Collaborative Filtering: Applications & Main Challenges

Collaborative filtering filters information by utilizing the system's interactions and data collected from other users. It is based on the assumption that people who agreed on certain items evaluations are likely to agree again in the future. The idea is simple: when we want to watch a new movie, we often ask our friends for recommendations. We naturally place more trust in recommendations from friends who have similar tastes to our own. The so-called similarity index-based technique is used by the majority of collaborative filtering systems. A number of users are chosen in the neighborhood-based approach based on their similarity to the active user. A weighted average of the ratings of the selected users is used to infer the active user. The relationship between users and items is the focus of collaborative-filtering systems. The similarity of items is determined by the similarity of their ratings by users who rated both items.
There are two classes of Collaborative Filtering:

- User-based, which measures the similarity between target users and other users.
- Item-based, which measures the similarity between the items that target users rate or interact with and other items.

All around us, we see the use of recommendation systems. These systems personalize our web experience by telling us what to buy (Amazon), what movies to watch (Netflix), who to friend (Facebook), what songs to listen to (Spotify), and so on. These recommendation systems use our shopping, watching, and listening patterns to predict what we might like in the future. The most fundamental models for recommendation systems are collaborative filtering models, which assume that people like things that are similar to other things they like and things that are liked by other people with similar tastes.

Types of collaborative filtering techniques

- Memory based :
  There are two types of Memory-Based Collaborative Filtering approaches: user-item filtering and item-item filtering. A user-item filtering algorithm takes a specific user, finds users who are similar to that user based on rating similarity, and recommends items that those similar users liked. Item-item filtering, on the other hand, will take an item and find users who liked it, as well as other items that those users or similar users liked. It takes items and generates recommendations for other items.
- Model based :
  Models are developed in this approach using various data mining and machine learning algorithms to predict user's ratings of unrated items. Many model-based CF algorithms exist like Bayesian networks, clustering models, and latent semantic models such as singular value decomposition, probabilistic latent semantic analysis, multiple multiplicative factors, latent Dirichlet allocation, and Markov decision process based models are all examples of latent semantic models. Dimensionality reduction methods are mostly used as a complementary technique to improve the robustness and accuracy of the memory-based approach in this approach. Methods such as singular value decomposition and principal component analysis, also known as latent factor models,

compress user-item matrices into a low-dimensional representation in terms of latent factors. One advantage of using this approach is that we will be dealing with a much smaller matrix in lower-dimensional space rather than a high-dimensional matrix with a large number of missing values. A simplified presentation could be used for either the user-based or item-based neighborhood algorithms discussed in the previous section. This paradigm has a number of advantages. It handles the original matrix's sparsity better than memory-based ones. Comparing similarity on the resulting matrix is also much more scalable, particularly when dealing with large sparse datasets.

- Hybrid :
  Several applications combine memory-based and model-based CF algorithms. These improve prediction performance by overcoming the limitations of native CF approaches. Importantly, they overcome CF issues such as sparsity and information loss. However, they are more complicated and costly to implement. Most commercial recommender systems are hybrid, such as Google's news recommender system.
- Deep Learning :
  A number of neural and deep-learning techniques have been proposed in recent years. Some use a nonlinear neural architecture to generalize traditional Matrix factorization algorithms, while others use new model types such as Variational Autoencoders. While deep learning has been used in many different scenarios, such as context-awareness, sequence-awareness, social tagging, and so on, its true effectiveness in a simple collaborative recommendation scenario has been called into question. A systematic review of publications using deep learning or neural methods to solve the top-k recommendation problem and published in top conferences (SIGIR, KDD, WWW, RecSys) revealed that less than 40% of articles are reproducible on average, with as few as 14% in some conferences.

