Artificial Intelligence (CS303)

Lecture 8: Recommender System

Hints for this lecture

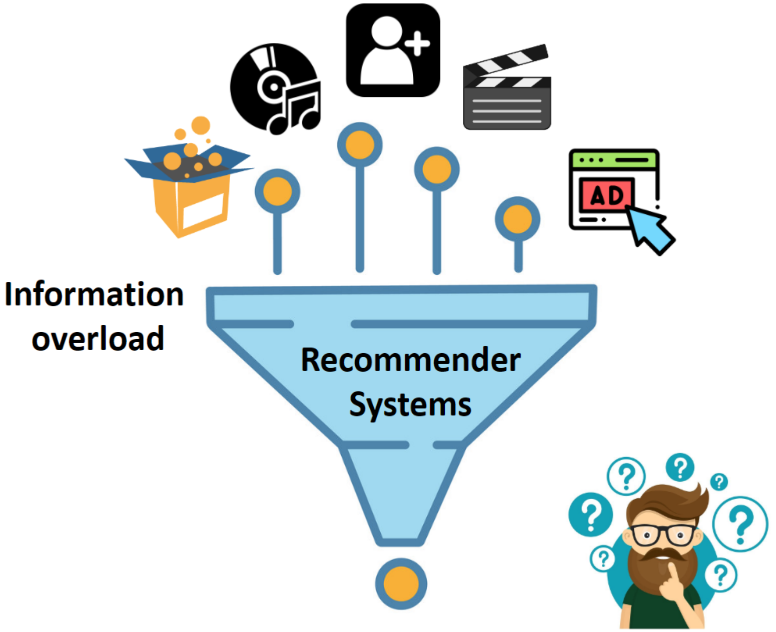
- **Recommender systems infer the preference of users and recommend relevant information to users, and are usually built with machine learning techniques.**

Outline of this lecture

- **Overview of recommender system (RS)**
- **How does RS do recommendation?**
- **How to build a RS?**

Why we need a Recommender System?

