A Comparison Analysis of Collaborative Filtering Techniques for Recommeder Systems

Amarajyothi Aramanda, Saifullah Md. Abdul, and Radha Vedala

Abstract In E-commerce environment, a recommender system recommend products of interest to its users. Several techniques have been proposed in the recommender systems. One of the popular techniques is collaborative filtering. Generally, the collaborative filtering technique is employed to give personalized recommendations for any given user by analyzing the past activities of the user and users similar to him/her. The memory-based and model-based collaborative filtering techniques are two different models which address the challenges such as quality, scalability, sparsity, and cold start, etc. In this paper, we conduct a review of traditional and state-of-art techniques on how they address the different challenges. We also provide the comparison results of some of the techniques.

Keywords E-commerce · Recommender system · Recommendation · Collaborative filtering · Memory-based collaborative filtering · Model-based collaborative filtering

1 Introduction

E-commerce systems such as amazon, flipkart and ebay, etc., are very popular in the present days' world. Users can gather information about the products of their interest in E-commerce sites without moving from their place and without visiting physical store. However, user faces difficulty to choose the right product from huge

A. Aramanda (✉) · S. Md. Abdul
SCIS, University of Hyderabad, Hyderabad 500 046, Telangana, India
e-mail: 18mcpc07@uohyd.ac.in

S. Md. Abdul
e-mail: saifullah@uohyd.ac.in

A. Aramanda · R. Vedala
Institute for Development and Research in Banking Technology, Castle Hills Road no: 1, Masab Tank, Hyderabad 500 057, Telangana, India
e-mail: vradha@idrbt.ac.in

A. Kumar and S. Mozar (eds.), *ICCCE 2020*, Lecture Notes in Electrical Engineering 698, https://doi.org/10.1007/978-981-15-7961-5_9

number of products. To address this, the Recommender System (RS) has been proposed to help him/her [1].

The RS provides recommendations for a set of products/items to its users. Generating right recommendations is a major challenge in RS. To address this, several approaches have been proposed, such as content-based, collaborative and hybrid filtering [4, 12, 13]. In Collaborative Filtering (CF) technique, the items are suggested to a user based on his/her preference history and items that are preferred by similar users. The first CF is Tapestry [13] used for email filtering. The CF is a widely adopted technique for personalized RS. There are several challenges for CF, such as quality, scalability, sparsity and cold start, etc. Techniques like, memory-based CF and model-based CF are popular in the literature to address these challenges.

In this paper, we conduct a review of traditional and state-of-art techniques on how they address the above mentioned challenges. We made an effort to study the different CF techniques, analyze each technique's strengths and situation in which it gives good results. We also provide the comparative analysis of some of the techniques.

The organization of the paper is as follows. In Sect. 2, we provide the overview about CF. In Sect. 3, we describe CF techniques, challenges and metrics. In Sect. 4, We explain the comparative analysis of CF techniques. Finally, we conclude the paper in Sect. 5.

2 Collaborative Filtering Overview

The RS must collect the data about users' preferences to build a model for personalized recommendations [9]. The data gathered for this can be either explicit or implicit. Explicit data can be user's preferences, 1–5 stars, like/dislike, or textual feedback. Implicit data can be derived by monitoring user's activities such as books read, songs heard, items purchased and websites visited. To know the performance of CF techniques, we need to experiment on a dataset. Most widely used datasets are MovieLens and BookLens datasets from GroupLens. Other than GroupLens, there are Netflix, Jester, BookCrossing, Delicious, MSD, and Amazon [12].

Consider a scenario that, there are n number of users, set $U = \{user_1, user_2, \cdots, user_n\}$ and m number of items, set $I = \{item_1, item_2, \cdots, item_m\}$. The item could be a book in amazon.com, URL in del.icio.us, a song in last.fm, a photo on flicker.com, or any product in E-commerce sites, etc. We can represent users and items data as $n \times m$ matrix R and the value of element $x_{i,j}$ in R, is a preference of i^{th} user on a j^{th} item. The preference can be 1 or any value on a given scale (like 1–5 from poor to excellent) if a user prefers an item, otherwise zero. Now, the RS suggests *top-N* items or predicts the rating to the new/active user on item(s) that are not yet bought by him/her.

