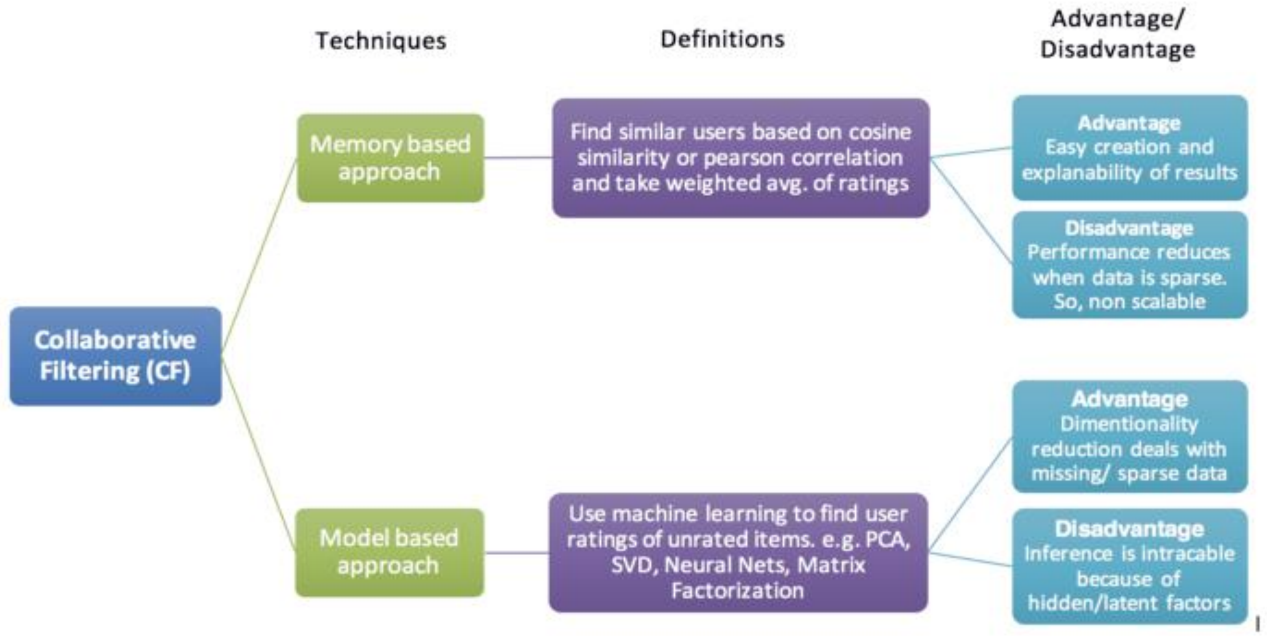
Collaborative filtering: applications and main challenges

Introduction - A recommender system makes prediction based on users' historical behaviors. Specifically, it's to predict user preference for a set of items based on experience. To build a recommender system, the most two popular approaches are Content-based and Collaborative Filtering.

Collaborative Filtering - Collaborative filtering is an early example of how algorithms can leverage data from the crowd. Information from a lot of people online is collected and used to generate personalized suggestions for any user. Because it's based on historical data, the core assumption here is that the users who have agreed in the past tend to also agree in the future. In terms of user preference, it usually expressed by two categories. Explicit Rating is a rate given by a user to an item on a sliding scale, like 5 stars for Titanic. This is the most direct feedback from users to show how much they like an item. Implicit Rating suggests user's preference indirectly, such as page views, clicks, purchase records, whether listen to a music track, and so on.

There are two types of CF systems:

- **Model-based systems**: Model-based systems use machine learning models to learn and predict ratings. Ratings can be binary, or a real valued number. Hence, given a user and a product, the system predicts the rating the user would give the product. Different algorithms can be used to perform these predictions, such as logistic regression, neural networks, SVMs, or Bayesian networks. The most successful method for model-based recommender systems, however, has been matrix factorization methods.
- **Memory-based systems:** Memory-based systems use user rating data to compute the similarity between users or items and make recommendations. This was an early approach used in many commercial systems; it is effective and easy to implement. Typical examples of this approach are neighborhood-based CF [90] and item-based/user-based top-N recommendations


