

Collaborative Filtering Techniques in Recommendation Systems



Sandeep K. Raghuwanshi and R. K. Pateriya

1 Introduction

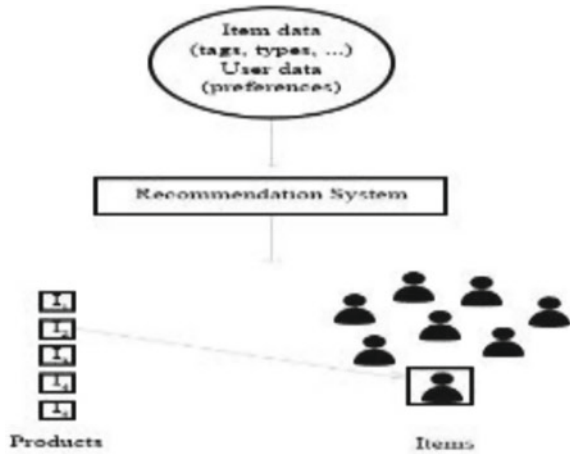
Recommendation systems are information filtering systems that urge to predict preferences that user might have for an item over other. Recommendation systems are very popular in applications like movies, books, research articles, search queries, social tags, product, financial services, restaurants, twitter pages, job, university, friends and what not. To increase product sales is the primary goal of recommendation system by bringing a relevant item to the user and thus increasing the overall profit, which covers the functional goal of recommendation system such as [1]—relevancy, serendipity and diversity. Most popular recommender systems of today are Group Lens recommender system, Amazon.com recommender system, Netflix Movie recommender system, Google News personalisation system, Facebook friend recommendations, link prediction recommender system [1].

First recommendation system was developed in 1992 by Goldberg, Nichols, Oki and Terry. This was called Tapestry which allows users to rate an item good or bad and further used keyword filtering for recommendation [2–4]. Thus, recommendation system works on available information in any form and then applies different filtering techniques to find the most appropriate choice (like the movie, show, web page, scientific literature and news that a user might have interest in). The recommendation system makes use of data mining techniques [4, 5] and prediction algorithm to find out user's interest in information, item and their other interests. Later on, several recommendation systems developed which use different filterings to lure their customers and make them feel more attended (Fig. 1).

S. K. Raghuwanshi (✉) · R. K. Pateriya
Computer Science and Engineering, Maulana Azad National Institute of Technology,
Bhopal, M. P, India
e-mail: sraghwanshi@gmail.com

R. K. Pateriya
e-mail: pateriyark@gmail.com

Fig. 1 Recommendations and recommender system



The reason for many companies care about recommendation system is to deliver actual value to their customer. Recommender systems provide a scalable way of personalising content for users in scenarios with many items. It engages many scientists, since it is a major problem of data science, a perfect intersection of software engineering, machine learning and statistics. Recommender systems are an effective tool for personalisation. Since it is based on actual user behaviour, users can make decisions directly based on the results. These systems work on unstructured and dynamically changing data because of which predictions are more specific and up to date.

Although recommender systems are application-specific and require specific filtering process, few properties must be addressed by all of them [6] like user preference, prediction accuracy, confidence score, user's trust on a recommendation system.

The rest of this paper is organised as follows: first section deals with the introduction of recommender system with their applicability and importance in present era. Section 2 presents the goals and critical challenges of recommendation systems. Section 3 presents the classification of recommendation system based on the approach to build recommendation engine. This section presents a brief introduction of content-based recommender system with collaborative techniques in detail and presents two different approaches of collaborative as memory-based and model-based systems. Section 4 presents experimental set-up to show the methods implementation and results. Section 5 gives the conclusion of work and possible future scope.

2 Goals and Critical Challenges

2.1 Goals

Recommender systems are used in different fields, from e-commerce to government applications. Most widely used application of recommendation system comes from e-commerce where companies are competing for enhancing their sales and improve user experience. By recommending interested and preferred items to users' recommender system helps merchants to increase their profit. Apart from this, the general operational and technical goals of recommendation systems are as follows:

Relevance: The most common operational goal of recommender system is to provide or recommend relevant items to the users. Users are more likely to purchase or opt in items which are of his/her preferences.

Novelty: Recommendation systems are supposed to provide novel or new items each time. The system should not repeatedly show popular items as this may also leads to reduction in user interest [7].

Serendipity: Serendipity is notion to define somewhat unexpected recommendation. It is different from novelty as it is truly surprising to user instead that they did not know about before [8].

