CS410 Technology Review: Collaborative Filtering

Parth Maheshkumar Patel (pp32)

Introduction

Recommender systems have become more and more prevalent in our lives over the past few decades with the emergence of websites like Youtube, Amazon, Netflix, and many others. Recommender systems are becoming a necessary part of our daily online experiences, whether it be in e-commerce (which suggests articles to customers that might be of interest) or online advertising (which suggests to users the appropriate contents, matching their tastes). In general, recommender systems are algorithms that suggest relevant items to users (items being movies to watch, text to read, products to buy or anything else depending on industries).

Recommender systems are extremely important in some industries because they can generate a significant amount of revenue when they are efficient or serve as a way to differentiate significantly from competitors. As evidence of the importance of recommender systems, a few years ago, Netflix organized a challenge (the "Netflix prize") in which the goal was to create a recommender system that outperformed its own algorithm, with a prize of $1 million to be won.

Recommendation algorithms can be divided in two paradigms:

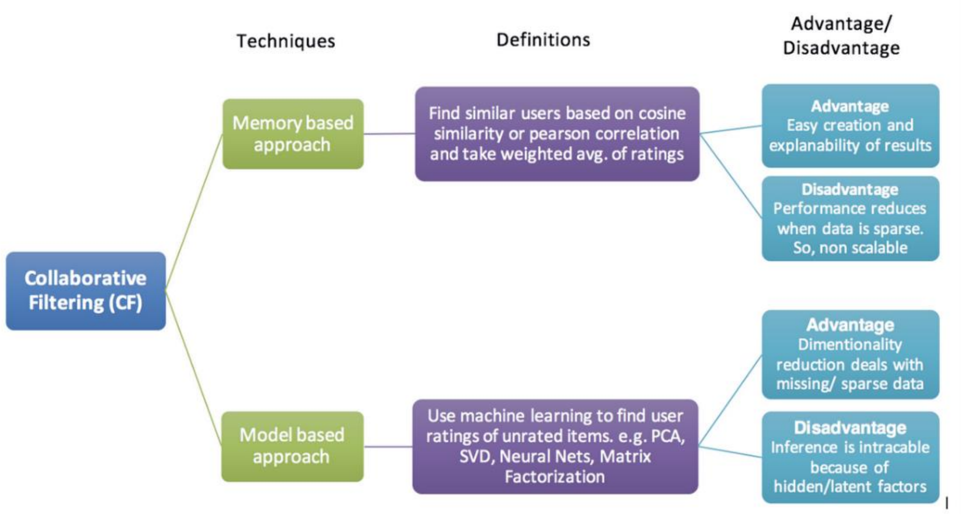
1. **Content-based filtering**, which uses similarities between items to recommend items similar to what the user likes, based on their previous actions or explicit feedback.

2. **Collaborative filtering**, which uses similarities between users to recommend items similar to those liked by similar users.

This technology review will focus on collaborative filtering, algorithms, application and major challenges.

Collaborative Filtering Algorithms

The class of collaborative filtering algorithms is divided into two sub-categories: memory-based approaches and model-based approaches.



1. Memory Based Approach

Memory-based algorithms directly use historical data of interactions between users and items (e.g. rating history, purchase history, watch history) to find similar users and predict how much a user will like an item based on how much their "nearest neighbour" users have liked it.

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. The idea behind User-Item Collaborative Filtering is that "Users who are similar to you also liked…"

b) **Item-item filtering** algorithm 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. The idea behind Item-Item Collaborative Filtering is that "Users who liked this item also liked…"

The primary distinction between the model-based techniques and the memory-based approach is that no parameters are learned using gradient descent (or any other optimization algorithm). Only arithmetic operations, such as Cosine similarity or Pearson correlation coefficients, are used to determine the closest user or item. It is a simple approach to use because no training or optimization is required. However, its performance suffers when we have sparse data, limiting its scalability for the majority of real-world problems.

