

Recommender Systems: Issues, Challenges, and Research Opportunities

Shah Khusro, Zafar Ali and Irfan Ullah

Abstract A recommender system is an Information Retrieval technology that improves access and proactively recommends relevant items to users by considering the users' explicitly mentioned preferences and objective behaviors. A recommender system is one of the major techniques that handle information overload problem of Information Retrieval by suggesting users with appropriate and relevant items. Today, several recommender systems have been developed for different domains however, these are not precise enough to fulfil the information needs of users. Therefore, it is necessary to build high quality recommender systems. In designing such recommenders, designers face several issues and challenges that need proper attention. This paper investigates and reports the current trends, issues, challenges, and research opportunities in developing high-quality recommender systems. If properly followed, these issues and challenges will introduce new research avenues and the goal towards fine-tuned and high-quality recommender systems can be achieved.

Keywords Recommender system · Issues · Challenges · Content-based filtering · Collaborative filtering · Hybrid filtering

1 Introduction

The World Wide Web or simply the Web has brought numerous changes in the way we live and communicate with others. Today, we rely on the Web and its related technologies like web search engines that search and retrieve relevant information on almost any aspect of life. However, this change comes with the cost of information and cognitive overload on the user who is searching for relevant and reliable information on the Web. To cope with these issues, among other solutions,

S. Khusro(✉) · Z. Ali · I. Ullah

Department of Computer Science, University of Peshawar, Peshawar 25120, Pakistan
e-mail: {khusro,zafarali,cs.irfan}@upesh.edu.pk

© Springer Science+Business Media Singapore 2016

K.J. Kim and N. Joukov (eds.), *Information Science and Applications (ICISA) 2016*,
Lecture Notes in Electrical Engineering 376,

DOI: 10.1007/978-981-10-0557-2_112

1179

a recommender system works as a helper in finding relevant and related items by making relevant suggestions to the users. These systems navigate the Web to come up with relevant suggestions for users based either on their explicitly mentioned preferences or objective behaviors [1]. Recommender systems are used in various domains including products, videos, images, articles, news, and books. In spite of numerous research and development advances, interest in this area is still broader because of the growing user demands. It is therefore necessary to build high-quality recommender systems for providing fine-tuned recommendations to users in a wide range of daily-life applications. In this regard, researchers and industry practitioners need to come forward and work on some of the prominent issues and challenges in the area of recommender systems, which are being presented in this research paper. Therefore, the objectives of the paper can be summarized as: a) to present the general concepts and techniques associated with different types of recommender systems so that novice researchers can get an idea of how a recommender system approaches relevant items in relation to the item being currently presented. b) To identify and present some of the prominent issues and challenges in the design and development of a fine-tuned recommender system. c) To present solutions, techniques, and research guidelines that might help in coping with some of these issues and in designing a fine-tuned recommender system.

Rest of the paper is organized as Section 2 presents general concepts and techniques related to recommender systems, Section 3 presents issues and challenges that recommender systems face, Section 4 presents some solutions and techniques to cope with these issues, and finally, Section 5 concludes our discussion.

2 General Concepts and Techniques

Recommender system recommends and suggest items to user by considering their objective behaviors and preferences [1]. A general model of a typical recommendation process is shown in Figure 1 highlighting the vital role of users and items in recommendation process.