Diversity: Recommendation system generally recommends list of similar items which increases the chance that user might not like any time. So, diversity is one of the important goals of recommender system which supports range of items for recommendation.

2.2 Challenges

Following are the critical challenges of recommendation systems:

Scalability: Most collaborative filtering techniques show poor performance with an increase in user and item base.

Grey Sheep: Grey sheep denotes the group of peoples whose opinions do not match with any group of people. These users basically create a problem in the smooth functioning of recommendation system [9].

Synonymy: Most recommender systems face problem to predict accurately the items which are same in features but have different names [10, 11].

Cold Start: New users and items suffer from accurate prediction as not much information is available to start the system [12].

Privacy Breach: Privacy has always been the biggest challenge of a recommender system. While providing an accurate prediction of user system demand to get personalised information of the user.

Shilling Attack: Recommendation is a public activity, so people get biased for their feedbacks and give millions of positive reviews for their own products or items and sometimes negative views of their competitors [13].

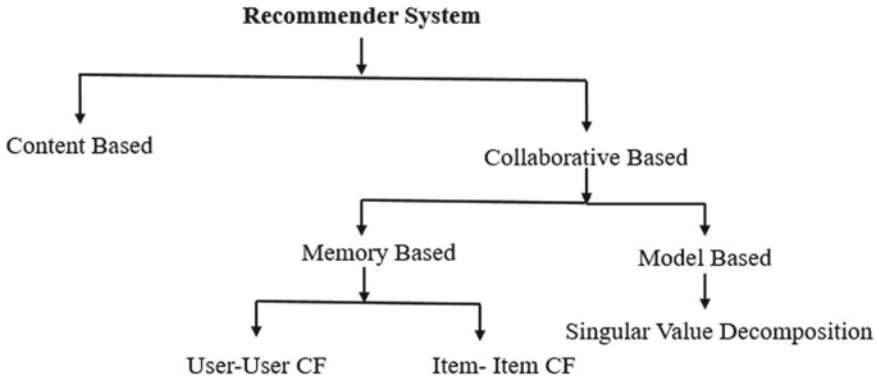


Fig. 2 Classification of recommendation system

3 Classification

Depending on the type of input used to make recommendations, recommender systems are classified into several categories. Out of which, most commonly used techniques are content-based filtering and collaborative filtering. This paper accentuates various recommendation systems used today with their pitfalls and comparative analysis of two major recommendation models (NN and Latent factor model) (Fig. 2).

3.1 Content-Based Filtering System

Content-based filtering is the most common type of filtering system used. These systems work on rating, which a user gave while creating a profile to get initial information about a user in order to avoid not knowing a new user [14]. To create user profile, two types of information are mainly focused: user's preferences and users interaction with recommendation system. It simply recommends items on the basis of comparison between the content of the item and a user's profile. Engines in these systems compare positively rated item by a user with the item he/she did not rate yet. The items with maximum similarities will then be recommended to the users. Different distances are used for measuring distances/similarities between user's choice and among items in the database (Fig. 3).

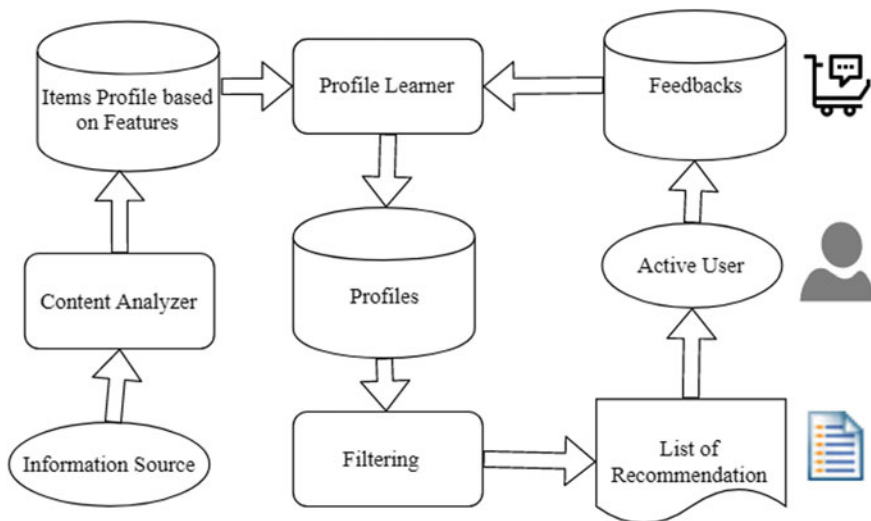


Fig. 3 Content-based filtering workflow

3.2 Collaborative Filtering

Collaborative filtering algorithm works by collecting and analysing a large amount of information on user behaviour, their preferences and their activities. Collaborative filtering is capable of recommending complex items more accurately because it does not reside on content analysed by machine. Such recommendation systems work on assumption that a user agreed in past will be interested in future as well and they are more probable to like similar kind of item.

