comparable profiles and preferences, enhancing the relevance of recommendations (Fig a, next slide)

- Conversely, if the system directs its focus toward products, it aligns with the subtype called **Item-based Collaborative Filtering**. Here, the similarity function considers products that are likely to be deemed similar if other users who have common purchases have bought them together. This strategy relies on the idea that items frequently bought together are likely to share similarities (Fig b, next slide)

Collaborative/Social-filtering

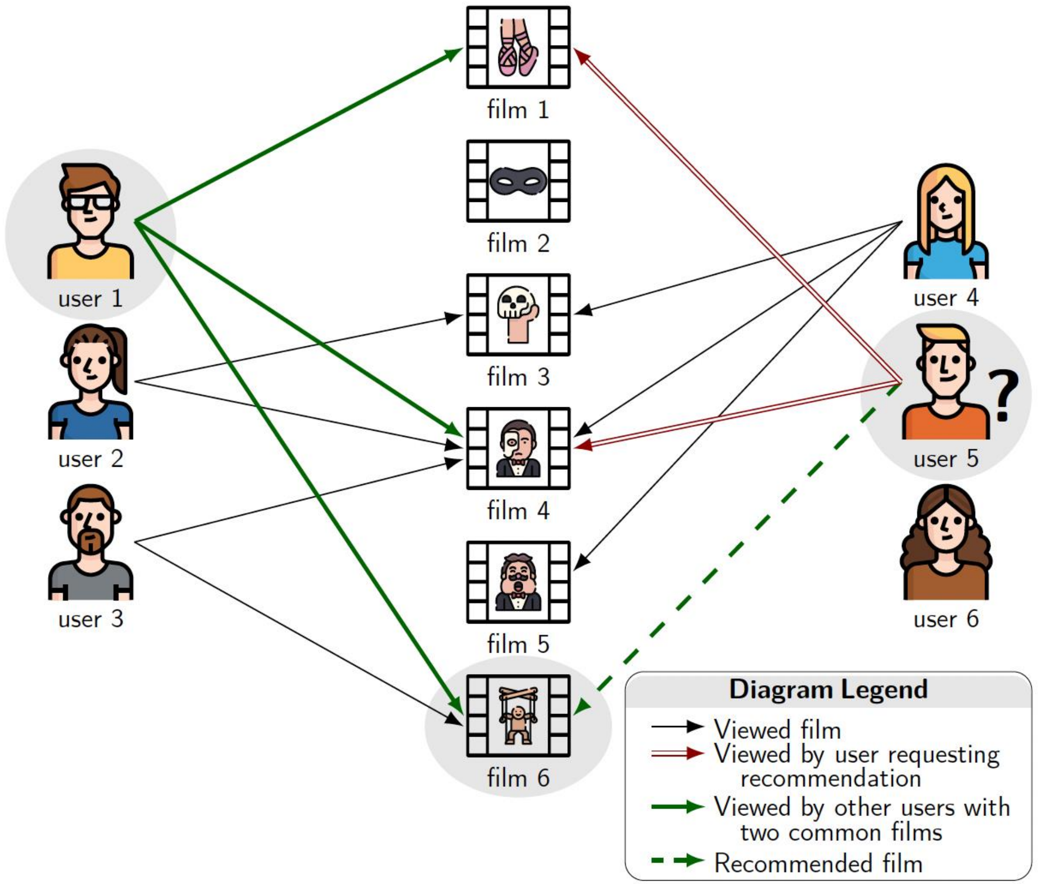


(a)

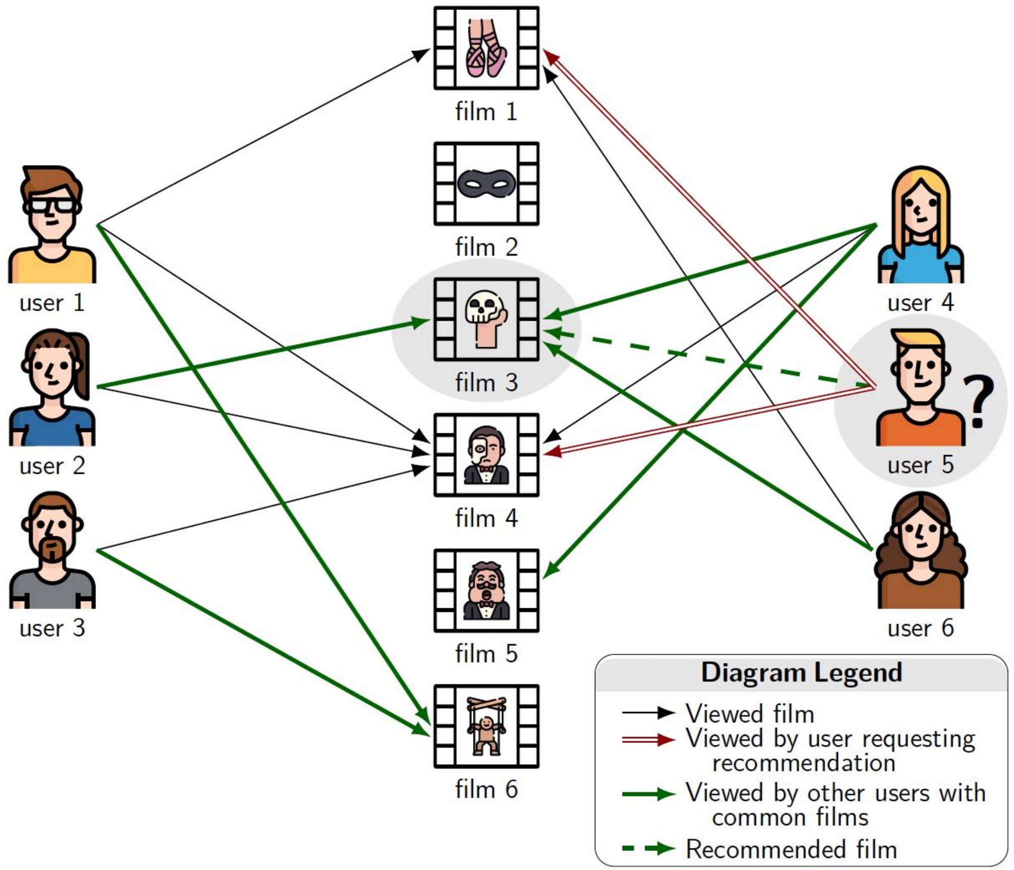


(b)