The basic CF process can be viewed as 3 steps process [15]. (i) The *data representation* is the first step to preprocess the data by removing sparsity, removing synonymy and reducing the dimensions of data. Further, data is also normalized to prepare for filtering process. (ii) The *neighborhood formation* is the second step, to compute similar interested users, called neighbors. The neighborhood is a collection of similar users. The neighbors are computed by finding similarities/correlations between user's interests/preferences using proximity measures. Proximity measures used in CF are cosine-based similarity, Pearson correlation and Spearman's rank correlation coefficient, etc. [2, 3, 6, 7]. Depending on the type of data and need suitable proximity measure is chosen. The similarity is computed either between users or between items. The correlations between users is called as *user-centric* correlations and is represented as user × user correlation matrix, *UCR*. The correlation between items is called *item-centric* correlations and is represented as item × item correlation matrix, *ICR*. (iii) The *recommendation generation* is third step to generate the recommendations by either predicting the preference of the user on a particular item or listing the *top-N* popular items to the user. The *prediction*, $P_{a,b}$, denotes a numerical value, prediction of likeliness score of user a on a particular item b. The *top-N* recommendations are list of N not yet bought items that the active user will like the most.

3 Collaborative Filtering Techniques

CF techniques can be categorized according to the way they retrieve information from repositories. They are memory-based CF and model-based CF [3, 7]. The memory-based CF techniques use the entire user-item data to compute similarities on the fly and predict or list the *top-N* items for an active user. The model-based CF techniques use precomputed similarities of users/items stored in an offline repository and generate recommendations for an active user. In this section, we explain memory-based CF, model-based CF, challenges and metrics.

3.1 Memory-Based Collaborative Filtering

The memory-based CF techniques further classified as *User-User* CF and *Item-Item* CF. Now, we explain these techniques in the following.

User-User CF: The basic assumption for this technique: many users show interest to buy the items bought by his/her neighbors [8]. This method takes either implicit or explicit feedback matrix R as input. This method can use either low dimensional data representation or high dimensional data representation. The neighborhood is formed from *UCR* using K-nearest-neighbors (KNN) algorithm. The recommendations are generated from neighborhood. The *prediction* can be done by weighted average of nearest neighbors. The neighborhood for active user a are l neighbors, represented as set, $NB_a = \{nb_1, nb_2, \cdots, nb_l\}$. Let $NB_{a,b}$ is subset of NB_a containing all users in NB_a who have rated item b. The weights are proximity values from *UCR*. Each element, $w_{a,i} \in UCR$, is proximity value between user a and user i, $i \in NB_{a,b}$. The $x_{i,b}$ is rating by user i on item b. The *prediction* for user a on item b is given in an Eq. 1.

$$P_{a,b} = \frac{\sum_{i \in NB_{a,b}} x_{ib} * w_{a,i}}{\sum_{i \in NB_{a,b}} |w_{a,i}|} \tag{1}$$

The Eq. 1 can be enhanced by considering user's average rating μ_a, and is given in [8]. *Top-N* recommendations is an item set, $T = \{t_1, t_2, \cdots, t_N\}$, can be computed by either *most-frequent-item* recommendations or *association rule-based* recommendations [15].

Item-Item CF: The steps followed in this technique are similar to *User-User* CF technique [8]. But, *Item-Item* CF uses, similarities between items instead of users. Neighborhood is formed for each item by using *ICR*. The *prediction* on item b for user a is generated by considering the items similar to item b. Let, $INB_{a,b} = \{inb_1, inb_2, \cdots, inb_l\}$ is neighborhood for item b, rated by user a. The $w_{b,i}$ is proximity value between item b and item i, $i \in INB_{a,b}$. The *prediction* is computed as weighted average of user's ratings on the items in $INB_{a,b}$ [1, 8]. *Top-N* recommendations are computed similar to *User-User* CF except item's neighborhood is considered instead of user's neighborhood.

3.2 Model-Based Collaborative Filtering

The model-based CF techniques are neighborhood-based, latent-factor-based, Bayesianbased and association rule-based. In this section, different model-based CF techniques are discussed.