Techniques
Definitions
Advantage/
Disadvantage
Collaborative Filtering (CF)
Memory based approach
Find similar users based on cosine similarity or pearson correlation and take weighted avg. of ratings
Advantage
Easy creation and explanability of results
Disadvantage
Performance reduces when data is sparse. So, non scalable
Model based approach
Use machine learning to find user ratings of unrated items. e.g. PCA, SVD, Neural Nets, Matrix Factorization
Advantage
Dimentionality reduction deals with missing/ sparse data
Disadvantage
Inference is intracable because of hidden/latent factors

Example - Friend recommenders in social networking websites. Facebook features a "People You May Know" section, which is essentially a recommendation of people to add as friends. This uses social network data to guess at what edges might be missing from a network. For example, if you are friends with 9 out of 10 densely connected people, it is likely that you are also friends with the 10th person.

Collaborative Filtering Method - The collaborative filtering techniques can be further classified into **Neighborhood methods** and **Latent factor models**.

The standard method of Collaborative Filtering is known as Nearest Neighborhood algorithm. Neighborhood methods predict the user-item preferences by first finding a cohort of users or items which are similar to the user or the item whose ratings are to be predicted. The latent factor models tend to explain the ratings by representing both the users and the items on a set of common factors. This set of factors is inferred from the known user-item ratings matrix

There is user-based collaborative filtering and item-based collaborative filtering.

- **In user-based filtering** basically, the idea is to find the most similar users to your target user (nearest neighbors) and weight their ratings of an item as the prediction of the rating of this item for target user. We can calculate similarity using Pearson Correlation and Cosine Similarity.
- **In item-based filtering** we say two items are similar when they received similar ratings from a same user. Then, we will make prediction for a target user on an item by calculating weighted average of ratings on most X similar items from this user.

Applications - We see recommendation systems all around us. These systems are personalizing our web experience, telling us what to buy (Amazon, eBay, BestBuy etc.), which movies to watch (Netflix, Hulu, Disney), whom to be friends with (Facebook, Instagram, twitter), which songs to listen (Spotify) etc. These recommendation systems leverage our shopping/ watching/ listening patterns and predict what we could like in future based on our behavior patterns.

Challenges: Below are few challenges for collaborative filtering.

1. **Data sparsity -** Many commercial recommender systems are based on large datasets. As a result, the user-item matrix used for collaborative filtering could be extremely large and sparse, which brings about the challenges in the performances of the recommendation. One typical problem caused by the data sparsity is the cold start problem. As collaborative filtering methods recommend items based on users' past preferences, new users will need to rate enough items to enable the system to capture their preferences accurately and thus provides reliable recommendations.

2. **Scalability -** As the numbers of users and items grow, traditional CF algorithms will suffer serious scalability problems

3. **Synonyms -** Synonyms refers to the tendency of a number of the same or very similar items to have different names or entries. Most recommender systems are unable to discover this latent association and thus treat these products differently.

4. Shilling attacks - In a recommendation system where everyone can give the ratings, people may give many positive ratings for their own items and negative ratings for their competitors'. It is often necessary for the collaborative filtering systems to introduce precautions to discourage such manipulations.

Conclusion: Collaborative Filtering provides strong predictive power for recommender systems and requires the least information at the same time. However, it has a few limitations in some situations.

First, the underlying tastes expressed by latent features are not interpretable because there are no content-related properties of metadata. For movie example, it doesn't necessarily to be genre like Sci-Fi in my example. It can be how motivational the soundtrack is, how good the plot is, and so on. Collaborative Filtering is lack of transparency and explain ability of this level of information.

On the other hand, Collaborative Filtering is faced with cold start. When a new item coming in, until it must be rated by substantial number of users, the model is not able to make any personalized recommendations. Similarly, for items from the tail that didn't get too much data, the model tends to give less weight on them and have popularity bias by recommending more popular items.

It's usually a good idea to have ensemble algorithms to build a more comprehensive machine learning model such as combining content-based filtering by adding some dimensions of keywords that are explainable, but we should always consider the tradeoff between model/computational complexity and the effectiveness of performance improvement.

References :

https://www.sciencedirect.com/topics/computer-science/collaborative-filtering

https://towardsdatascience.com/various-implementations-of-collaborative-filtering-100385c6dfe0

https://towardsdatascience.com/intro-to-recommender-system-collaborative-filtering-64a238194a26