Unlike the traditional mainstream media model, collaboratively filtered social media can have a large number of editors, and content improves as the number of participants grows. Reddit, YouTube, and Last.fm are common examples of collaborative filtering-based media. One application scenario for collaborative filtering is to recommend interesting or popular information based on community judgment. As an example, as stories are "voted up" (rated positively) by the community, they appear on the front page of Reddit. As the community grows in size and diversity, the promoted stories will better reflect the general interest of the community members. Wikipedia is another example of collaborative filtering in action. Volunteers help the encyclopedia by distinguishing between facts and lies. Another feature of collaborative filtering systems is the ability to generate more personalized recommendations by analyzing information from a specific user's past activity or the history of other users deemed to have similar tastes to a given user. These resources are used for user profiling and to help the site recommend content to individual users. The more a given user interacts with the system, the more accurate the recommendations become as the system gathers data to improve its model of that user.

A collaborative filtering system does not always succeed in matching content to one's preferences. Unless the platform achieves unusually high levels of diversity and independence of opinion, one point of view will always predominate over another in a given community. The introduction of new users or new items, as in the personalized recommendation scenario, can cause the cold start problem because there will be insufficient data on these new entries for collaborative filtering to work correctly. To make appropriate recommendations to a new user, the system must first learn about the user's preferences by analyzing previous voting or rating activities. Before a new item can be recommended, the collaborative filtering system requires a large number of users to rate it.

For CF systems, producing high-quality predictions or recommendations depends on how well they address the challenges, which are characteristics of CF tasks as well.
Some of the challenges faced during collaborative filtering are:

- Data sparsity :
  The data sparsity challenge appears in several situations, most notably the cold start problem, which occurs when a new user or item has just entered the system and it is difficult to find similar ones due to a lack of information (the cold start problem is also known as the new user problem or new item problem in some literature. New items cannot be recommended until some users rate it, and new users are unlikely given good recommendations because of the lack of their rating or purchase history. Coverage is the percentage of items for which the algorithm could make recommendations. When the number of user ratings is very small in comparison to the large number of items in the system, the recommender system may be unable to generate recommendations for them.
- Scalability :
  When the number of existing users and items increases dramatically, traditional CF algorithms will face serious scalability issues, with computational resources exceeding practical or acceptable levels. A CF algorithm with the complexity of O(n) is already too large with tens of millions of customers (M) and millions of distinct catalog items (N). Furthermore, many systems must respond immediately to online requests and make recommendations for all users, regardless of their purchase or rating history, necessitating a high scalability of a CF system. Dimensionality reduction techniques such as SVD can address the scalability issue and produce high-quality recommendations quickly, but they require costly matrix factorization steps.
- Synonyms :
  The tendency for a number of the same or very similar items to have different names or entries is referred to as synonymy. Because most recommender systems are unable to detect this latent association, these products are treated differently. For example, the seemingly disparate items "children movie" and "children film" are actually the same item, but memory-based CF systems would find no match to compute similarity. The degree of variation in descriptive term usage is higher than commonly assumed. The prevalence of synonyms reduces the performance of CF systems in terms of recommendation. The SVD techniques, particularly the Latent Semantic Indexing (LSI) method, are capable of dealing with the synonymy problems.
- Gray sheep : Gray sheep are users who consistently agree or disagree with no group of people and thus do not benefit from collaborative filtering. The opposite group is the

black sheep, whose idiosyncratic tastes make recommendations nearly impossible. Although this is a recommender system failure, non-electronic recommenders also have significant problems in these situations, so black sheep is an acceptable failure. Claypool et al. provided a hybrid approach combining content-based and CF recommendations by basing a prediction on a weighted average of the content-based prediction and the CF prediction.

- Shilling attacks : People may give a lot of positive recommendations for their own materials and a lot of negative recommendations for their competitors in cases where anyone can provide recommendations. It is preferable for CF systems to implement safeguards that discourage this type of phenomenon. Recently, shilling attacks models for collaborative filtering systems were discovered and their effectiveness was investigated.

The main issues with collaborative filtering recommendation algorithms are data sparsity, cold start, and robustness, as well as recommendation efficiency in large data environments. Researchers have proposed a variety of solutions, the most common of which is to incorporate methods from other fields, and collaborative filtering interdisciplinary research has progressed. With the rapid growth of information on the Internet, improving collaborative filtering recommendations based on large data sets is a daunting task.