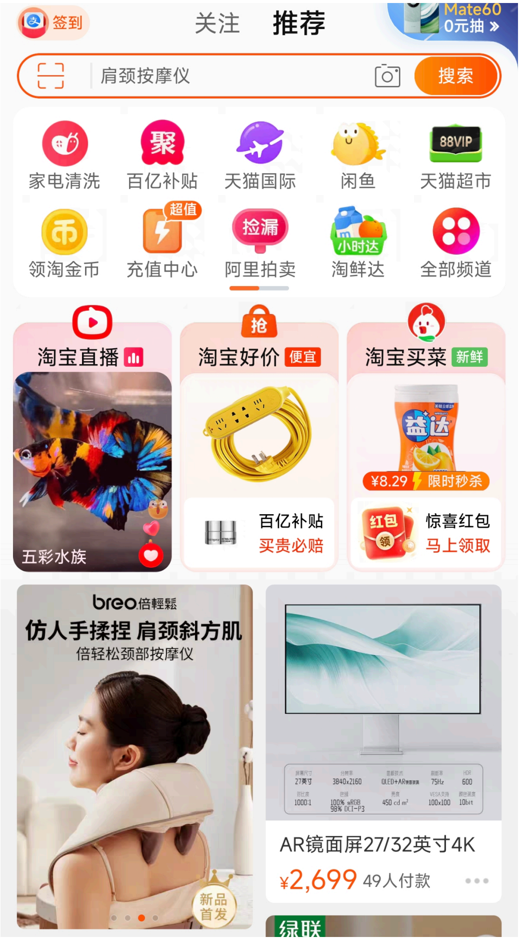
Age of Information Explosion

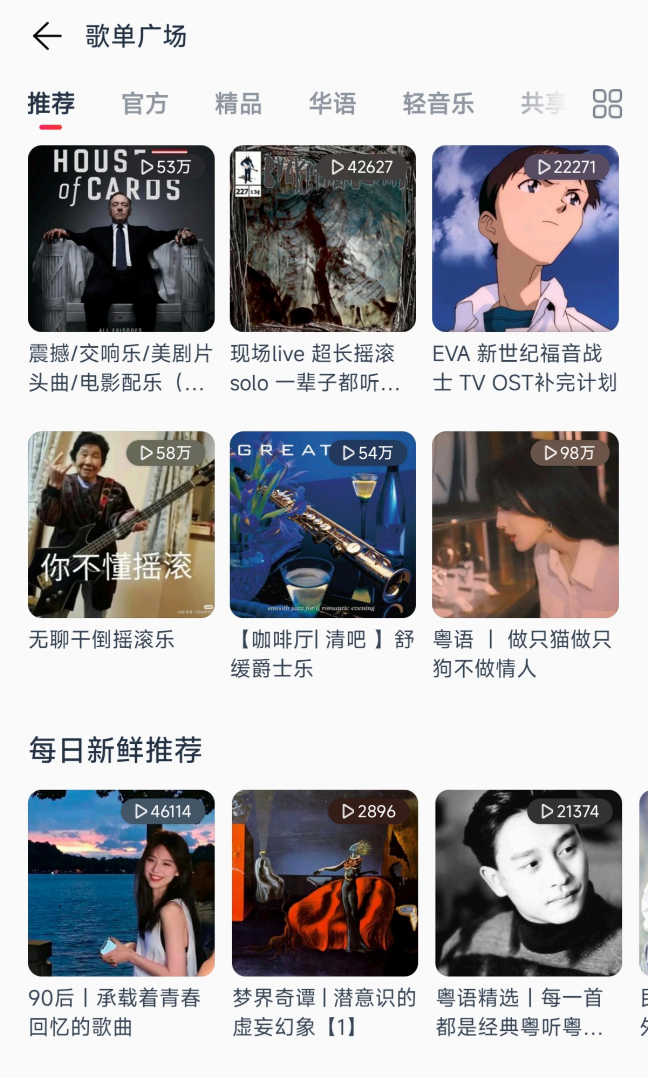


Recommend item X to user

Items can be: Products, News, Movies, Videos, Friends, etc.

