Recommender Systems

A recommender system is a tool that is deployed in the environment where items (products, movies, events, articles) are to be recommended to users (customers, visitors, app users, readers) or the opposite.

A recommendation engine filters the data of user and items using different algorithms and recommends the most relevant items to users or the opposite.

Recommender systems function with two kinds of information:

Characteristic information. This is information about items (keywords, categories, etc.) and users (preferences, profiles, etc.).

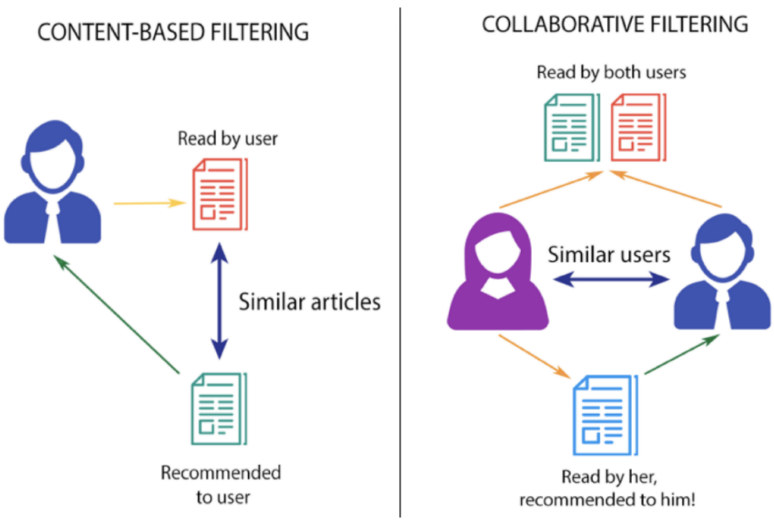
User-item interactions. This is information about interaction of users and items (ratings, number of purchases, likes)

Based on this, we can distinguish between three algorithms used in recommender systems:

Content-based systems, which use characteristic information.

Collaborative filtering systems, which are based on user-item interactions.

Hybrid systems, which combine both types of information with the aim of avoiding problems that are generated when working with just one kind.



Content-based recommender systems

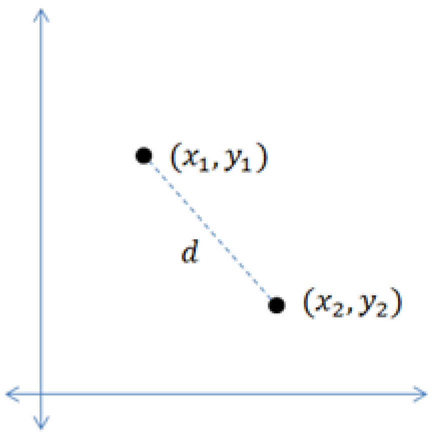
Content-Based Recommendations systems are the systems that look for similarity before recommendations. We all have seen whenever we are looking for a movie or web series on Netflix, we get the same genre movie recommended by Netflix. But how does this work? How does Netflix compute what I like? This is all done through content-based systems. The similarity of different movies is computed to the one you are currently watching and all the similar movies are recommended to us. In the case of e-commerce website similarity in terms of products is calculated. Considering I am looking for a MacBook then the website will look for all similar products that are similar to MacBook and straight away will recommend us.

How to measure the similarity?

The similarity is the main key fundamental in the case of content-based recommendation systems. A most similar thing to what we are currently watching gets recommended to us. There are different techniques or similarity measures that are used to compute the similarity.

- Euclidean Distance

Suppose (x1, y1) and (x2,y2) are two datapoints.



$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

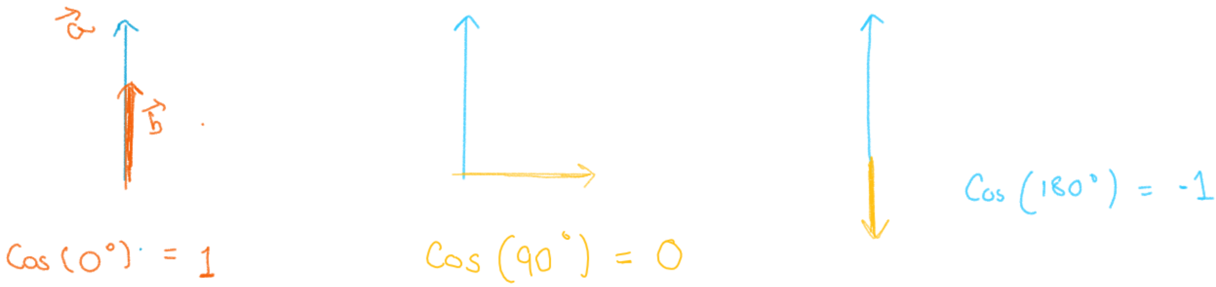
- Cosine similarity

It defines the linear relationship b/w two vectors

Suppose A and B are 2 movie vectors, then the similarity between them can be calculated as:

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

Cosine value ranges between -1 to 1. If it's close to 1, they have positive relation, if -1, then negative relation and if 0, then no relation.