Collaborative/Social-filtering User-based

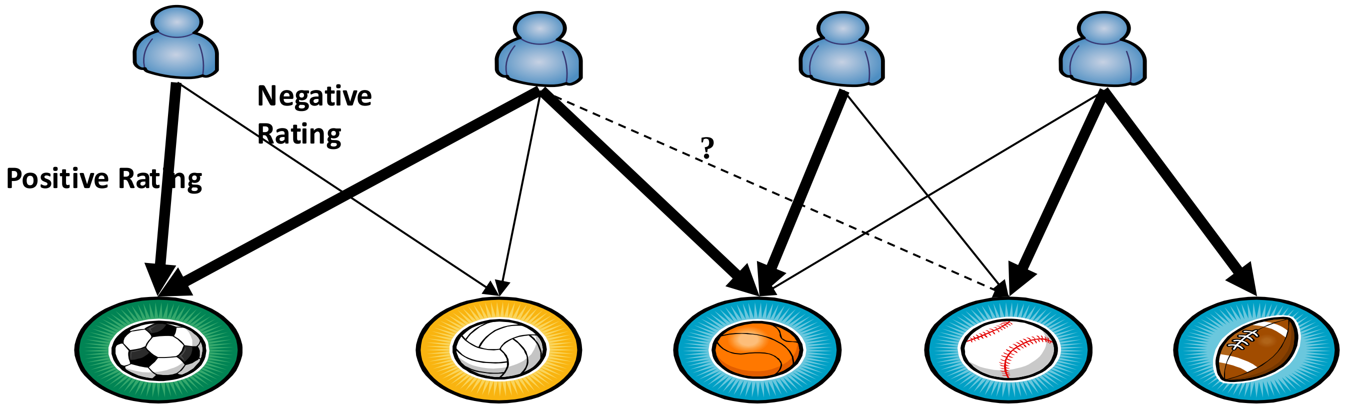


Collaborative/Social-filtering User-based



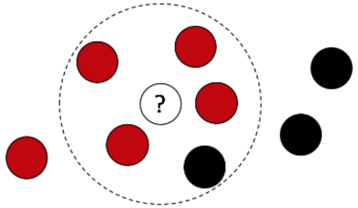
Collaborative/Social-filtering

- Trying to predict the opinion the user will have on the different items and be able to recommend the "best" items to each user based on: the user's previous likings and the opinions of other like minded ("Similar") users