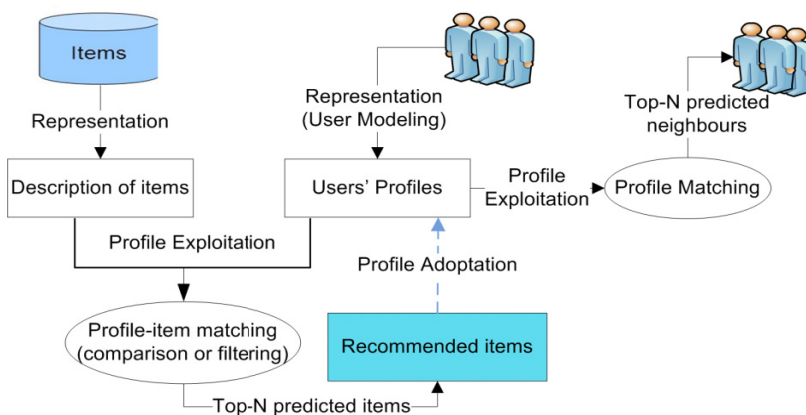


Fig. 1 General Model of Recommender Process

Formally we can define recommender system with S as the collection of all those items that may be recommended to users, U as the collection of all users for which the system makes recommendations, and F as the utility function that can be used to measure the usefulness of a particular item $s \in S$, to specific user $u \in U$. Mathematically it can be shown as $F: U \times S \rightarrow R$, where R shows the total recommended set. For every user $u \in U$, the system recommends such an item $s' \in S$ that it maximizes F for the user, which is generally denoted by rating provided by user. Usually item ratings are represented on a rating scale e.g., 1-5, in which the greater number say 5 represents that the user likes the item very much. For each user, a user profile is developed and updated containing information like his personal information, items visited, rated, purchased and downloaded. Similarly, each item has its own features and properties which are implicitly or explicitly updated as the users interact with the system. It is not possible that user provides ratings for all items in the catalogue, and therefore, some items are not rated at all. Therefore, a recommender system must use appropriate filtering methods so that such items are eventually rated either implicitly or explicitly and consequently recommended [2]. These filtering methods can be broadly divided into three categories, discussed in the coming sections:

2.1 Content-Based Filtering

A content-based recommender system suggests those items that are similar in features to items user has already liked in past [3]. A typical CB recommender first creates user profile using user feedback and ratings about items. The user profile is then compared with item features and the matched items are recommended. An example is LIBRA [4] that recommends books using book descriptions extracted from Amazon.com webpages. It learns user profile and recommends books using user ratings as well as features extracted from these webpages.

In CB approach, different profile-item matching techniques are used to match features of new items with user profile and to decide whether a particular item is interesting to user or not. User profile are either implicitly or explicitly updated, whereas utilities are assigned to items on the basis of utilities previously assigned to observed items [2]. Item profile is represented by the features and descriptions of these items using either Boolean values or integer values representing term frequency (TF), or term frequency-inverse document frequency (TF-IDF) [2]. Items are automatically classified into relevant and non-relevant by comparing item representations with representations of user interest and preferences using keyword matching, finding nearest neighbour approach, cosine similarity and typical classification.

2.2 Collaborative Filtering

Collaborative filtering recommends items by matching users with other users having similar interests [2]. It collects user feedback in the form of ratings provided

by user for specific item and finds match in rating behaviours among users in order to find group of users having similar preferences. Here, a user profile represents user preferences that the user has either explicitly or implicitly provided. An example is Amazon [5] uses CF approach, which suggests items based on the purchase patterns of its users as well as user ratings.

In a typical CF environment, there is a list of m users and n items represented by $\{u_1, u_2, u_3 \dots u_n\}$ and $\{i_1, i_2, i_3 \dots i_n\}$ respectively. Each user has a list of items that are rated either explicitly or implicitly. This way a user-item rating matrix ‘R’ is generated, where user preferences about items are represented. For finding missing ratings, different techniques are used including finding “nearest neighbor” for new users in recommending items to them by considering ratings provided by their nearest-neighbors.

2.3 Hybrid Filtering

In hybrid approach, two or more filtering methods are combined to gain better performance over CB and CF approaches when they are applied separately. Several researchers combined CB and CF techniques for gaining better results and mitigating various shortcomings faced by these approaches. Burke [6] has categorized hybridization methods into seven different types including: (1) weighted, (2) switching, (3) mixed approach, (4) feature combination, (5) cascade, (6) feature , and (7) meta-level hybridization approach. According to Burke, combining different filtering methods results in peak performance and alleviates problems faced by different filtering techniques when used separately.

