Report:

Collaborative Filtering:

Collaborative filtering (CF) is a popular recommendation algorithm that bases its predictions and recommendations on the ratings or behaviour of other users in the system. The fundamental assumption behind this method is that other users' opinions can be selected and aggregated in such a way as to provide a reasonable prediction of the active user's preference. Intuitively, they assume that, if users agree about the quality or relevance of some items, then they will likely agree about other items — if a group of users likes the same things as Mary, then Mary is likely to like the things they like which she hasn't yet seen. There are other methods for performing recommendation, such as finding items similar to the items liked by a user using textual similarity in metadata (content-based filtering or CBF). The focus of this survey is on collaborative filtering methods, although content-based filtering will enter our discussion at times when it is relevant to overcoming a particular recommender system difficulty. The majority of collaborative filtering algorithms in service today, including all algorithms detailed in this section, operate by first generating predictions of the user's preference and then produce their recommendations by ranking candidate items by predicted preferences.

There are mainly 6 types of popular collaborative filtering algorithms.

a) Baseline predictions
b) User-to-User Collaborative Filtering
c) Item-to-item Collaborative Filtering
d) Dimensionality Reduction
e) Probabilistic method
f) Hybrid recommenders

We have, in our project, mainly drawn inspiration from User-to-User Collaborative Filtering and Item-To-Item Collaborative Filtering.

User-to-User Collaborative Filtering

User–user collaborative filtering, also known as k-NN collaborative filtering, was the first of the automated CF methods. It was first introduced in the GroupLens Usenet article recommender.

Ringo music recommender and the BellCore video recommender also used user-user CF or variants thereof. User–user CF is a straightforward algorithmic interpretation of the core premise of collaborative filtering: find other users whose past rating behaviour is similar to that of the current user and use their ratings on other items to predict what the current user will like. To predict Mary's preference for an item she has not rated, user–user CF looks for other users who have high agreement with Mary on the items they have both rated. These users' ratings for the item in question are then weighted by their level of agreement with Mary's ratings to predict Mary's preference. Besides the rating matrix R, a user–user CF system requires a similarity function s: $U \times U \rightarrow R$ computing the similarity between two users and a method for using similarities and ratings to generate predictions.

Computing User Similarity Using Pearson correlation:

This method computes the statistical correlation (Pearson's r) between two user's common ratings to determine their similarity. GroupLens and BellCore both used this method. The correlation is computed by the following:

$$s(u,v) = \frac{\sum \in Iu\, i \cap Iv\ (ru,i - \bar{r}u\,)(rv,i - \bar{r}v)}{\sqrt{\sum \in Iu\, i \cap Iv\ (ru,i - \bar{r}u\,)2}\ \ \sqrt{\sum \in Iu\, i \cap Iv\ (rv,i - \bar{r}v)2}}$$

Pearson correlation suffers from computing high similarity between users with few ratings in common. This can be alleviated by setting a threshold on the number of co-rated items necessary for full agreement (correlation of 1) and scaling the similarity when the number of co-rated items falls below this threshold. Experiments have shown a threshold value of 50 to be useful in improving prediction accuracy, and the threshold can be applied by multiplying the similarity function by min value.

Item-to-Item Collaborative Filtering:

User–user collaborative filtering, while effective, suffers from scalability problems as the user base grows. Searching for the neighbours of a user is an O(|U|) operation (or worse, depending on how similarities are computing — directly computing most similarity functions against all other users is linear in the total number of ratings). To extend collaborative filtering to large user bases and facilitate deployment on e-commerce sites, it was necessary to develop more scalable algorithms. Item–item collaborative filtering, also called item-based collaborative filtering, takes a major step in this direction and is one of the most widely deployed collaborative filtering techniques today. Item–item collaborative filtering was first described in the literature by Sarwar et al. and Karypis, although a version of it seems to have been used by Amazon.com at this time.

Rather than using similarities between users' rating behaviour to predict preferences, item–item CF uses similarities between the rating patterns of items. If two items tend to have the same users like and dislike them, then they are similar and users are expected to have similar preferences for similar items. In its overall structure, therefore, this method is similar to earlier content-based approaches to recommendation and personalization, but item similarity is deduced from user preference patterns rather than extracted from item data. In its raw form, item–item CF does not fix anything: it is still necessary to find the most similar items (again solving the k-NN problem) to generate predictions and recommendations. In a system that has more users than items, it allows the neighbourhood-finding to be amongst the smaller of the two dimensions, but this is a small gain. It provides major performance gains by lending itself well to pre-computing the similarity matrix. As a user rates and re-rates items, their rating vector will change along with their similarity to other users. Finding similar users in advance is therefore complicated: a user's neighbourhood is determined not only by their ratings but also by the ratings of other users, so their neighbourhood can change as a result of new ratings supplied by any user in the system. For this reason, most user–user CF systems find neighbourhood at the time when predictions or recommendations are needed. In systems with a sufficiently high user to item ratio, however, one user adding or changing ratings is unlikely to significantly change the similarity between two items, particularly when the items have many ratings. Therefore, it is reasonable to pre-compute similarities between items in an item–item similarity matrix. The rows of this matrix can even be truncated to only store the k most similar items. As users change ratings, this data will become slightly stale, but the users will likely still receive good recommendations and the data can be fully updated by re-computing the similarities during a low-load time for the system. Item–item CF generates predictions by using the user's own ratings for other items combined with those items' similarities to the target item, rather than other users' ratings and user similarities as in user–user CF. Similar to user–user CF, the recommender system needs a similarity function, this time s: I×I → R, and a method to generate predictions from ratings and similarities.

Computing Item Similarity using Cosine similarity

Cosine similarity between item rating vectors is the most popular similarity metric, as it is simple, fast, and produces good predictive accuracy.

$$s(i,j) = \frac{ri.r}{||ri||_2 \, ||r\,j\,||_2}$$