# Recommender System

## An Overview

Khushali Dave
MS in Computer Science
Stevens Institute of Technology
Hoboken, NJ 07030, USA
kdave2@stevens.edu

ABSTRACT

Today abundance of data is available but nobody has time for analyzing that data but still we need the best content available from that data for our daily use. Recommender Systems solves this issue by looking into these large volumes of data and searching for the most relevant personalized information according to users interest. In this paper, we first introduce Recommender systems and types of data available categorizing into implicit , explicit and hybrid. We then introduce types of Recommender Systems like Content Based Filtering, Collaborative filtering & Hybrid Filtering with suitable examples of each category, and analyzing the performance of the system with the ability to face the challenges. We conclude with learning evaluation metrics for measuring the performance of the Recommender system

## 1. INTRODUCTION

The exponential growth in the amount of information and the number of users using that information have created a potential challenge in timely access to the items of interest. Hence we go for obtaining recommendations for items we haven't tried from other people who have some experience. Recommender System assist this natural social process for providing suggestions for items to be of use to a User [1]. The suggestions relate to various decision making processes, such as what items to buy, what music to listen, or what online news to read.

These days providing just one product to users is simply not adequate. Companies should be able to provide *multiple* products that meet *multiple* needs of *multiple* customers. Therefore, companies has to provide customers with more options. However, in providing this new level of customization, businesses increase the amount of information that customers must process before they are able to select which items meet their needs. One solution to this information overload problem is the use of Recommender systems[2].

Recommender systems enhance the sales of E-Commerce sites by helping users find items they intend to purchase. It also improves cross sell by suggesting products for the user based on items in the shopping cart. Recommender systems helps to gain customer loyalty by creating valuable relationship between customer and site.Business invest in learning about their users, this learning is used by recommender system and customized user interface is provided to match customer needs. Customers repay these sites by returning to the one that best match their needs.

In a recommendation system application we have two entities users and items. Users $u_i$ have preference for certain items, and these preferences must be teased out of the data by collecting rating of all the items $I_u$ given by that user. The ratings can either be on a 1-5 scale or purchase or click throughs [3]. For example, we can see in Table1 user-item matrix representing users rating of movies on a 1-5 scale, with 5 the highest rating. We can see some blanks which represent the situation where the user has not rated the movie. The movie names are HP1, HP2, and HP3 for Harry Potter I, II, and III, TW for Twilight , and SW1, SW2, and SW3 for Star Wars episodes 1, 2, and 3. The users are represented by capital letters A through D [4].

***Table 1:*** *A user-item matrix representing ratings of movie*

| | HP1 | HP2 | HP3 | TW | SW1 | SW2 | SW3 |
|---|---|---|---|---|---|---|---|
| *A* | 4 | | | 5 | 1 | | |
| *B* | 5 | 5 | 4 | | | | |
| *C* | | | | 2 | 4 | 5 | |
| *D* | | 3 | | | | | 3 |

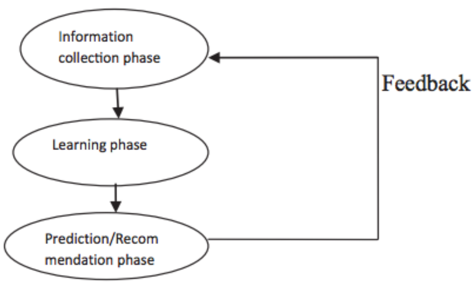
The aim of recommendation system is to predict the unrated items in the user-item matrix. For example : Consider A as the active user we would like to predict whether A will like SW2 or not.