3 Issues and Challenges

This section investigates issues and challenges in recommender systems and presents different solutions provided by researchers to handle these issues.

3.1 Cold Start Problem

This problem occurs when new users enter the system or new items are added to the catalogue. In such cases, neither the taste of the new users can be predicted nor can the new items be rated or purchased by the users leading to less accurate recommendations. The cold-start problem can be solved in many ways including: a) asking the user at the beginning to rate some items b) asking the users explicitly to state their taste in aggregate; d) and suggesting items to the user based on the collected demographic information [7]. User demographic information can be used to know about the location, zip-code along with interactions of the new user with the system in order to recommend items on the basis of ratings provided by other similar users having similar demographic information. In some domains, there may “sleepers” which are items that are good but are left unrated. Sleepers can be

handled using metadata or content-based methods [7], using item popularity, item entropy, and user personalization [3] as well as using Linked Open Data (LOD) by extracting data about items using LOD data sources without waiting for users to explicitly provide ratings [8].

3.2 *Synonymy*

Synonymy arises when an item is represented with two or more different names or entries having similar meanings [9]. In such cases, the recommender cannot identify whether the terms represent different items or the same item. For example, a memory-based CF approach will treat “comedy movie” and “comedy film” differently. The variation in using descriptive terms is greater than commonly thought and the excessive usage of synonym words decreases the performance of CF recommenders. Since item contents are thoroughly ignored, therefore, the recommender does not consider the latent association between items. This is the reason why new items are not recommended as long as these are rated by the users. To alleviate the problems of synonymy, different techniques including ontologies [10], the Single Value Decomposition (SVD) techniques [11], and Latent Semantic Indexing (LSI) could be used.

3.3 *Shilling Attacks*

What happens if a malicious user or competitor enters into a system and starts giving false ratings on some items either to increase the item popularity or to decrease its popularity [12]. Such attacks can break the trust on the recommender system as well as decrease the performance and quality of recommenders. This threat is of more concern in CF techniques but lesser threat to the item-based CF technique. There are different attack models like bandwagon, random, average, and reverse bandwagon attack. Attacks can be detected through different approaches like generic and model specific attributes, prediction shift, and hit ratio. These types of attacks can be categorized by dimensions like intent of attacking, size of attack, and the required knowledge to start the attack [13]. Other detection methods and metrics for shilling attacks can be found in [14].

3.4 *Privacy*

Feeding personal information to the recommender systems results in better recommendation services but may lead to issues of data privacy and security. Users are reluctant to feed data into recommender systems that suffer from data privacy issues. Therefore, a recommender system, whether CB or CF, should build trust among their users, however CF recommenders are more prone to such privacy issues [9]. In CF technique, user data including ratings are stored in a centralized repository which can be compromised resulting in data misuse. For this purpose, cryptographic mechanisms can be used by providing personalized

recommendations without involving third parties and peer users. Other techniques include using randomized perturbation techniques [15], allowing users to publish their private data without exposing their identities, and using Semantic Web technologies especially ontologies in combination with NLP techniques to mitigate the unwanted exposure of information [16].

3.5 Limited Content Analysis and Overspecialization

Content-based recommenders rely on content about items and users to be processed by information retrieval techniques. The limited availability of content leads to problems including overspecialization [3]. Here, items are represented by their subjective attributes, where selecting an item is based mostly on their subjective attributes. Features that represent user preferences in a better way, are not taken into account. For many domains, content is either scarce such as books or it is challenging to obtain and represent the content such as movies. In such cases relevant items cannot be recommended unless the analyzed content contains enough information to be used in distinguishing items liked/disliked by the user [17]. This also leads to representation of two different items with same set of features, where, e.g., well-written research articles can be difficult to distinguish from bad ones if both are represented with same set of keywords.

Limited content analysis leads to overspecialization in which CB recommenders recommend items that are closely related to user profile and do not suggest novel items. In order to recommend novel and serendipitous items along with familiar items, we need to introduce additional hacks and note of randomness [2], which can be achieved by using genetic algorithms that brings diversity to recommendations being made. The problem is relatively small in CF recommenders where unexpected and novel items may get recommended.