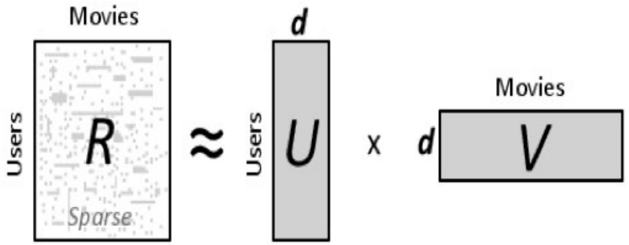


Some Collaborative/Social-filtering Techniques

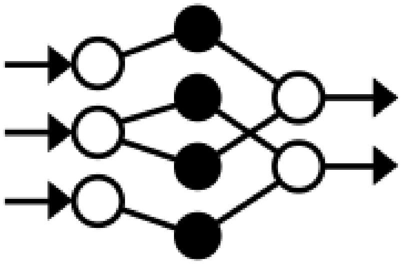
- Nearest Neighbor



- Matrix Factorization



- Deep Learning



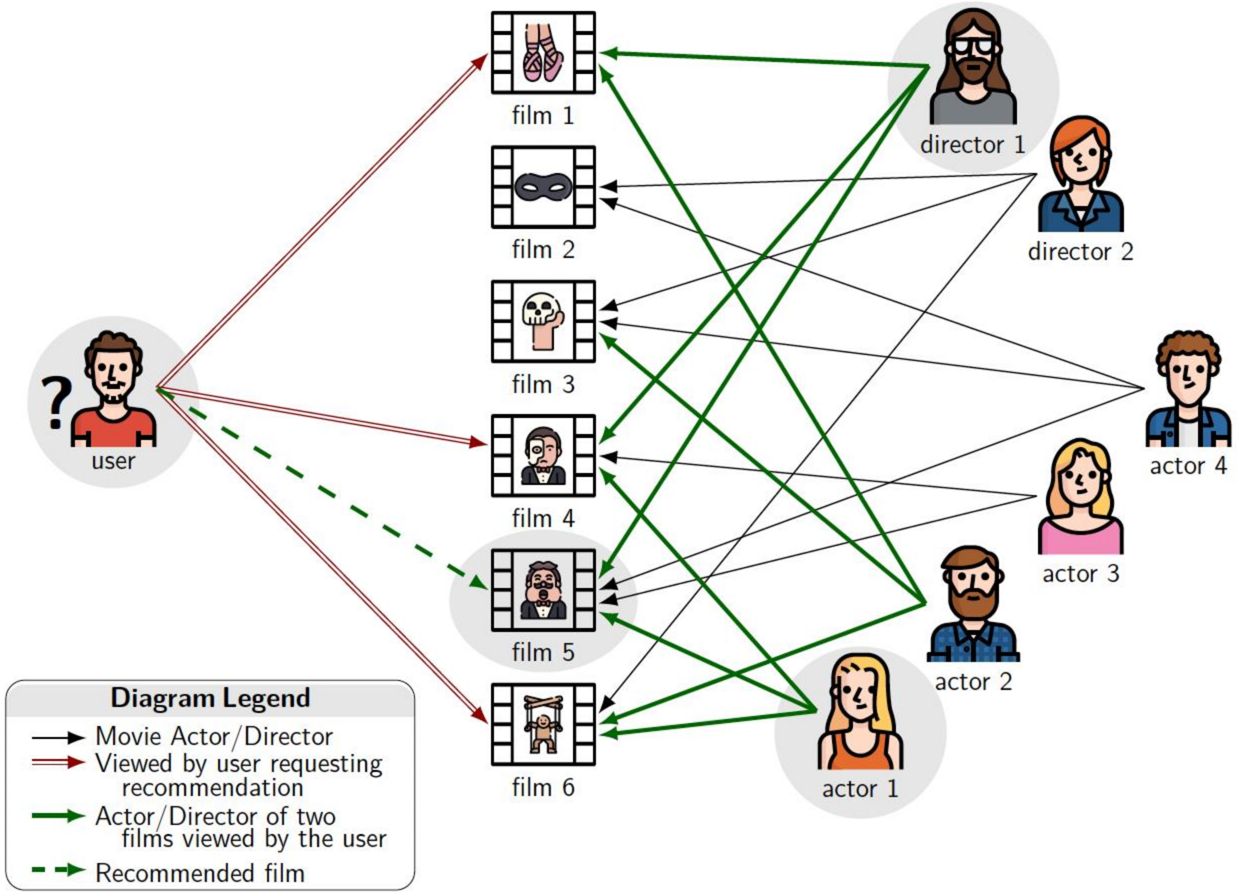
Content-Based Filtering

- The CBF technique tries to capture information from within unstructured or unorganized item's content, such as textual or descriptive attributes
- This technique can generally uses powerful text mining algorithms, from the information retrieval area, to automatically extract content and meta information of the available items
- The system discovers how to recommend items that are identical to those the user liked in the past.
- The similarity of the items is calculated based on the features associated with the items in question. For example, if a user has rated positively one comedy film, then the system will learn to recommend this genre of films

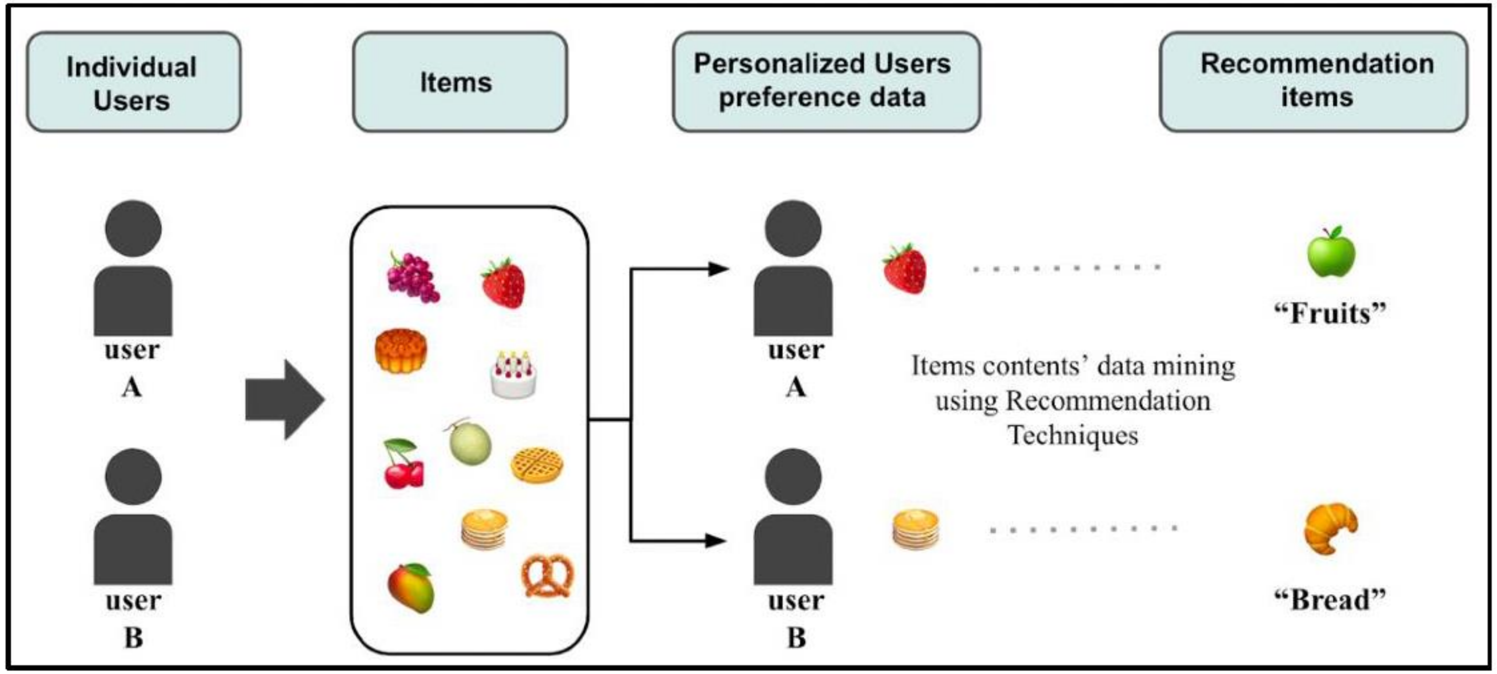
Content-Based Filtering

- The information used to identify the item's similarity can be retrieved in different manners, for example:
  - Extracting relevant keywords from textual descriptions, and comparing them with the user model or other item keywords, using probabilistic estimations
  - Computing full texts into weighted vectors and comparing the similarity of those vectors using bi-dimensional distance mathematical functions