Recently, various approaches have been developed for building Recommendation System, which are made utilizing Collaborative Filtering technique or Content Based Filtering technique or Hybrid Filtering technique [5-7]. Collaborative Filtering recommends items to a user by grouping other users with similar interest; it uses their past ratings to recommend items to the active user.On the other hand, Content-based filtering makes recommendations based on user choices made in the past and also by matching content resources to user characteristics (e.g. in a web-based e-commerce RS, if the user purchased some fiction films in the past, the RS will probably recommend a recent fiction film that he has not yet purchased on this website). Hybrid Filtering uses both the content based Filtering as well as Collaborative Filtering to take advantage of both and also to overcome the disadvantages of both. NewsWeeder is a net news-filtering system that maintains user profile by letting the user rate his or her interest level for each article being read (1-5), and then learning a user profile based on these ratings. News weeder uses both CF and CBF technique by looking at the article's text to determine its relevance and also by focusing on using ratings from early readers of an article to predict later reader's rating.Fab depends mostly on the ratings from various users in order to make dataset and it is an example of content-based recommender system.Other applications who use content-based filtering to help users find information include Letizia [9]. The system makes use of a web page that assists users in browsing the Internet; it is able to track the browsing pattern of a user to predict the pages that they might be interested in.Ringo is an online social information filtering system that uses collaborative filtering to build users profile based on their ratings on music albums [10].

As different applications have different needs we survey a range of properties which are commonly considered while selecting recommendation approach. Some applications may provide recommendation with high quality, but only for small amount of data but as data increases so does its quality decrease, such property is called scalability, We cover a range of such properties like Cold Start, Coverage, Privacy, Scalability etc. in Section[2].

To build a recommendation system there are phases of recommendation process which can be referred in Figure 1. First phase is information collection phase in which we collect relevant information of users as well as items based on which prediction will be made is covered in Section[3].After collecting data we apply learning algorithm which we refer as recommendation filtering techniques to filter and exploits users features is covered in Section[4]. The last phase is prediction/ recommendation phase after applying algorithm. The quality of recommendation algorithm is evaluated by using different measurements for example accuracy for recommendation which is covered in Section[5].

*Figure 1 :* Recommendation Phases




## 2. Recommendation Properties / Challenges

In this section we survey various properties which are commonly considered when deciding recommendation approach to select. As different applications have different demands, the designer of the system must pick on the important properties to measure for the concrete application at hand. Some of the properties can be traded-off, the most obvious example perhaps is the decline in accuracy when other properties (e.g. diversity) are improved. It is important to understand and evaluate these trade-offs and their effect on the overall performance [11].

### *2.1. Cold Start*

The cold start problem happens when it is not possible to make reliable recommendations due to lack of initial ratings. We can divide this problem into two categories: new item, new user.

The new item problem arises when new item is added to the system and it is not recommended as initial ratings are not present for that item.The new user problem is the most common and important problem faced in recommender system. [13].Since new user is added in the system we don't have ratings on items, that user might like, hence they cannot receive personalized recommendations.

### *2.2. Data Sparsity*

Data sparsity problem arises as most users do not rate most items and hence the user rating matrix is typically very sparse. As we are not able to get common items rated by various users for CF. This problems often occurs when a system has a very high item-to-user ratio, or the system is in the initial stages of use[12].

### *2.3. Scalability*

With the growth of the data set, many algorithms either slow down or require additional resources such as computation power or memory. Scalability is typically measured by experimenting with growing data sets, showing how the speed and resource consumption behave as the task scales up . For example if there are M users and N items if the algorithm runs in O(n) and M and N are very big then O(n) becomes too large and also on e-commerce sites recommendation has to be provided in real time for all users regardless of their purchase and ratings history[14].

### *2.4. Synonymy*

Synonymy refers to the tendency of similar items having different names. Recommender systems are mostly not able to identify such items as same. e.g. Work wears and Office wears. The pitfall of these method is that when some items are added who have different meaning from what is intended can lead to degradation of the performance of recommendation system.

### *2.5. Privacy*

Privacy has become import concern this days that most of the users pertain, that no third party should get their preferences to learn about a specific user. Studies [15] have found that complete privacy is not realistic and therefore we must compromise on minimizing the privacy breaches. Another solution is to set different levels of privacy to be accessible to Recommender system. Sometimes privacy comes at the expense of the accuracy of the recommendations.