3.6 Grey Sheep

Grey sheep occurs in pure CF systems where opinions of a user do not match with any group and therefore, is unable to get benefit of recommendations [9]. Pure CB filtering can resolve this issue where items are suggested by exploiting user personal profile and contents of items being recommended. Similarly, sparse rating and first rater in CF filtering can also be resolved by CB filtering [2]. Integrating CB with CF techniques may also yield more serendipitous and novel suggestions [18]. Grey sheep users can be identified and separated from other users by applying offline clustering techniques including k-mean clustering. This way performance gets better and recommendation error is minimal [19].

3.7 Sparsity

The availability of huge size of data about items the catalogue and the disinclination of users to rate items make a dispersed profile matrix leading to less accurate

recommendations [9, 20]. The sparse rating in CF systems makes it difficult to make accurate predictions about items. CF uses nearest neighbours to recommend items, and less ratings make it computationally hard to calculate neighbours. The problem can be severe in context-aware recommenders as they use multidimensional vectors, where it becomes very difficult to provide accurate recommendations for very few rated items [21]. To cope with this situation, several approaches can be used including multidimensional recommendation model [22], SVD techniques [11], demographic filtering [3], and using content-boosted CF algorithm [23].

3.8 Scalability

The rate of growth of nearest-neighbour algorithms shows a linear relation with number of items and number of users. It becomes difficult for a typical recommender to process such large-scale data. For example, Amazon.com recommends more than 18 million items to more than 20 million customers. Different techniques have been proposed including clustering, reducing dimensionality, and Bayesian Network [20]. The problem can be addressed by using clustering CF algorithms that search users in small clusters instead of searching the entire database [9]; by reducing dimensionality through SVD [11], by pre-processing that combines clustering and content-analysis with CF algorithms [24], and by using item classification with weighted slope one scheme in determining vacant ratings in the sparse dataset of CF systems [25].

3.9 Latency Problem

CF recommenders face latency problem when new items are added more frequently to the database, where the recommender suggests only the already rated items as the newly added items are not yet rated. Using CB filtering can reduce waiting times but it may introduce overspecialization. To cope with this situation, the category based approach in combination with the user stereotype can be used [26]. To further increase performance, several clustering techniques [27] can be applied and calculations can be made offline. Model-based CF approaches can also enhance the scalability and performance.

3.10 Evaluation and the Availability of Online Datasets

Evaluating a recommender system determines among the other things, its quality. The design of evaluation criteria and selection of suitable evaluation metrics is a key problem in recommender systems. Most traditional recommender systems evaluate system results and algorithms by considering dataset divided into a test set and apply metrics like MAE, Precision, and F-Measure for evaluation [28, 29]. The available evaluation metrics are for general-purpose recommenders and are not applicable in different domains especially in evaluating context-aware

recommenders. To evaluate context-aware recommenders, contextual precision and contextual ROC can be used [30]. Other evaluation methods include questionnaires, interviews, and user studies. However, these approaches are costly and time-consuming.

Another issue related to RS, is the non-availability of benchmark dataset that can be used in evaluating a recommender system in a particular domain. Such datasets are important as their availability encourages research and development of recommenders. Most of the available datasets are proprietary and therefore, cannot be used as benchmark datasets especially in evaluating context-aware recommenders. Fortunately, some datasets have been released for evaluating context-aware recommenders that include Food, Movie and LDOS-CoMoDa1[30].

3.11 Context-Awareness

From an operational point of view, context-awareness aggregates all categories that represent the setting in which recommender is deployed, e.g., the current location, the current activity, and the time. It is envisioned that the upcoming recommender systems will use contextual information obtained through mobile services infrastructure and will include the user's short and long term history, location, entries in the calendar, and the information that the user provides to social networks. This can greatly affect the performance of recommenders [31]. Finding out user preferences and context-related information is the key to come up with relevant recommendations for the user. Moreover, the performance can be improved if user context-related information can be found out in an unnoticeable way. For obtaining this unobtrusive preference elicitation, three basic methods can be used including detecting facial expressions [32], recording speech interpretation, and physiological signals analysis [33].

