

Recommendation Engine

The Back bone of recommendation engine is history of user data collected and its correlation with other users. Higher the correlation higher is the similarity between the user, thus we can suggest the users based of their selections. These factors are saved in a database in the form of vector. Some methods of finding the similarity between tow vectors are as follows

Cosine similarity

All the information related to movies is stored in another vector called the *item vector*. Item vector contains the details of each movie, like genre, cast, director, etc.

The content-based filtering algorithm finds the cosine of the angle between the profile vector and item vector, i.e. **cosine similarity**. Suppose A is the profile vector and B is the item vector, then the similarity between them can be calculated as:

$$\text{Similarity}(A,B)= \cos(\text{teta})= \frac{A.B}{|A||B|}$$

Based on the cosine value, which ranges between -1 to 1, the movies are arranged in descending order and one of the two below approaches is used for recommendations:

- **Top-n approach:** where the top n movies are recommended (Here n can be decided by the business)
- **Rating scale approach:** Where a threshold is set and all the movies above that threshold are recommended

Euclidean distance:

Similar items will lie in close proximity to each other if plotted in n-dimensional space. So, we can calculate the distance between items and based on that distance, recommend items to the user.

Pearson's Correlation:

It tells us how much two items are correlated. Higher the correlation, more will be the similarity. A major drawback of this algorithm is that it is limited to recommending items that are of the same type. It will never recommend products which the user has not bought or liked in the past. So if a user has watched or liked only action movies in the past, the system will recommend only action movies.

Selecting users instead of checking correlation between every user will save time.

- Select a threshold similarity and choose all the users above that value
- Randomly select the users
- Arrange the neighbours in descending order of their similarity value and choose top-N users
- Use clustering for choosing neighbours.

Different methods of recommendations are as follows

User User collaborative filtering:

This method used average user ratings for different movies and compares with a user with the same rating for similar movies. When the correlation between the user for rating different movies is high then the movies watched by the highly correlated user and suggestion to this user. The movies can be recommended vise versa.

In this method the average user rating is considered:

User/ Movies	Movie 1	Movie 2	Movie 3	Mean user rating
User 1	3	2	4	3
User 2	1	4	1	3
User 3	--	2	4	2
User 4	1	5	--	

Form the above table base on the user behaviour on the movie ratings. Movies are suggested to the other user.

User 3: we suggest movie 1

User 4: its better not to suggest movie 3

This Algorithm is time consuming. As it tends to get suggestion from all possible user. Sometimes, there is no point in spending time check similar movies for suggestion with user that doesn't add up any important value. So, in order to filter out the irrelevant user comparison there are different methods like

Randomly selecting users,

Arrange user in the descending order if the correlation value and select the to n no. of user.

Using cluster analysis and selecting k user with common characteristics.

Item-Item collaboration Filtering:

We can take a simple example for the same approach:

Customer A	Customer B	Customer C	Customer D
Bread	Milk	Bread	Bread
Jam	Eggs	Cheese	Butter
Milk	Bread	coffee	?
Butter	coffee	?	?

With Item to Item collaboration we can recommend

Customer C: Eggs and milk

Customer D: Milk, Jam and Eggs

The Item to Item Collaborative Filtering is less tedious. As it is check for the most paired item for one particular item and recommends according. Whereas based on User to User rating will be more time consuming as we might have to check for correlation between all the user.

The whole process of recommendation is completely dependent on the data collected from the user behaviour his ratings, history of his clicks, navigation etc. what how would be predict and suggest a new user.

The Best way to handle new users are to suggest them the most popular items or movies, along with items and movies that might fetch huge business returns for the organisation.

Over all I feel item-item base collaboration is the most effective method of recommendation.