Content-Based Filtering



Content-Based Filtering



Knowledge-Based Filtering

- The Knowledge-Based Filtering (KBF) use is almost required because it implies using any form of domain knowledge in a RS
- In this technique, the focus is placed on the items, their properties and attributes. This kind of information is entirely unavailable in the CBF, which is domain-dependent and represents domain knowledge
- For example, if a film genre can contain only a few pre-defined genres, the actual existence of a genre attribute already distinguishes that item from other domains. Therefore, similarity functions are then performed between items, using those attributes as the basis of comparison, resulting in recommendations of items most similar to items already consumed by the user

Knowledge-Based Filtering

- Apart from similarities between items, items themselves have to be matched, using the user profile defined by the user model, by mixing both kinds of semantics,
- For example: if a system detects that the user has selected several horror movies when trying to recommend new movies for him to watch, it will search for movies of the same genre, possibly with similar titles or actors.
- The main difference between this technique and the CF is that no focus is put on other users whatsoever, but rather on the items themselves

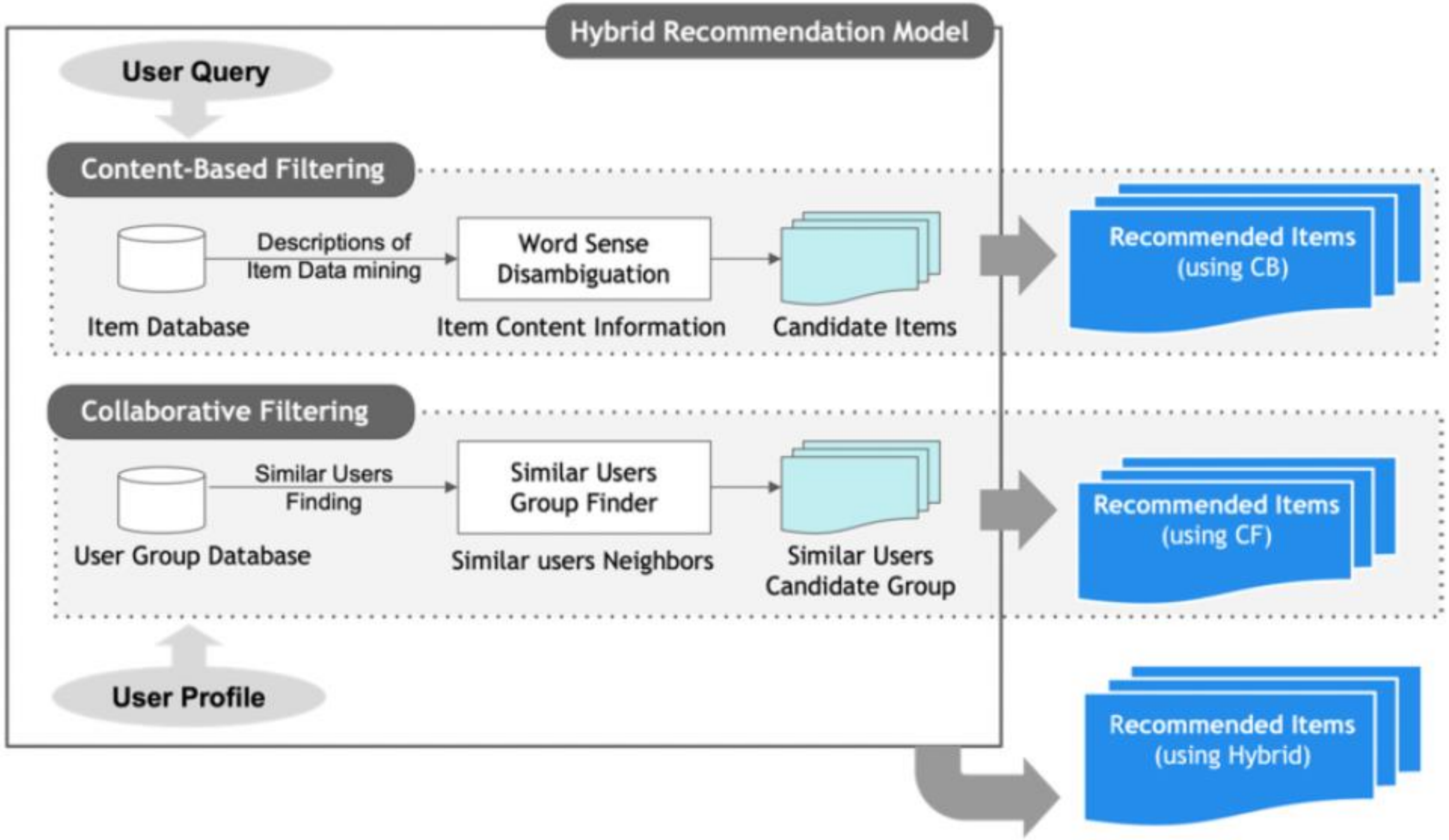
Knowledge-Based Filtering

- Two types of knowledge-based systems are identified:
  - Case-based recommender systems: these systems employ a similarity function to assess the match between user needs (problem descriptions) and recommended solutions. The resulting similarity score directly signifies the utility of the recommendation for the user.
  - Constraint-based recommenders: primarily relying on predefined knowledge bases, these systems incorporate explicit rules on how to correlate customer requirements with item features