Services that deploy recommender system



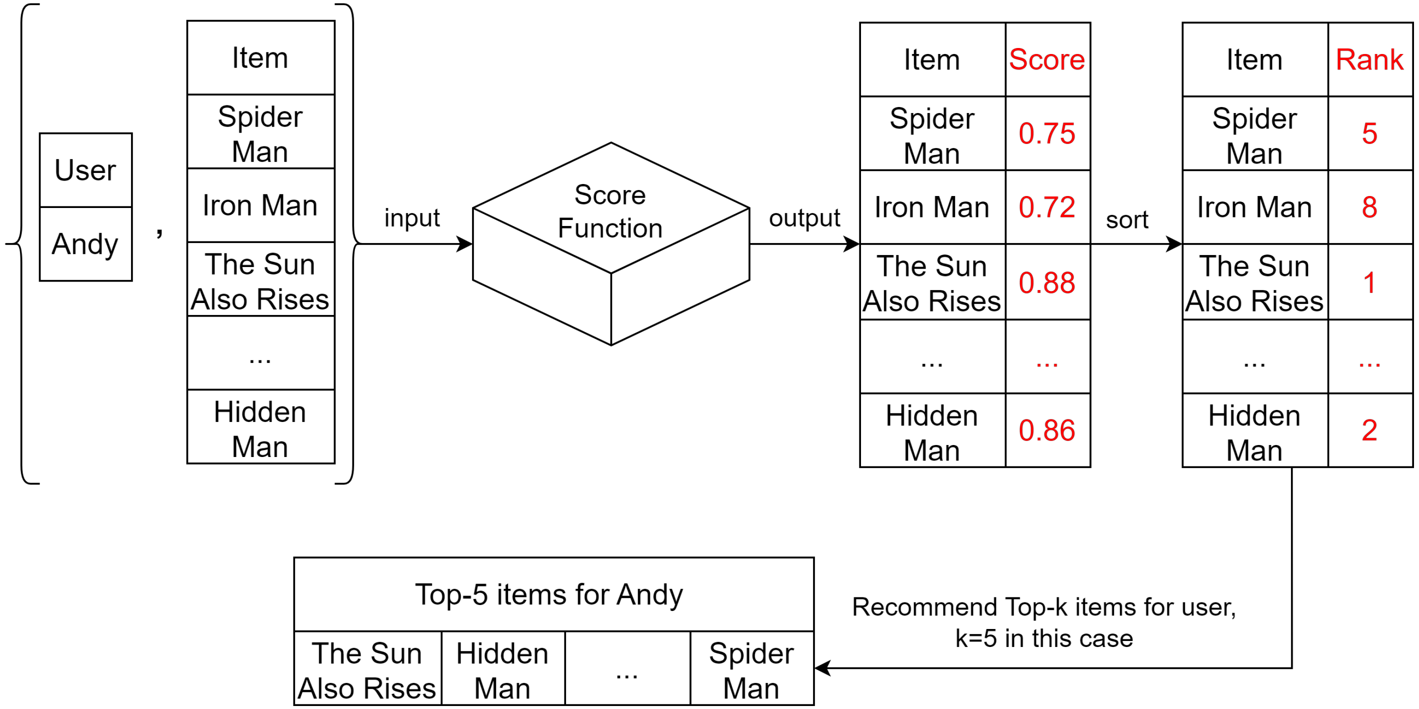


What is a Recommender System?

- Recommender System recommend new items to its user.
- Based on what can we recommend an item (to a user)?
  - The items that the user has been interacted.
  - The users who have been interacted with similar items.
- The recommendation is personalized.
- The key of Recommender System is a score function.
  - Input: a user and an item.
  - Return value: a score, indicating how likely the user would be interested in the item.

The key of RS is a score function

- Suppose there is a function to score the combination of user and item.
  - RS basically estimate the probability of interaction between a user and an item,
  - The score function is essentially a **model trained with data**.



Recommendation based on score functions

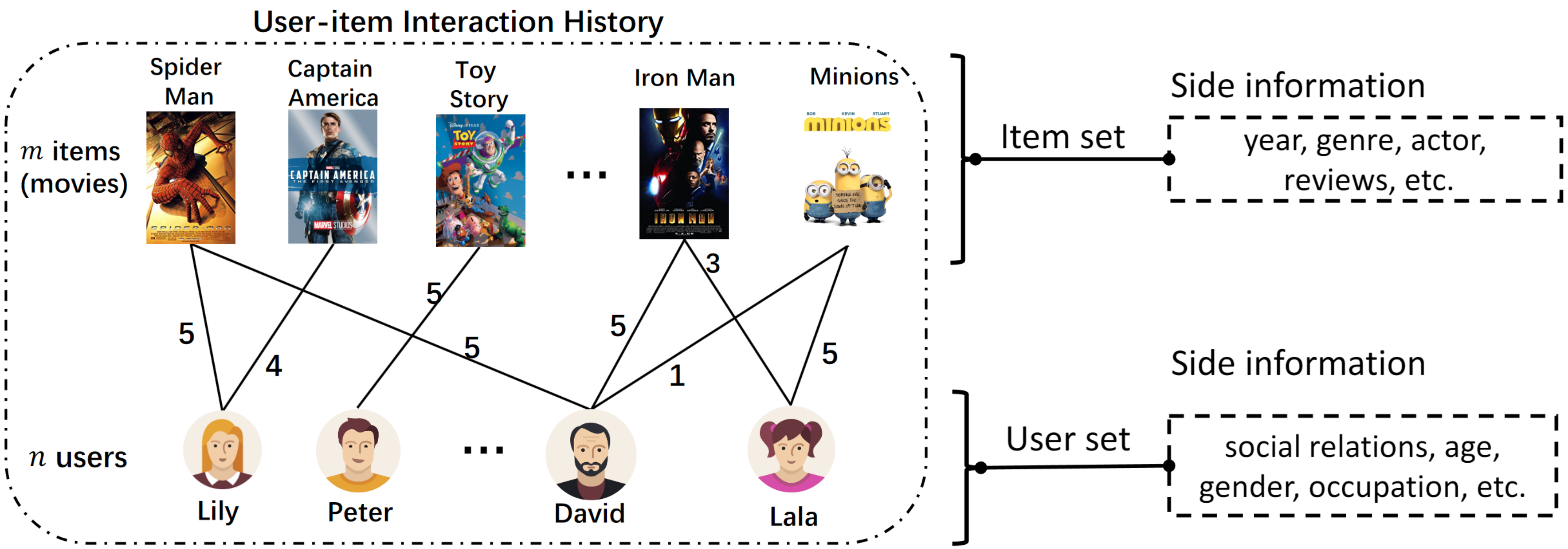
- **In practice, the score function could be very complicated since**
  - The RS needs to be efficient (make recommendations in seconds)
  - In many applications, we may have millions of users and items
  - There is always a trade-off between efficiency and accuracy