Collaborative filtering techniques use distance feature to calculate similarities between user's choice and among items in database such as cosine distance, Pearson distance and Euclidean distance. We have implemented cosine and Pearson similarities to calculate similarity. Cosine similarity— this uses a coordinate space in which items are represented as a vector. It measures the angle between vectors and gives out their cosine values [5]. Pearson distance—it is a measure of linear correlation between two variables.

$$S_{\text{pearson}} = \frac{\sum_{i=1}^n (r_{u,i} - \hat{r}_u) \times (r_{v,i} - \hat{r}_v)}{\sqrt{\sum_{i=1}^n (r_{u,i} - \hat{r}_u)^2 \times (r_{v,i} - \hat{r}_v)^2}} \quad (1)$$

$$S_{\text{cosine}} = \frac{\sum_{i=1}^n r_{u,i} \cdot r_{v,i}}{\sqrt{\sum_{i=1}^n (r_{u,i})^2 \times (r_{v,i})^2}} \quad (2)$$

The idea is to create a community that shares a common interest [15]. Such users form a neighbourhood. And thus, a user gets a recommendation for items that he/she



Fig. 4 Framework of collaborative filtering

have not rated before but rated positively by users in his/her community. Collaborative filtering is of following types.

Memory-based approach: they are also called as neighbourhood-based collaborative filtering algorithms, in which the ratings of user–item combinations are predicted on the basis of their neighbourhoods which include user–user-based collaborative filtering and item–item-based collaborative filtering [16–18].

User-based Collaborative Filtering: In this, the rating predictions are calculated based on similar minded users of the target user. To predict rating preference for user A , the idea is to find top k similar users of A and compute weighted average of ratings of peer group.

Item-based Collaborative Filtering: In this, item similarity is used to determine rating prediction for target user. The idea is to find a set of similar items for which prediction was sought and then use these items' rating to compute final prediction of user to item.

Model-based approach: In this, machine learning and data mining methods are used in the context of predictive models. For example: decision tree, rule-based model, Bayesian model and latent factor model. In this paper, we have implemented latent factor model, using SVD (singular value decomposition) (Fig. 4).

4 Experimental Set-up and Results

This paper illustrates the implementation of two basic recommender systems (memory-based and model-based) and compares their performance on the basis of various evaluation parameters.

4.1 Data set

This paper has used the following data sets to implement recommendation algorithms.

Movie lens: This data set describes 5-star rating and free-text tagging activity from movie lens, a movie recommendation service. It contains 100,004 ratings and 1296 tag applications across 9125 movies. These data were created by 671 users between 09 January 1995 and 16 October 2016. This data set was generated on 17 October 2016 [19].

Jester: Over 4.1 million continuous ratings (-10.00 to $+10.00$) of 100 jokes from 73,421 users were collected between April 1999 and May 2003 [20].

4.2 Working Process

4.2.1 Memory-Based Collaborative Filtering: User-Based Collaborative Filtering

User-based collaborative filtering is based on the assumption that similar users with similar preferences will rate their choices similarly. One has to find that similarity and predict missing ratings for that user. When missing, ratings are known, and we can also recommend user items as per his/her taste [21].

Item-based collaborative filtering—This looks into the sets of items that target user has rated and compute how similar they are to the target item i and then select more similar item k , and also compares their corresponding similarities. The prediction is then computed by taking a weighted average of the target user's ratings on these similar items [22].

$$P_{u,j} = \hat{r}_u + K \sum_{i=1}^n S(u, i) \times (r_{i,j} - \hat{r}_i) \quad (3)$$

Model-based filtering: SVD—Singular value decomposition is a well-established technique for identifying latent semantic factors in information retrieval. Collaborative filtering uses SVD by factoring user item rating matrix.