- Pearson's Correlation: It tells us how much two items are correlated. Higher the correlation, more will be the similarity. Pearson's correlation can be calculated between u and v using

$$sim(u,v) = \frac{\sum(r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum(r_{ui} - \bar{r}_u)^2}\sqrt{\sum(r_{vi} - \bar{r}_v)^2}}$$

here

r_{ui} = rating given by user (u) to item (i)

r_{vi} = rating given by user (v) to item (i)

r_v (mean) =mean of al rating given by user (v)

A major drawback of this algorithm is it's a very narrow way of building an engine. This is because it is limited to recommending items that are of the same type. Also 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.

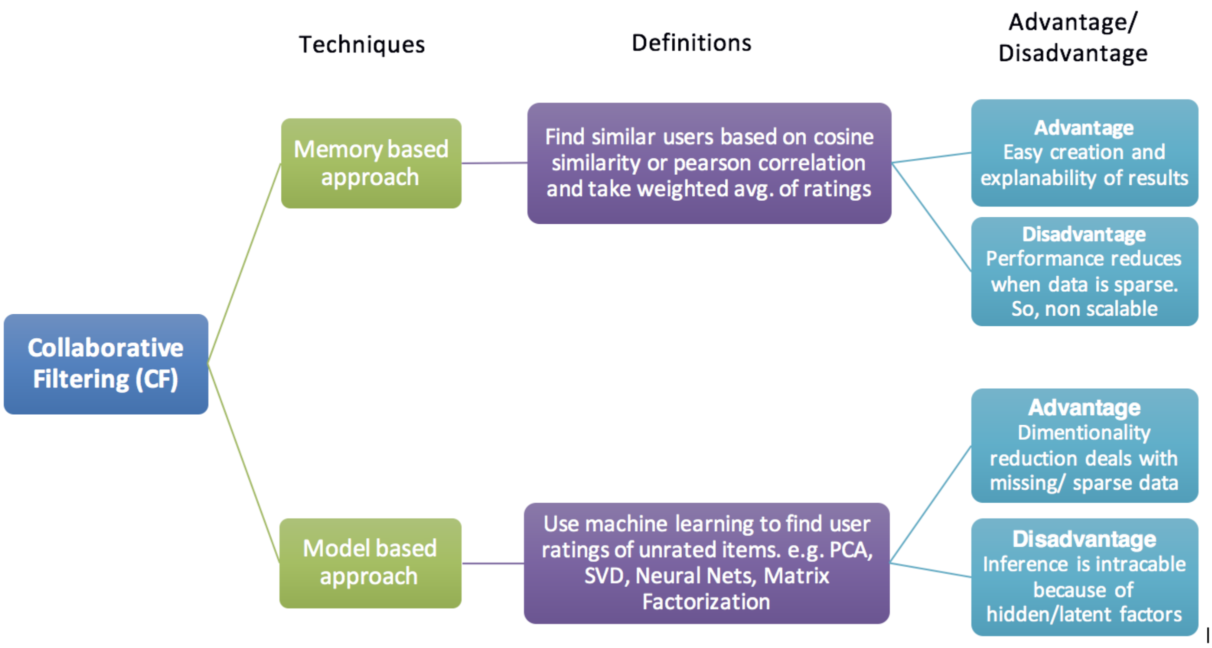
Collaborative filtering

Collaborative filtering is a technique that can filter out items that a user might like on the basis of reactions by similar users. It works by searching a large group of people and finding a smaller set of users with tastes similar to a particular user. In it the behavior of a group of users is used to make recommendations to other users.

Collaborative is based on the preference of other users. The main idea that rules collaborative methods is that these past user-item interactions are sufficient to detect similar users and/or similar items and make predictions based on these estimated proximities.

A simple example would be recommending a movie to a user based on the fact that their friend liked the movie.

Types of Collaborative Filtering Techniques



Memory Based Collaborative Filtering

Memory Based Collaborative Filtering approaches can be divided into two main categories

- User-item Filtering

“Users who are similar to you also liked …”. A user-item filtering takes a particular user, find users that are similar to that user based on similarity of ratings, and recommend items that those similar users liked.

For example, if Derrick and Dennis like the same movies and a new movie comes out that Derick likes, then we can recommend that movie to Dennis because Derrick and Dennis seem to like the same movies.

- Item-item Filtering

"Users who liked this item also liked …". In contrast, item-item filtering will take an item, find users who liked that item, and find other items that those users or similar users also liked. It takes items and outputs other items as recommendations.

For example, if users A, B and C gave a 5-star rating to books X and Y then when a user D buys book Y they also get a recommendation to purchase book X because the system identifies book X and Y as similar based on the ratings of users A, B and C.

Model Based Collaborative Filtering

In this approach models are developed using machine learning algorithms to predict user's rating of unrated items.