How to build a RS?

(How to get a suitable score function)

Problem Formulation

- **Input:** Historical user-item interaction records or additional side information (e.g. user's social relations, item's knowledge, etc.)
- **Output:** The score function

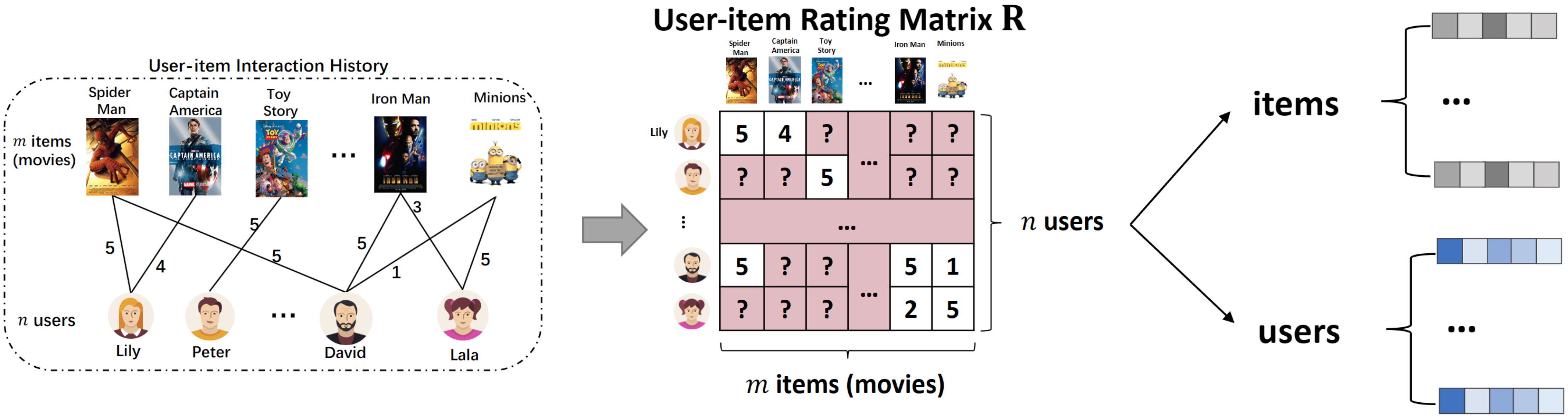


Typical methods

- **Content-based method**
  - The very basic idea: build a regression/classification model for each user
    - Focusing on the side information of the items (i.e., attributes, features of items)
    - Suggesting items by comparing their features to a user's past behaviors
- **Collaborative Filtering method:**
  - Predicting user preferences based on the behaviors of other users.
  - Based on the historical user-item interaction data
- **Hybrid method:** Combination of CF-based and Content-based method

CF-based method

- **Attributes/features of users and items are not available,**
  - How to build the regression/classification model (as the score function)?
  - Learning representation of users and items



Task: predicting missing movie ratings in Netflix.