Let the user item rating matrix is described as $R_{n \times m}$ with N number of users' rate M items, and R_{ij} describes the rating of item j given by user i . For a matrix R , its SVD is factorisation of R into three matrices such that:

$$R = P \Sigma Q^T \quad (4)$$

where Σ is the diagonal matrix whose values σ_i are the singular values of decomposition, and both P and Q are the orthogonal matrices, which means $P^T P = I_{n \times n}$ and $Q^T Q = I_{m \times m}$. Originally, matrix P is $n \times k$, Σ is $k \times k$, and Q is $m \times k$, where R is $n \times m$ and has rank k .

The SVD represents an expansion of the original rating matrix in a coordinate system where the covariance matrix is diagonal. Matrix P represents user latent values, and matrix Q gives the item latent feature for given rating matrix [23].

5 Results

This paper considered the following parameters for evaluation and for a comparison of different algorithms against a data set, which has been shown in Table 1 (Fig. 5).

5.1 RMSE

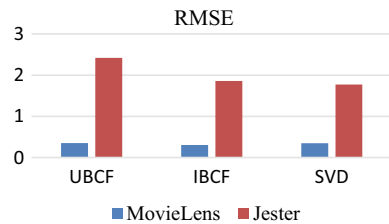
The RMSE (root-mean-square error) is computed by averaging the square of the differences between UV and the utility matrix, in those elements where the utility matrix is nonblank. The square root of this average is the RMSE [24].

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^n (P_i - R_i)^2} \quad (5)$$

Table 1 Results

Technique	Data set	RMSE	MAE	F-measure
UBCF	Movie lens	0.504	0.253	0.35179
	Jester	4.238	3.336	2.42
	Movie lens	0.483	0.206	0.3056
IBCF	Jester	4.241	3.335	1.860
	Movie lens	0.560	0.225	0.349
SVD	Jester	0.949	3.507	1.772

Fig. 5 Comparison of RMSE value of UBCF, IBCF and SVD on movie lens data set and Jester data set



5.2 MAE

The mean absolute error is an average of the absolute errors (Fig. 6).

$$MAE = \frac{1}{N} \sum_{i=1}^n |P_i - R_i| \tag{6}$$

5.3 F-Measure

Metric combines Precision and Recall into a single value for comparison pulse.

$$F\text{-Measure} = \frac{2 * \text{Precision} * \text{Recall}}{(\text{Precision} + \text{Recall})} \tag{7}$$

Precision is the measure of exactness. It determines the fraction of relevant items retrieved out of all items.

$$\text{Precision} = \frac{\text{True Positive}}{(\text{True Positive} + \text{False Positive})} \tag{8}$$

Recall is the measure of completeness. It determines the fraction of relevant items retrieved out of all items (Fig. 7).

$$\text{Recall} = \frac{\text{True Positive}}{(\text{True Positive} + \text{False Negative})} \tag{9}$$

Fig. 6 Comparison of MAE value of UBCF, IBCF and SVD on movie lens data set and Jester data set

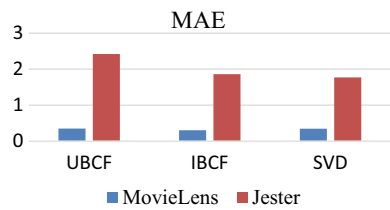
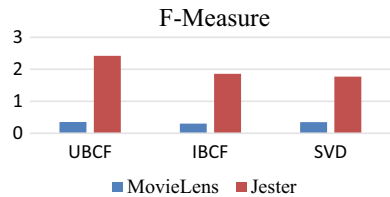


Fig. 7 Comparison of F-Measure value of UBCF, IBCF and SVD on movie lens data set and Jester data set



6 Conclusion and Future Scope

Recommendation system serves as a useful tool for users in expanding their interest and their experience over the Internet. Recommendation accelerates profits for developer and business person by knowing their customers well serving them best. Along with mobiles and computers, they open new security doors for the automobile industry and devices used on daily basis. Among several solutions and facilities, there are some issues related to available recommendation system that needs to be addressed specifically to take most out of them [25].

The recommendation can be made more complete and accurate by using latest data mining techniques and machine learning approach. Incorporating artificial intelligence into underlying algorithm strengthens the system to a greater extent as it helps in knowing the audience well and enough. Further improvisation is required so that recommendation system can do the intended job without compromising privacy and information leakage as mentioned above. All these factors imply that we are still in the urge to make promising systems, and there is way more to go for their development [26].