Hybrid Filtering

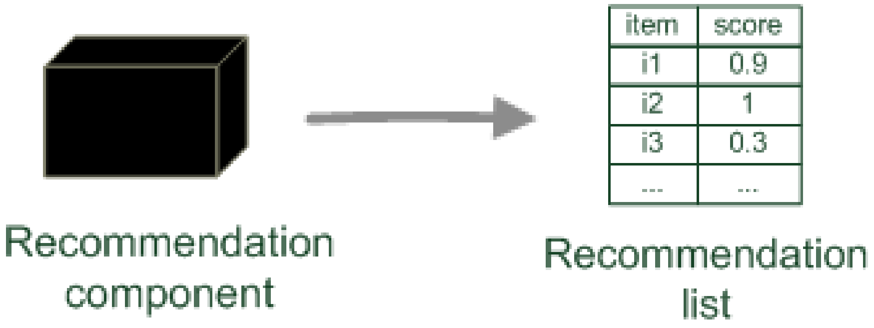
- Authors generally call Hybrid Filtering (HF) to any system whose recommender component is made combining more than one filtering techniques described above or eventually other techniques that result from other perspectives (e.g., stereotypes, demographics, among others)
- For example, a hybrid system may use CBF and simultaneously, searches for domain or semantic keywords amongst those contents, combining their results