2. Model Based Approach

Model-based collaborative filtering algorithms create a machine learning model from historical data of interactions between users and items, which is then used to make recommendations. The entire dataset is only required in the model building stage of model-based algorithms, and it is not required to make each prediction of whether a user will like an item or not. This category of algorithms takes a probabilistic approach, imagining the collaborative filtering process as computing the expected value of a user prediction based on his/her ratings on other items. As a result, given a user and a product, the system predicts the user's rating of the product.

These predictions can be made using a variety of algorithms, including logistic regression, neural networks, SVMs, and Bayesian networks. However, the matrix factorization method has proven to be the most successful method for model-based recommender systems.

- For the collaborative filtering problem, the Bayesian network model develops a probabilistic model.
- The clustering model approaches collaborative filtering as a classification problem, clustering similar users in the same class and estimating the probability that a

particular user is in a specific class C, from which the conditional probability of ratings is computed.

- The rule-based approach finds associations between co-purchased items using association rule discovery algorithms and then generates item recommendations based on the strength of the association between items.

Applications

We see recommendation systems everywhere. These systems personalize our web experience by telling us what to buy (Amazon, eBay, BestBuy, etc.), what movies to watch (Netflix, Hulu, Disney), who to friend (Facebook, Instagram, Twitter), what songs to listen to (Spotify), and so on. These recommendation systems take advantage of our shopping, watching, and listening habits to predict what we might like in the future.

Challenges of Collaborative Filtering

User-based collaborative filtering systems have been very successful, but their widespread use has revealed some significant challenges, such as:

1. **Data Sparsity:**

   Active users may have bought less than 1% of the items (1% of 2 million books equals 20,000 books). As a result, a recommender system based on nearest neighbor algorithms may be unable to recommend any items to a specific user. As a result, recommendations may be inaccurate.

2. **Scalability:**

   The amount of processing needed for nearest neighbor algorithms increases with the quantity of users and items. With millions of users and items, a typical web-based recommender system using traditional algorithms will have serious scalability issues.

3. **Cold start Problem:**

   One challenge faced by both memory-based and model-based collaborative filtering systems is known as the cold-start problem, which occurs when a new user or item is added to the system and there is no historical data available. For example, if a new user joins Netflix and hasn't watched or rated anything yet, no data about their likes/dislikes is available.

   Active learning is a popular technique for dealing with the cold-start issue. This method involves asking a new user to rate a variety of items in order to learn more about their preferences. These first items that the user rates can be chosen in a number of ways, such as choosing "controversial" items with a wide range of ratings or favoring more "well-known" items that the new user is more likely to be familiar with. A hybrid recommender system, which combines collaborative filtering and content-based filtering, can handle new items without any prior user interaction history by making recommendations based on known attributes of the items (e.g., film genre, actors, etc.).

Conclusion

A useful and necessary function of any recommendation system is collaborative filtering algorithms. This filtering method makes recommendations based on what other users have rated as similar. Netflix's video streaming platform employs collaborative filtering, with the recommendation system accounting for 80% of the company's total view time. Many contemporary recommender systems integrate collaborative and content-based approaches into a hybrid system to address the cold-start problem, which is a common problem for both types of collaborative filtering systems. We will undoubtedly see improved filtering techniques in the coming years that will overcome many limitations.

References

1. https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada
2. https://developers.google.com/machine-learning/recommendation/content-based/basics
3. https://developers.google.com/machine-learning/recommendation/collaborative/basics
4. https://towardsdatascience.com/various-implementations-of-collaborative-filtering-100385c6dfe0
5. Model-Based Collaborative Filtering as a Defense against Profile Injection Attacks (aaai.org)
6. http://www10.org/cdrom/papers/519/node8.html
7. http://www10.org/cdrom/papers/519/node9.html
8. https://www.researchgate.net/publication/332511384_Cold_Start_Solutions_For_Recommendation_Systems