Neighborhood-based: Neighborhood-based techniques build the model by learning the neighborhood information. The neighborhood is formed using either KNN or clustering algorithms.

Item-based CF: This technique is similar to *Item-Item* CF. But, in *Item-Item* CF the neighborhood is computed on the fly, where as in *Item-based* CF neighborhood is computed offline and stored in a model [2].

Cluster-based CF: This technique forms all users into k partitions from entire user × item data using k-means clustering algorithm, in which a user belongs to only one partition. Each partition is a neighborhood and is of fixed or variable length. The recommendations are generated using neighborhood information [15].

Cluster and Nearest-neighbors-based CF (CNCF): Another variation for clustering is, CNCF technique. The dataset is divided into k partitions (clusters) using k-means algorithm, in which a user belongs to only one partition. The partitions are formed using *user-centric* proximity measure. First, build the model with $k$surrogate users derived from k-centroids of k-clusters. A surrogate user is a centroid for a cluster. Finally, recommendations are generated by finding nearest neighbors for active user from model [2].

Neighborhood with Interpolation Weights CF (NIWCF): This technique is an enhanced version of *item-based* CF. In this method, first step is to normalize the dataset by removing global effects. The global effects are the tendencies and preferences of users that change from time to time, which affects users' true preferences. In second step, neighborhood is formed using *item-centric* correlation. Unlike *item-based* CF all neighbors are not given an equal weightage. In a given set of neighbors, each neighbor is assigned with weight called interpolation weights [5]. Finally, *prediction* of rating for user on item is computed using interpolation weights.

Latent-factor-based: Latent-factor-based techniques extract latent factors from users' preference data, unlike *UCR* and *ICR*, in CF techniques. The latent factors are computed by either dimensionality reduction techniques like singular value decomposition and principal component analysis or matrix factorization techniques.

Singular Value Decomposition (SVD): This is mainly used to reduce the dimensionality by producing low-rank linear approximation [3]. Another variation of SVD is incremental SVD to minimize the computation time by using *folding-in* technique [14].

Matrix Factorization: Matrix factorization (MF) builds the model by factorizing the rating matrix into two lower-dimensional latent feature matrices [3]. The MF maps both the items and users to a joint latent factor space dimensionality [10]. The variations of MF are non-negative matrix factorization (NMF) [11] and probabilistic matrix factorization (PMF) [16].

Bayesian-based: Bayesian-based techniques are based on Na¨ıve Bayes approach. This type of techniques predict each user's unknown preferences based on conditional probability of his/her known preferences. The improved versions of Naïve Bayes approaches are available in [20].

Association rule-based: The techniques use rule-based mining to discover an association between the items [18, 21].

Other Techniques: Other than above mentioned techniques, new research towards applying neural networks to CF [22], graph-based techniques to combine the content and CF techniques [8], context-aware recommendation [19] and so on are in progress. To improve the performance, CF algorithms are combined with sentiment analysis [16].

3.3 Challenges and Metrics

The CF techniques give good results and are widely adopted in RS applications. However, they still have to defeat different challenges [13]. There are two primary challenges for CF techniques. The first challenge is *quality of recommendations*, i.e., suggest the items to users which they like most. The second challenge is *speed/scalability*, i.e., CF algorithm should be able to process millions of neighbors. Different techniques have been proposed in the literature to achieve these.

There are secondary challenges also to address in addition to above. Those are *sparsity* and *cold start*. The *sparsity* is also referred as *reduced coverage* problem. In real world scenario most of ratings/purchase data is empty. For example, as Ecommerce is still growing, users have purchased less than the 10% of items from all items available, keeping rating matrix as almost empty. This leads to poor recommendation accuracy. The *cold start* is a problem of not able to provide good recommendations due to lack of data. This issue can be a *new community* problem for new RS, a *new item* problem for newly launched item or a *new user* problem for new user [23].

Several metrics are available in the literature to evaluate the usefulness of CF techniques. These are divided into two ways: *prediction* and *top-N* recommendations. The *prediction* metrics are: Mean Absolute Error [20] and Root Mean Squared Error [5]. The *top-N* recommendations metrics are: *Precision*, *Recall*, and *F1* [17]. In addition, how good a RS is evaluated by different concepts like Utility, reliability, novelty, diversity, unexpectedness, serendipity and coverage [17].