Hybrid Filtering



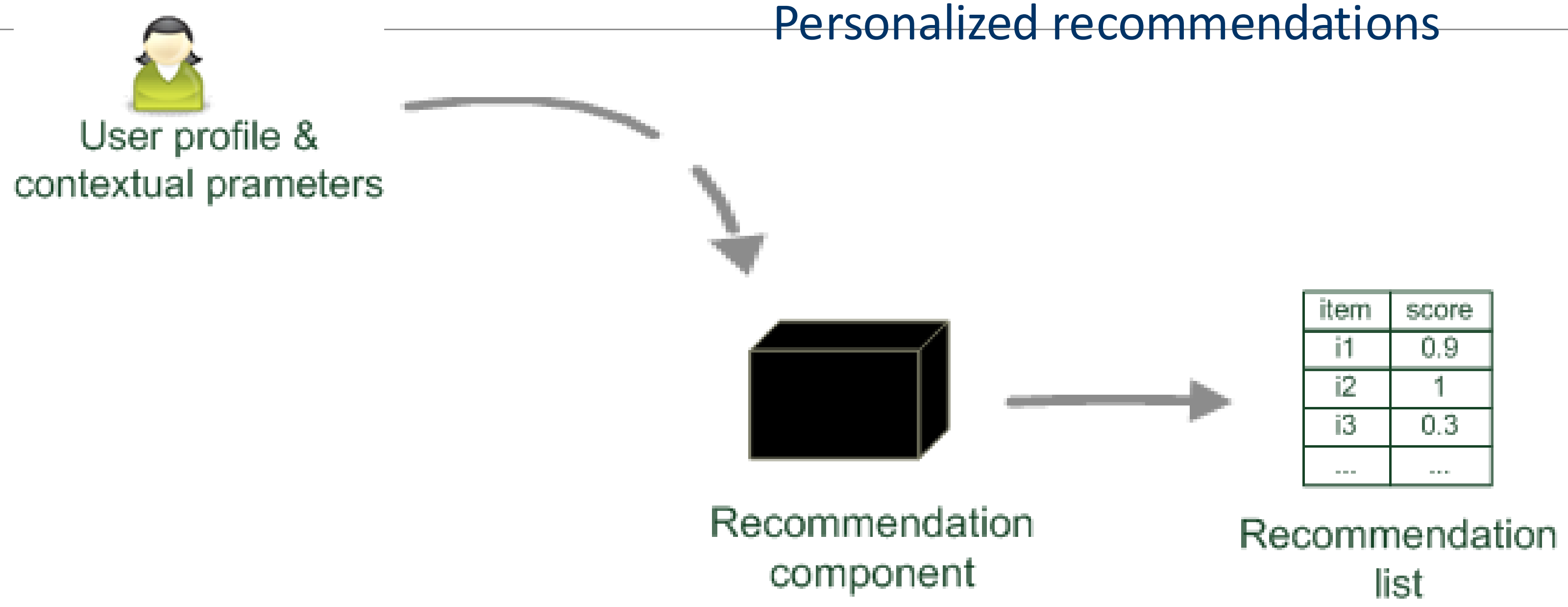
Paradigms of recommender systems

Recommender systems reduce information overload by estimating relevance



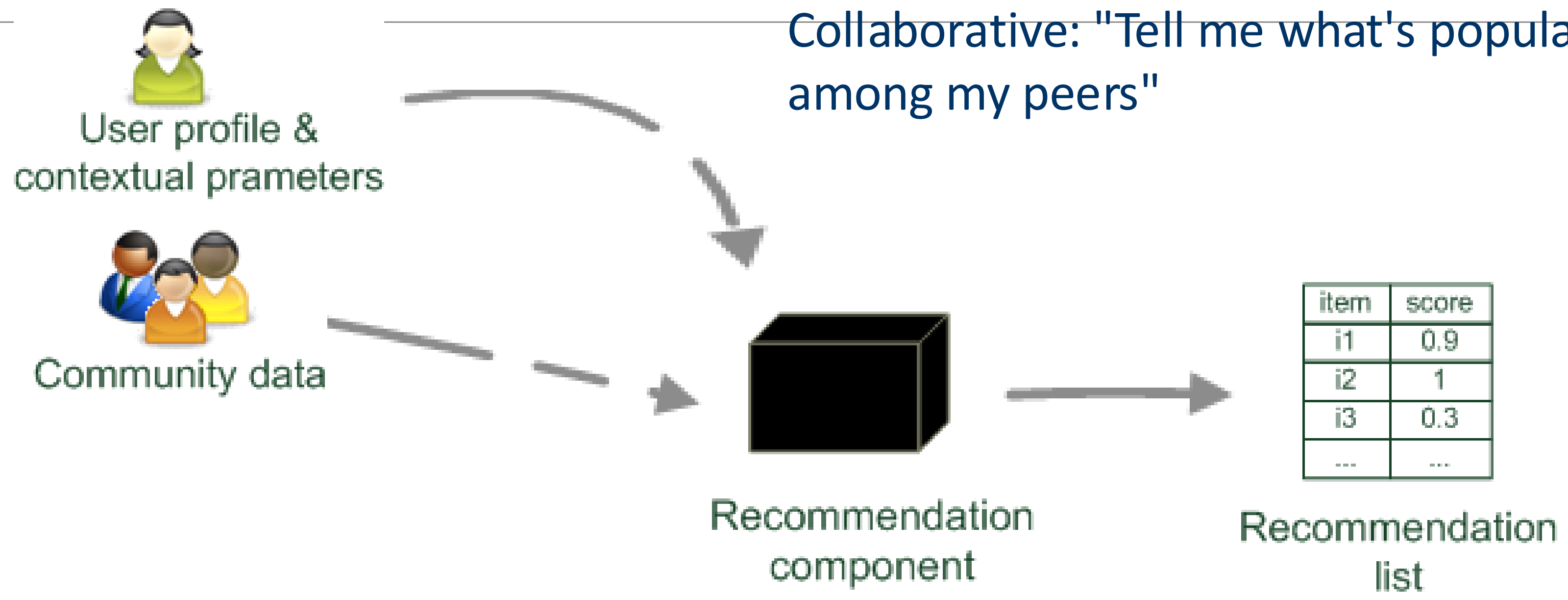
Paradigms of recommender systems

Personalized recommendations



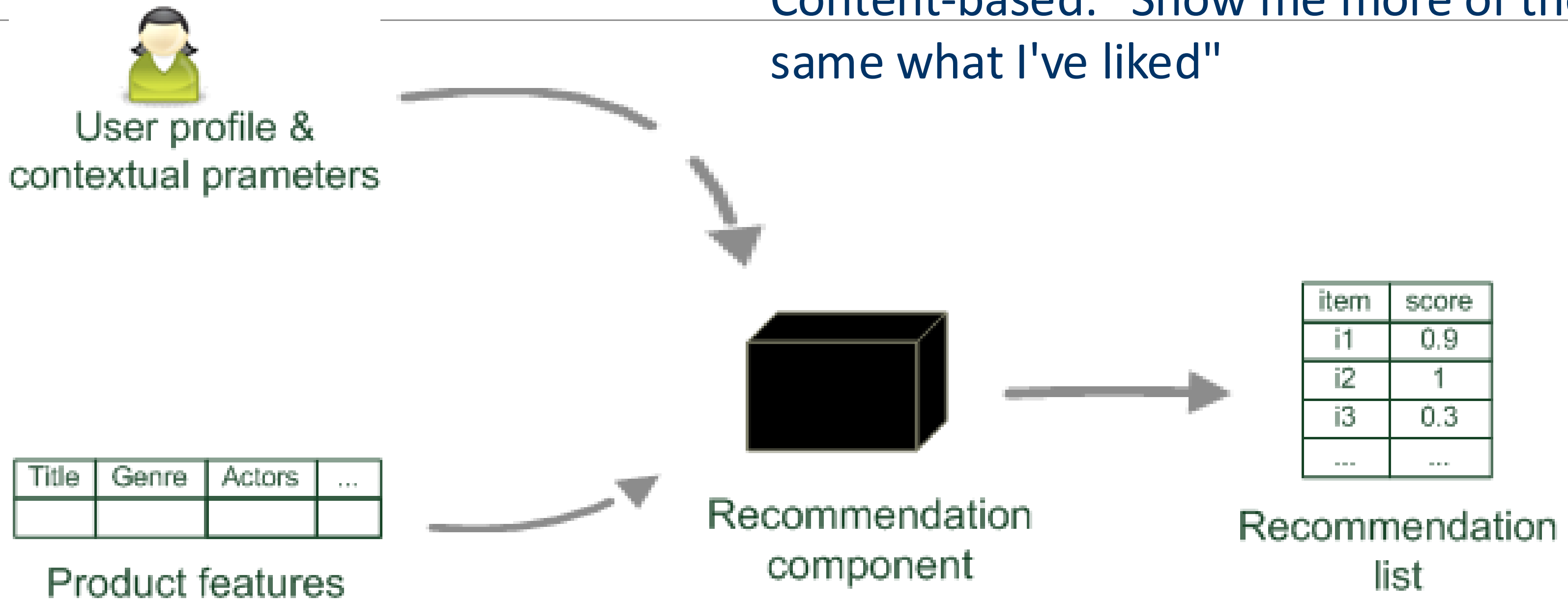
Paradigms of recommender systems

Collaborative: "Tell me what's popular among my peers"



