Reccomandation system

The idea is to match consumers with the most appropriate products/services is key to enhancing user satisfaction and loyalty.

R.S. formalism

- Set of users $U = \{u_1 \dots u_m\}$
- Set of items $I = \{i_1 \dots i_n\}$
- Utility function (user-item matrix) $r : U \times I \to R$
- $R \subseteq \mathbb{R}$, it defines the set of **ratings**, $R$is made by either discrete ratings (i.e. from 1 to 5 stars) or continuous values.

The utility functions

You create a matrix made by **m rows (number of users)** and **n cols (number of items)** , each matrix's entries rappresent the rating give to a specific items by a specific user. Howewer this matrix tends to be very sparse, there are lot of missing data, not every users rates every items.

The 3 Key problems for a r.s.

- **1. Data collection**, how to populate the matrix
- **2. Rating prediction**, fill the gap w.r.t the missing rating
- **3. R.S. eval**, measure the perfomance of recommender methods

1. Data collection

Two ways to do it:

- **Explicit**, ask people to rate items;
- **Implicit**, find out using user's behaviour i.e. the user queries and the click etc..., but the user doesn't explicit give a value to the item

2. Rating prediction

Two main issues:

- The matrix is very spare;
- Cold start, the new users cannot have an history, and the system cannot have any specific

information about him to give a recomandation.

3. R.S. eval

- RMSE
- Precision, Recall at K
- Personalizzation
- Serendipity

Recommendation strategies

1. Content-based filtering
2. Collaborative filtering
3. Hybrid filtering

1. Content based filtering

Reccomend items to user u similiar to previous items rated highly by u.
Given the **item profiles**, the description of am item, we want to be able to create a **user profiles**, it says what the users likes.
After do that we want combine **user profiles** with the item trough all catalog.

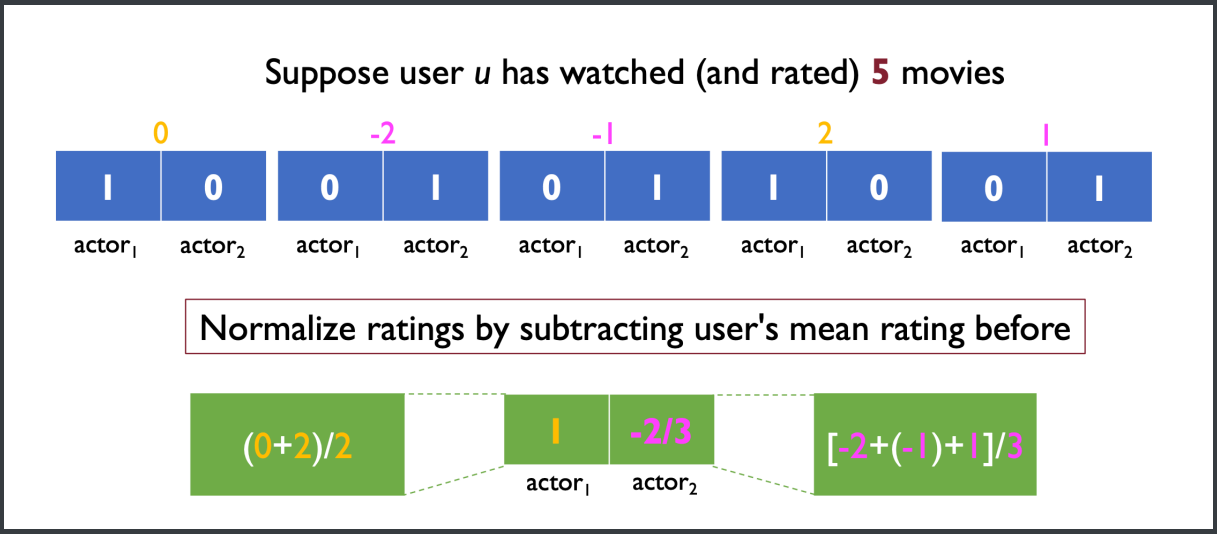
- **item profiles**, for each item i create a profile, i.e. a set of featues, that's better rappresent the specific item, this is a f.e. taks.

> Let's take an item as article, the possible profile could be an array of word for each article, and this array contains the a score for every word in the arrays.

User profiles strategies

For each user create a user profile.

- **the simplest solution** is to take the average through all items rated by the user, this approach is not fair, all items are treated equally



The slides above shows a list of item profiles, which are rappresentad using features actor1, actor2, up to every cell there's the scaled rating for that moovie.

> **The user profile is the green rectangle, which is the mean of the rating across the 5 items profile previously chosen.**

Building predictions, content-based filtering

- Given user profile **u** we can estimate the missing entries of the utility matrix for u;
- For each item **unrated** by **u**, compute the cosine sim (or another) between u and the corresponding item profile vectors;
- Finally, we pick the top-k items with **the highest similarity score**, and we reccomend those to u.

$$A^0 = \emptyset$$
$$A^1 = argmax_i\{sim(u,i) : i \in I - I_u - A^0\}$$
$$A^2 = argmax_i\{sim(u,i) : i \in I - I_u - A^0 - A^1\}$$
$$R_{u,k} = \bigcup_{j=1}^{k} A^j$$

PROs

- No need for data on other users: only the user and the items rated by her/him are needed;
- no **item cold start** problem, when the item is new or unpopular, it can still be reccomended to users with highest profile similarity;
- you have to perfom a sort of feature engineerings strategies, and it's very human understable.