Represented by correlation

- Represent the user/item by its correlation with the other users/items.
  - Users with similar historical interactions are likely to have the same preferences.
  - Items that are interacted by similar users are likely to have hidden commonalities.

- A user/item is represented by a vector that consists of all the correlation between itself and all the users/items.

Represented by correlation

- How to define the correlation between 2 users or items?
  - Pearson Correlation Coefficient：a normalized measurement of the covariance

$$c_{u_1 u_2} = \frac{\sum_{i \in M} \left(r_{u_1,i} - \bar{r}_{u_1}\right)\left(r_{u_2,i} - \bar{r}_{u_2}\right)}{\sqrt{\sum_{i \in M} \left(r_{u_1,i} - \bar{r}_{u_1}\right)^2} \sqrt{\sum_{i \in M} \left(r_{u_2,i} - \bar{r}_{u_2}\right)^2}}$$

- An example of user's Pearson Correlation Coefficient
- M: The item set
- $r_{u,i}$: Interaction record between user u and item i
- $\bar{r}_u$: Mean value of all the interaction records of user u

Represented by correlation

- A user/item is represented by a vector that consists of all the correlation between itself and all the users/items
- Interaction probability of user u and item i can be calculated by a function $f(u,i)$, some simple example of f:
  - $f(u,i) = \sum_{u' \in U} c_{u,u'} r_{u',i}$, U is user set, $c_{u,u'}$ is the correlation coefficient between u and u', $r_{u',i}$ is the interaction record of u' and i.
  - $f(u,i) = \sum_{i' \in M} c_{i,i'} r_{u,i'}$, M is item set, $c_{i,i'}$ is the correlation coefficient between i and i', $r_{u,i'}$ is the interaction record of u and i'.

Represented by correlation

- Represent the user/item by its correlation with the other users/items.
- **Advantage**: high interpretability
  - It is easy to explain why the system recommend the item to the user.
- **Disadvantage**: low scalability
  - What if there are millions of users and millions of items?
  - High-dimensional, sparse feature representation

Represent by matrix factorization

- A matrix $R \in \mathbb{R}^{n\times m}$, it is approximate to the product of two matrix:

$$R \approx PQ^{\mathrm{T}}, P \in \mathbb{R}^{n\times d}, Q \in \mathbb{R}^{m\times d}$$

- For a user-item interaction matrix $R \in \mathbb{R}^{n\times m}$, n is the number of users, m is the number of items.
- Representing the user and item as a d-dimension vector
- Matrix P, Q consist of the representation vectors of all the users and items.
- The low-dimension vector representation is also called as **embedding vector**.

Represent by matrix factorization

- How to find a good representation of user and item?

$$\min_{P,Q} \sum_{r_{u,i} \in R'} \left\| r_{u,i} - P_u Q_i^{\mathrm{T}} \right\|$$

  - We can't know all the elements in the R, $R^{'}$ is the set of the known elements in the R.
  - r_{ui} is an interaction record of user u and item i.
  - P_u is user u's embedding vector, and Q_i is item i's embedding vector.
- Interaction probability of user u and item i is $r_{u,i}^{'} = P_u Q_i^{\mathrm{T}}$

Represent by matrix factorization

- If we define an interaction probability function $f(P_u, Q_i^{\mathrm{T}})$, P_u is user u's embedding vector, and Q_i is item i's embedding vector.
  - $r_{u,i}^{'} = f(P_u, Q_i^{\mathrm{T}}) = P_u Q_i^{\mathrm{T}}$ is a simple example for this function.
- We can replace the matrix multiplication with a complex model M, such as MLP
  - The objective function will be: $\min_{P,Q,M} \sum_{r_{u,i} \in R'} \left\| r_{u,i} - f(P_u, Q_i^{\mathrm{T}}, M) \right\|$
  - The model with higher complexity **may** have better prediction performance in **big data scenario**.

To be continued