4 Research Opportunities in Alleviating Issues and Challenges

By carefully analyzing the issues and challenges presented in Section 3, we present some design guidelines that can be helpful designing a fine-tuned recommenders that will perform better in mitigating issues like latency, cold-start, scalability, context-awareness, grey-sheep, and sparsity. These guidelines include:

1. Using demographic filtering and clustering, a recommender system may cluster users having similar preferences and demographic features that the system can only look into the appropriate user group rather than the entire dataset. This will minimize latency, increase performance, handle sparsity, and grey sheep problem.

¹ <http://212.235.187.145/spletnastran/raziskave/um/comoda/comoda.php>

2. The personal information of newly registered users can be obtained through registration. Contextual information such as location, time etc. can be obtained through their IP address and those items are recommended that have been mostly viewed, downloaded and purchased by other users having similar contextual information. This can easily avoid cold-start problem.
3. For a user with frequently changing preferences, two recommendations list should be maintained. The first one should be maintained according to the current preferences of the user, while the second one should keep track of user's long-term preferences so that the system should recommend items that match user previous transaction history.
4. The recommender system should filter out obsolete and older items. A time threshold should be used to find out such items, and consequently newer items should be presented to users along with accurate suggestions.

5 Conclusion

In the last few decades, recommender systems have been used, among the many available solutions, in order to mitigate information and cognitive overload problem by suggesting related and relevant items to the users. In this regards, numerous advances have been made to get a high-quality and fine-tuned recommender system. Nevertheless, designers face several prominent issues and challenges. Although, researchers have been working to cope with these issues and have devised solutions that somehow and up to some extent try to resolve these issues, however we need much to do in order to get to the desired goal. In this research article, we focused on these prominent issues and challenges, discussed what has been done to mitigate these issues, and what needs to be done in the form of different research opportunities and guidelines that can be followed in coping with at least problems like latency, sparsity, context-awareness, grey sheep and cold-start problem.

References

1. Pu, P., Chen, L., Hu, R.: Evaluating recommender systems from the user's perspective: survey of the state of the art. *User modeling and User-Adapted Interaction* **22**, 317–355 (2012)
2. Montaner, M., López, B., De La Rosa, J.L.: A taxonomy of recommender agents on the internet. *Artificial Intelligence Review* **19**, 285–330 (2003)
3. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* **17**, 734–749 (2005)
4. Mooney, R.J., Bennett, P.N., Roy, L.: Book recommending using text categorization with extracted information. In: *Proc. Recommender Systems Papers from 1998 Workshop*, Technical Report WS-98-08 (1998)