CONs

- Find the best featues which express well the items;
- unable to find the quality judgments of other users;
- **user cold start**, for new users, if the user hasn't rated any items, then there's no user profile;
- overspecializzation, never reccomends items outside user's prfile;

2. Collaborative filtering

Recommend items to user *u* based on preferences of other users similar to *u*

> Here we aren't using any sort of f.e. techniques, we don't need any effort in order to extract information

Three main approaches

- 2.1. Neighbourhood-based

2.1. Neighbourhood-based

Compute the relationship between useres or items, the first one evaluates a user's preference for an item based on ratings of "neighboring" users for that item; whereas item-based evaluate preference on rating of "neighboring" items by the same user

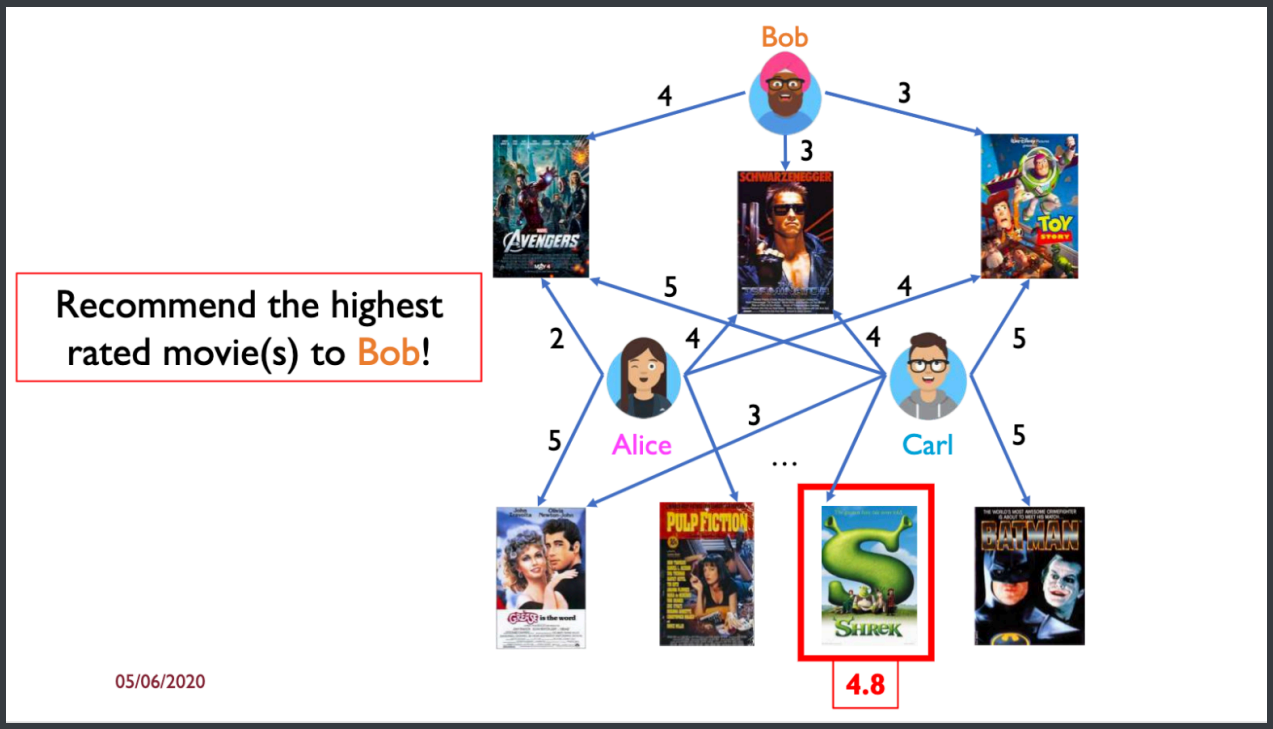
2.1.1. User based Neigh.

Given a user *u* and an item *i* not rated by *u*,we want estimate r(u,i), takes the set $\{u' : u' \neq u\}$, who have already rated the item *i*, extract a subest of k neighbours of *u*

> K-neigh of *u* is found on the basis of the similarity between user ratings without the need of explicit user profiles

In theory, rating prediction *r(u,i)* could be defined on any *i* not rated by *u*, in pratice we are interessed only in estimating *r(u,i)* for those *i* which have been rated by the u's k-neigh.

> Hint: if a user *v* is not in the k-n of *u* then very likely *u* will not be interested in any item that only *v* has rated.



Bob has Alice and Carl as your neigh., according to the mean compute through all moovies watched by Alice and Carl **user-based neigh.** reccommend Sherek which is the highest.