4 Comparative Analysis

This paper mainly focused on different types of CF techniques, how they are different from each other and what challenge they have addressed, as shown in Table 1. In this table, each row explains a particular technique with information such as: technique, measure used to find similarity, learning algorithm used to build the RS, prediction method, performance of the technique, challenge addressed, and observations we made.

We have observed that, the memory-based techniques improve quality of recommendations, but take more time when the user × item matrix size keep on increasing. The model-based techniques are more scalable, i.e., it takes less time to compute the recommendations even when data size is more. But, if more users/items are added or updated immediately after precomputation, then it gives a poor prediction. In *User-User* CF, choose the appropriate size of neighbors to improve the quality of recommendations and size of neighbors vary from dataset to dataset. When the number of users are large, then *Cluster-based* CF is best suited than *Item-based* CF. If the dataset is with less sparsity, then CNCF will take less time and less space for recommendations. The CNCF is best suited to implement the RS in handheld devices. If one needs to consider the interventions among neighbors then the NIWCF is good, because in this method each neighbor interpolation weight is considered for recommendation generation.

We also observed that, the dimensionality reduction method, SVD, is better considered to address the synonymy and semi-sparsity than neighborhood-based techniques. SVD also gives poor results, in case of higher sparsity. The alternative for this is MF. The advantage of MF is very flexible to add any biases. The biases deal with any data aspects and application-oriented requirements. For example, a

Table 1 Comparison of collaborative filtering techniques

Technique	Measures	Learning algorithm	Prediction	Performance	Criteria	Observation
User-User/ Item-Item [8]	Cosine Similarity	KNN	Weighted Average	Simple to implement	Quality, Scalability	Optimal number of neighbors and low dimensionality improves the performance
Item-based [8]	Adjusted cosine similarity	KNN	weighted Average	Improved performance than memory-based	Quality, Scalability	Regression gives best results when sparsity is more
Cluster-based [15]	Pearson Correlation	K-means	Adjusted weighted average	Simple and high throughput	Quality, Scalability	Poor prediction
CNCF [2]	Pearson Correlation	K-means, KNN	Adjusted weighted average	Reduce time for online computation	Quality, Scalability	Better for low storage and processing capacity
NIWCF [5]	Adjusted cosine similarity	Interpolation weights based	Linear regression	Best results than User/Item-based CF	Quality, Scalability	Interpolation weights avoids biased prediction
SVD [14]	–	SVD	SVD prediction	More scalable and accurate	Quality, Scalability, Sparsity	Expensive
MF [10, 11, 16]	Latent Factors	MF	Latent Factors dot product	Less memory space required	Quality, Scalability, Sparsity, Cold start	Allows to integrate many crucial aspects of data
Bayesian-based [20]	Conditional probability	Bayes classification	Probability based	More accurate prediction	Quality, Scalability, Sparsity	Understanding and justifying is easy
Association rule-based [21]	Item Frequency	Association Rule	Association rule-based	More efficient to analyse user behavior	Quality, Sparsity	Improves the coverage

critical user always rates any item lower than its average rating. So, to estimate the true rating for item by the critical user, biases are added. In practice, remove all users who have given less ratings than predefined minimal value even before preprocessing of the dataset. But, these users also have some significant information about user's interests. In such cases, the PMF gives good results. When items are frequently changing and users are less frequently purchasing them, association rule-based techniques gives good results, example, an auto industry. The major issue in CF is explanation for rating can't be given. It is not easy to interpret the meaning of latent factors in MF techniques. The PMF can alleviate this problem, however, user can't understand completely. In such cases, Bayesian-based CF will give better results. Another way to handle this is combining sentiments into CF techniques to improve reliability of RS. Now a days vendors also collecting textual feedback (sentiment) from users along with rating and like/dislike information. This user feedback gives more insight about user interests. At present, the MF techniques like PMF and NMF are performing better by considering the user sentiments. Choosing best technique is depends upon the nature of dataset and need of application.