5. Linden, G., Smith, B., York, J.: Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing* **7**, 76–80 (2003)
6. Burke, R.: Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction* **12**, 331–370 (2002)
7. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: *The adaptive web*, pp. 291–324. Springer (2007)
8. Heitmann, B., Hayes, C.: Using linked data to build open, collaborative recommender systems. In: *AAAI Spring Symposium: Linked Data Meets Artificial Intelligence*, pp. 76–81 (2010)
9. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. *Advances in Artificial Intelligence* **2009**, 4 (2009)
10. Lau, A., Tsui, E., Lee, W.: An ontology-based similarity measurement for problem-based case reasoning. *Expert Systems with Applications* **36**, 6574–6579 (2009)
11. Billsus, D., Pazzani, M.J.: Learning collaborative information filters. In: *ICML*, pp. 46–54 (1998)
12. Gunes, I., Kaleli, C., Bilge, A., Polat, H.: Shilling attacks against recommender systems: a comprehensive survey. *Artificial Intelligence Review* **42**, 767–799 (2014)
13. Zhang, F.: A survey of shilling attacks in collaborative filtering recommender systems. In: *International Conference on Computational Intelligence and Software Engineering, CiSE 2009*, pp. 1–4 (2009)
14. Mobasher, B., Burke, R., Bhaumik, R., Sandvig, J.J.: Attacks and remedies in collaborative recommendation. *IEEE Intelligent Systems* **22**, 56–63 (2007)
15. Polat, H., Du, W.: Privacy-preserving collaborative filtering using randomized perturbation techniques (2003)
16. Heupel, M., Fischer, L., Bourimi, M., Scerri, S.: Ontology-enabled access control and privacy recommendations. In: *Mining, Modeling, and Recommending ‘Things’ in Social Media*, pp. 35–54. Springer (2015)
17. Lops, P., De Gemmis, M., Semeraro, G.: Content-based recommender systems: State of the art and trends. In: *Recommender systems handbook*, pp. 73–105. Springer (2011)
18. Avazpour, I., Pitakrat, T., Grunske, L., Grundy, J.: Dimensions and metrics for evaluating recommendation systems. In: *Recommendation Systems in Software Engineering*, pp. 245–273. Springer (2014)
19. Ghazanfar, M.A., Prügel-Bennett, A.: Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems. *Expert Systems with Applications* **41**, 3261–3275 (2014)
20. Shahabi, C., Chen, Y.-S.: Web information personalization: challenges and approaches. In: *Databases in Networked Information Systems*, pp. 5–15. Springer (2003)
21. Zhang, Y., Wang, L.: Some challenges for context-aware recommender systems. In: *2010 5th International Conference on Computer Science and Education (ICCSE)*, pp. 362–365 (2010)
22. Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS)* **23**, 103–145 (2005)
23. Melville, P., Mooney, R.J., Nagarajan, R.: Content-boosted collaborative filtering for improved recommendations. In: *AAAI/IAAI*, pp. 187–192 (2002)
24. Shahabi, C., Banaei-Kashani, F., Chen, Y.-S., McLeod, D.: Yoda: an accurate and scalable web-based recommendation system. In: *Cooperative Information Systems*, pp. 418–432 (2001)

25. Kumar, A., Sharma, A.: Alleviating sparsity and scalability issues in collaborative filtering based recommender systems. In: Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA), pp. 103–112 (2013)
26. Sollenborn, M., Funk, P.: Category-based filtering and user stereotype cases to reduce the latency problem in recommender systems. In: Advances in Case-Based Reasoning, pp. 395–405. Springer (2002)
27. Mobasher, B., Dai, H., Luo, T., Nakagawa, M.: Improving the effectiveness of collaborative filtering on anonymous web usage data. In: Proceedings of the IJCAI 2001 Workshop on Intelligent Techniques for Web Personalization (ITWP 2001), September 2010
28. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* **22**, 5–53 (2004)
29. Schröder, G., Thiele, M., Lehner, W.: Setting goals and choosing metrics for recommender system evaluations. In: UCERSTI2 Workshop at the 5th ACM Conference on Recommender Systems, Chicago, USA (2011)
30. Zheng, Y., Burke, R., Mobasher, B.: Splitting approaches for context-aware recommendation: an empirical study. In: Proceedings of the 29th Annual ACM Symposium on Applied Computing, pp. 274–279 (2014)
31. Felfernig, A., Jeran, M., Ninaus, G., Reinfrank, F., Reiterer, S.: Toward the next generation of recommender systems: applications and research challenges. In: Multimedia services in intelligent environments, pp. 81–98. Springer (2013)
32. Xu, S., Jiang, H., Lau, F.: Personalized online document, image and video recommendation via commodity eye-tracking. In: Proceedings of the 2008 ACM Conference on Recommender Systems, pp. 83–90 (2008)
33. Janssen, J.H., van den Broek, E.L., Westerink, J.H.: Tune in to your emotions: a robust personalized affective music player. *User Modeling and User-Adapted Interaction* **22**, 255–279 (2012)