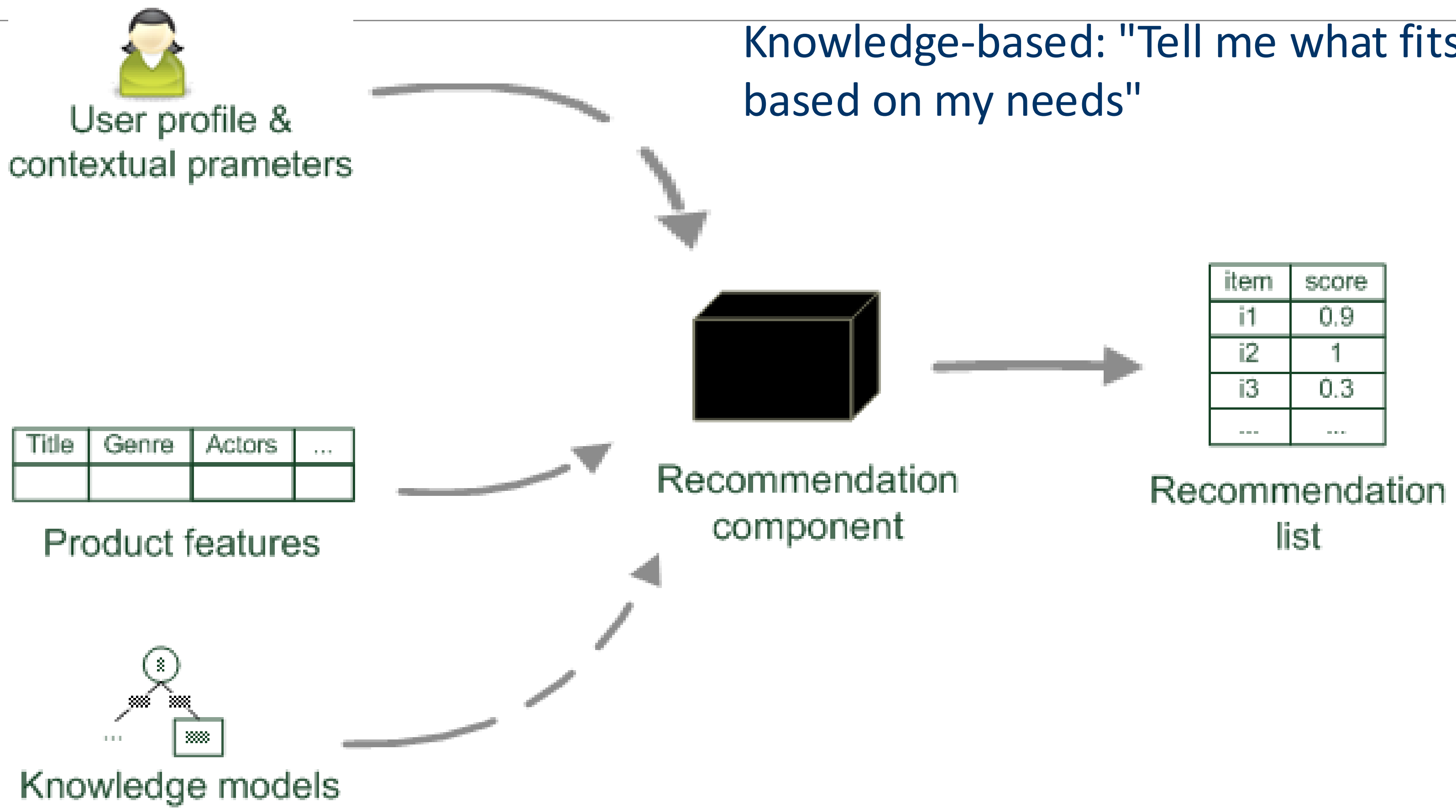
Paradigms of recommender systems

Content-based: "Show me more of the same what I've liked"



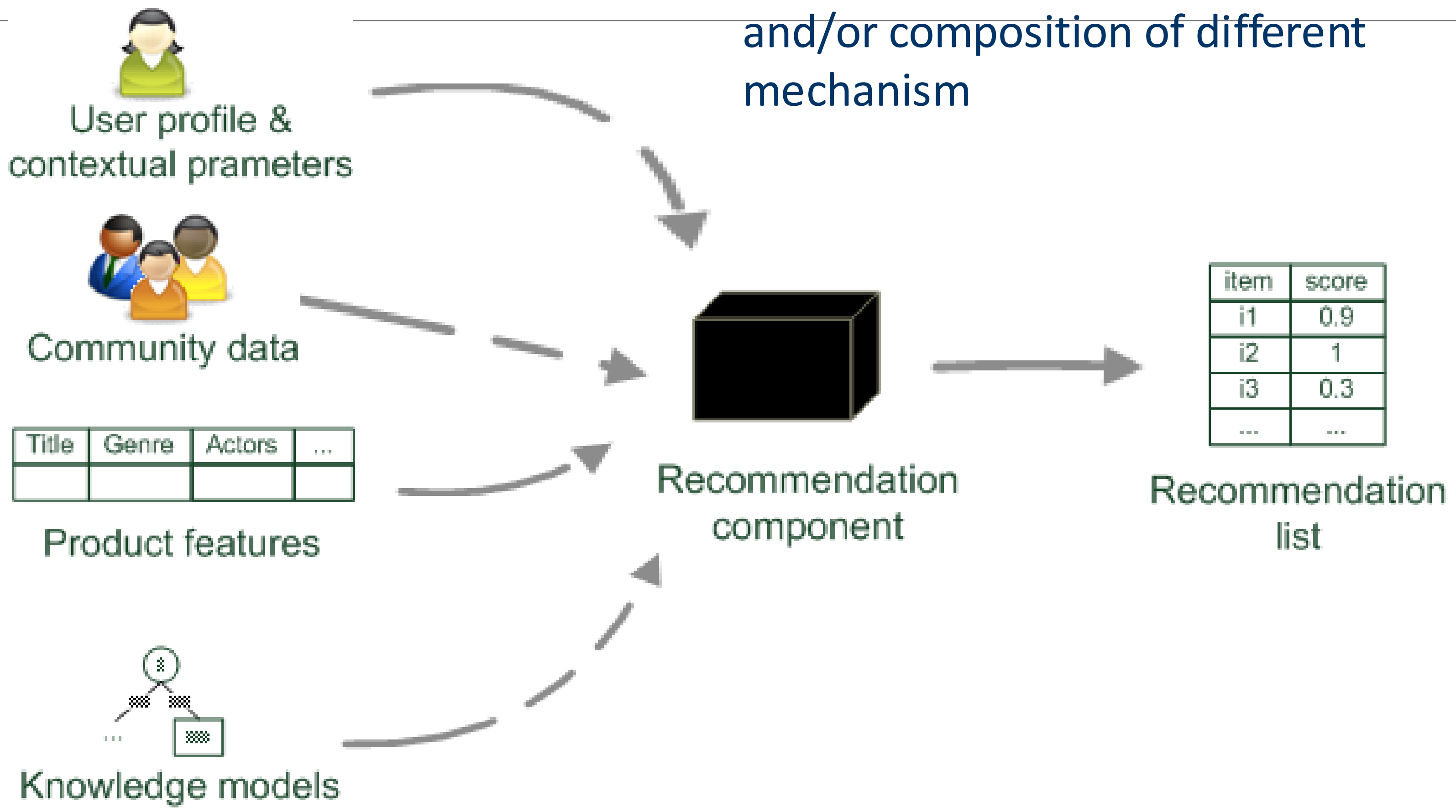
Paradigms of recommender systems

Knowledge-based: "Tell me what fits based on my needs"