5 Conclusion

In E-commerce environment, personalized RS helps the user to choose the item from huge number of items. To build personalized RS, CF is a widely adopted technique. In this paper, we presented an overview of the CF process, memory-based and model-based techniques of CF. This paper, described the challenges of CF and explained each technique with the challenges it addressed. It also described each technique's strengths and situation in which it gives good results. Sentiment analysis further strengthens the CF techniques in giving accurate recommendations. However, there is no single technique that addresses all challenges. We are focusing on this issue as a part of future work.

References

1. Aggarwal CC (2016) Recommender systems. Springer, Cham
2. Al Mamunur Rashid, SKL, Karypis G, Riedl J (2006) ClustKNN: a highly scalable hybrid model & memory-based CF algorithm. In: Proceedings of webKDD
3. Ambulgekar H, Pathak MK, Kokare M (2018) A survey on collaborative filtering: tasks, approaches and applications. In: Proceedings of international ethical hacking conference. Springer, pp 289–300
4. Amin SA, Philips J, Tabrizi N (2019) Current trends in collaborative filtering recommendation systems. In: World congress on services. Springer, pp 46–60
5. Bell RM, Koren Y (2007) Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In: ICDM, vol 7, pp 43–52. ACM

6. Grzegorzewski P, Ziembinska P (2011) Spearman's rank correlation coefficient for vague preferences. In: International conference on FQAS. Springer, pp 342–353
7. Isinkaye F, Folajimi Y, Ojokoh B (2015) Recommendation systems: principles, methods and evaluation. Egyptian Inform J 16(3):261–273
8. Kluver D, Ekstrand MD, Konstan JA (2018) Rating-based collaborative filtering: algorithms and evaluation. In: Social information access. Springer, pp 344–390
9. Konstan JA, Riedl J (2012) Recommender systems: from algorithms to user experience. User Model User-Adap Inter 22(1–2):101–123
10. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer 42:30–37
11. Luo X, Zhou M, Xia Y, Zhu Q (2014) An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. IEEE Trans Ind Inform 10 (2):1273–1284
12. Najafabadi MK, Mohamed AH, Mahrin MN (2019) A survey on data mining techniques in recommender systems. Soft Comput 23(2):627–654
13. Patel A, Thakkar A, Bhatt N, Prajapati P (2019) Survey and evolution study focusing comparative analysis and future research direction in the field of recommendation system specific to collaborative filtering approach. In: Information and communication technology for intelligent systems. Springer, pp 155–163 (2019)
14. Sarwar B, Karypis G, Konstan J, Riedl J (2002) Incremental singular value decomposition algorithms for highly scalable recommender systems. In: Fifth international conference on computer and information science, vol 27, p 28. Citeseer
15. Sarwar BM, Karypis G, Konstan J, Riedl J (2002) Recommender systems for large-scale ecommerce: scalable neighborhood formation using clustering. In: Proceedings of the fifth international conference on computer and information technology, vol 1, pp 291–324. ACM
16. Shen RP, Zhang HR, Yu H, Min F (2019) Sentiment based matrix factorization with reliability for recommendation. Expert Syst Appl 135:249–258
17. Silveira T, Zhang M, Lin X, Liu Y, Ma S (2019) How good your recommender system is? A survey on evaluations in recommendation. Int J MLC 10(5):813–831
18. Swamy MK, Reddy PK: Improving diversity performance of association rule based recommender systems. In: Proceedings of 26th international conference on database and expert systems applications, Part I. Springer, pp 499–508 (2015)
19. Phuong TM, Phuong ND (2019) Graph-based context-aware collaborative filtering. Expert Syst Appl 126:9–19
20. Valdiviezo-Diaz P, Ortega F, Cobos E, Lara-Cabrera R (2019) A collaborative filtering approach based on Naïve Bayes classifier. IEEE Access 7:108581–108592
21. Yao L, Xu Z, Zhou X, Lev B (2019) Synergies between association rules and collaborative filtering in recommender system: an application to auto industry. In: Data science and digital business. Springer, pp 65–80
22. Zhang S, Yao L, Sun A, Tay Y (2019) Deep learning based recommender system: a survey and new perspectives. ACM Comput Surv 52(1):1–38
23. Zhu Y, Lin J, He S, Wang B, Guan Z, Liu H, Cai D (2019) Addressing the item cold-start problem by attribute-driven active learning. IEEE Trans KDE 32:631–644