References

1. Aggarwal, C.C.: *Recommender Systems*. Springer International Publishing, Switzerland (2016). <https://doi.org/10.1007/978-3-319-29659-3>
2. Paul, R., Neophytos, I., Mitesh, S., Bergstrom, P., Riedl, J.: GroupLens: an open architecture for collaborative filtering of netnews. In: *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work* Chapel Hill, North Carolina, United States, pp. 175–186 (1994)
3. Witten, I.H., Frank, I.: *Data Mining*. Morgan Kaufman Publishers, San Francisco (2000)
4. Jhon Breese, S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, pp. 43–52, July (1998)
5. Deshpande, M., Karypis, G.: Item-based top-N recommendation algorithms. *ACM Trans. Inf. Syst.* **22**(1), 143–177 (2004)
6. Aamir, M., Bhusry, M.: Recommendation system: state of the art approach. *Int. J. Comput. Appl.* **120**(12), 25–32 (2015)
7. Fleder, D.M., Hosanagar, K.: Recommender systems and their impact on sales diversity. In: *ACM Conference on Electronic Commerce*, pp. 192–199 (2007)
8. Good, N., Schafer, J., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., Riedl, J.: Combining collaborative filtering with personal agents for better recommendations. In: *National Conference on Artificial Intelligence (AAAI/IAAI)*, pp. 439–446 (1999)
9. Claypool, M., Gokhale, A., Miranda, T., et al.: Combining content-based and collaborative filters in an online newspaper. In: *Proceedings of the SIGIR Workshop on Recommender Systems: algorithms and Evaluation*, Berkeley, Calif, USA (1999)
10. Jones, S.K.: A statistical interpretation of term specificity and its applications in retrieval. *J. Documentation* **28**(1), 11–21 (1972)
11. Gong, M., Xu, Z., Xu, L., Li, Y., Chen, L.: Recommending web service based on user relationships and preferences. In: *20th International Conference on Web Services*, IEEE (2013)
12. Rana, M.C.: Survey paper on recommendation system. *Int. J. Comput. Sci. Inf. Technol.* **3**(2), 3460–3462 (2012)

13. Resnick, P., Varian, H.R.: Recommender systems. *Commun. ACM* **40**(3), 56–58 (1997)
14. Balabanovi, M., Shoham, Y.: Fab: content based, collaborative recommendation. *Mag. Comm. ACM* **40**(3), 66–72 (1997)
15. Puntheeranurak, S., Chaiwitooanukool, T.: An item-based collaborative filtering method using item-based hybrid similarity. In: *Proceedings of the IEEE 2nd International Conference on Software Engineering and Service Science (ICSESS)*, pp. 469–472. ISBN: 978-1-4244-9699-0 (2011)
16. Miyahara, K., Pazzani, M.J.: Collaborative filtering with the simple Bayesian classifier. In: *Pacific Rim International Conference on Artificial Intelligence*, pp. 679–689 (2000)
17. Ghani, R., Fano, A.: Building recommender systems using a knowledge base of product semantics. In: *2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pp. 27–29 (2002)
18. Elgohary, A., Nomir, H., Sabek, I., Samir, M., Badawy, M., Yousri, N.A.: Wiki-rec: A semantic-based recommendation system using wikipedia as an ontology. *Intell. Syst. Des. Appl. (ISDA)* (2010)
19. <http://files.grouplens.org/datasets/movielens/ml-latest-small-README.html>
20. <http://eigentaste.berkeley.edu/dataset/>
21. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Analysis of recommendation algorithms for E-commerce. In: *Proceedings of 2nd ACM Conference on Electronic Commerce Minnesota USA*, pp. 158–167 (2000)
22. Karypis, G.: Evaluation of item-based top-N recommendation algorithms. In: *Proceedings of the International Conference on Information and Knowledge Management (CIKM '01)*, Atlanta, GA, USA, pp. 247–254 (2001)
23. Koren, Y.: Collaborative filtering with temporal dynamics. *Commun. ACM* **53**(4), 89–97 (2010)
24. <http://infolab.stanford.edu/~ullman/mmds/ch9.pdf>
25. Pronk, V., Verhaegh, W., Proidl, A., Tiemann, M.: Incorporating user control into recommender systems based on naive bayesian classification. In: *RecSys'07: Proceedings of the 2007 ACM Conference on Recommender Systems*, pp. 73–80 (2007)
26. Kanawati, R., Karoui, H.: A p 2p collaborative bibliography recommender system. In: *Proceedings Fourth International Conference on Internet and Web Applications and Services*, Washington DC, USA. *IEEE Comput. Soc.* 90–96 (2009)