Paradigms of recommender systems

Hybrid: combinations of various inputs and/or composition of different mechanism



Recommender Systems Problems and Limitations

- **Black sheep** users who correlate with very few or no other users and are rarely mentioned because the system cannot recommend anything to them.
- The problem of the 'black sheep' in recommendation systems refers to the situation where a user has preferences or behaviors very different from the rest of the community, becoming a kind of 'black sheep' whose preferences are challenging to predict or recommend due to a lack of similarity with other users
- White sheep and Grey-sheep?

Recommender Systems Problems and Limitations

- **Black sheep,** how to solve?

Combining different types of recommendation models Incorporating collaborative and content-based approaches can help balance recommendations, especially for users with distinct preferences

In addition to explicit feedback (ratings), consider including implicit feedback, such as clicks, viewing time, or navigation patterns. This can help better understand user preferences, including those who are 'black sheep

Deep learning, Contextual Information, patterns, Ddynamic Personalization…….

Recommender Systems Problems and Limitations

- **Cold Start** is one of the RS problems, especially in the CF recommenders, because the system does not have enough information, be it ratings, feedback or purchases, to deliver an accurate recommendation
- **Community cold start** can occurs, when a RS is established for the first time, and the user base, ratings and interactions with the systems is still reduced
- **Item cold start occurs**, when a new item is added to the system and does not have enough ratings and has not been bought by any user
- **Dynamic Personalization**, when a new user joins the system and has no purchase history, no defined preferences, and does not correlate with any other users

Recommender Systems Problems and Limitations

- **Cold Start, how to solve?**
- System engage users to provide explicit feedback or preferences during the onboarding process. This information can be used to initialize the recommendation model for new users
- Analyze implicit feedback, such as clicks, views, or dwell time, to infer user preferences. Behavioral data can provide valuable insights even for users with limited interaction history.
- Social Network Information
- Hybrid Models
  - Content-Based Recommendations
  - Knowledge-Based Recommendations

Recommender Systems Problems and Limitations

- **Sparsity**: lack of available data for many items or users in a dataset
- The matrix of interactions between users and items is often very sparse, with **most entries missing**
- Example: Consider a matrix where rows represent users, columns represent items, and cells contain interactions (such as ratings or views). If most users interacted with only a few items and most items were rated by only a few users, the matrix will be sparse

Recommender Systems Problems and Limitations

- **Sparsity**, two adverse effects:
- There are not enough common items between the recommended user and their neighbours, rendering it challenging to establish the similarity between the users accurately
- Recommending items to the user is challenging because their neighbours, most likely, had not rated positively the same items

Recommender Systems Problems and Limitations

- **Sparsity**, how to solve?
- Sparsity is often associated with the cold start problem, where it is difficult to make accurate recommendations for new items or users with little or no previous interaction
- **Same approach to cold start problem**
- Also, Fill missing values in the interaction matrix using imputation methods. Common imputation approaches include mean or Median imputation, k-nearest neighbors imputation, or advanced machine learning methods
- Matrix Factorization Techniques: like Singular Value Decomposition (SVD) or Alternating Least Squares (ALS) can be used to factorize the user-item interaction matrix into latent factors. This helps in capturing underlying patterns and reducing the impact of sparsity.

Recommender Systems Problems and Limitations

- **Over-specialization** occurs when the recommendation model becomes excessively customized to individual user preferences, potentially resulting in a limited variety of recommendations
- Means the **set of recommended items are similar**, and the **items are closely related to the items the user already purchased**
- Leading to the **RS tendency to suggest similar items**
- Ideally, a different range of options should be given to the user, and not a similar set of items

Recommender Systems Problems and Limitations

- **Over-specialization**, how to solve?
- Hybrid Models
- Incorporate diverse data types, including contextual information, demographic data, and implicit feedback, to provide a more holistic view of user preferences
- Actively introduce users to items outside their typical preferences
- Regularly evaluate the model's performance, integrate user feedback, and adjust its parameters to avoid over-specialization

Recommender Systems Problems and Limitations

- **Scalability** can occurs in environments with millions of users and products where RSs make recommendations. A real-world dataset growing extensively, demands high computational time for making recommendations, which grows proportionally to the number of users and products in the system
- Large traditional CF systems will suffer scalability problems severely with computational needs exceeding practical or acceptable levels
- **Model-based techniques** do not suffer from scalability because the time spent building the model, has no effects in the user response time

Other Problems

- Inconclusive user feedback forms
- Finding users to take the feedback surveys
- Weak Algorithms
- Poor results
- Poor or lack of Data
- Privacy and ethics

Next-generation RS?

- Multicriteria recommender systems: Exploiting multicriteria ratings containing contextual information as an additional source of knowledge for improving the accuracy
- Context awareness: Taking time aspects, geographical location and additional context aspects of the user into account. Emotional context ("I fell in love with a boy. I want to watch a romantic movie.")
- Group recommendations: Recommendations for a couple or for friends. Accompanying persons?
- NLP? LLM?

Next-generation recommenders might someday be able to simulate the behavior of an experienced salesperson instead of only filtering and ranking items from a given catalog.

References

Serão colocadas depois da entrega do trabalho