### *2.6. Shilling Attack*

For recommendation we trust the data provided by the user to enhance recommendation for other people. But sometimes malicious users and competing vendors might decide to insert fake profiles into user-item matrices in such a way that they can affect the predicted ratings on behalf of their advantages.

## 3. Information Collection Phase

In this phase we collect relevant information for generating user profile , capture user attributes, behaviors or content of the resources which user accesses. It is very important to capture correct information about user to get the recommendation / prediction accurately. Recommender system depends on different types of input such as the most convenient high quality explicit feedback, which requires explicit input provided by users (e.g. rating) to describe their interest about the item or implicit feedback by observing user behavior and making conclusions through it (e.g. amount of time spend by user on a particular new article reading it or time spend by user watching video to suggest there likability).Hybrid feedback can also be used through the combination of both explicit and implicit.For Example: In any E-learning platform, a user profile is a collection of personal information linked to a specific user. It comprises of skills, intellectual capability, learning methods, domain interest, preferences and interaction with the system. The user profile helps to retrieve necessary information required to build a model. The success of any recommendation system depends largely on how accurately it is able to depict users interest.

### *3.1. Explicit Feedback*

Most of the E-commerce applications these days prompts user to provide ratings for items consumed in order to create/ improvise their model . The accuracy of the recommendation system depends on the quantity of ratings provided by the user. The only disadvantage of this method is additional effort required from user to provide ratings, as most of the users are not willing to do so. Even though explicit feedback has disadvantage of putting additional effort it is considered most reliable data , as it does not involve extracting inferences from actions as those inferences sometimes can be misleading. This method also brings transparency in the recommendation process by confirming with users to provide data. Therefor this method results in better recommendation quality and confidence in recommendations.

### *3.2.Implicit Feedback*

These days systems automatically infer users preferences by observing their behavior like the history of purchases, navigation history, time spent on some web page reading or watching a video, links traversed by the user, content of an email, button clicks and many more. Implicit feedback reduces a burden on users by inferring their user's preferences from their behavior with the system, but it is less accurate as it can be misleading too. It also argued that implicit data is more objective as there is no bias involved for a user to behave in a socially acceptable way and there are no self-image issues.

### *3.3. Hybrid Feedback*

To minimize the weakness of both we can use Implicit and explicit feedback combined in a hybrid system in order to achieve best performance. This approach can be utilized by using implicit feedback as validation for explicit feedback.

## 4. Recommendation Filtering Techniques

In this phase we apply filtering technique on the data obtained in information collection phase to generate accurate recommendation.We learn various techniques for achieving that. ***Figure 2*** represents types of recommender system

### *4.1. Content Based Filtering*

Content based filtering technique algorithm is focused on domain and requires analysis of attribute of users as well as items to generate prediction. CBF is considered most successful technique when documents such as web-pages, publications, books & news are recommended. In CBF technique, recommendation is made by extracting features from the items being rated by that specific user in past by matching features of user. Items with most positive rating is suggested to the user. For instance , considering movie genre information and knowing that user liked "the Notebook" and "Fault in our stars" , we can conclude inclination towards "Romantic Chic Flick" movies and hence suggest " The Vow". In various approaches user's preferences are treated as query, and unrated documents are scored based on similarity to this query [16].

CBF can be implemented using different types of algorithms to find similarity between documents to generate meaningful recommendation. Types of algorithms are Term Frequency Inverse Document frequency (TF/IDF) or Decision Trees or multinomial naive Bayes Classifier or Neural Networks or k-nearest neighbor. These approach makes prediction without using profile of other users as they do not influence recommendation.The major disadvantage of this technique is to have in-depth knowledge about domain.

In NewsWeeder [8], each category news document are converted into TF/IDF word vectors and then averaged to get a prototype vector of each category. In Mooney et al. [17], which is book recommendation site, it uses title, reviews, author, synopses and subject terms to train a multinomial naive Bayes Classifier.

### *4.2.Collaborative Filtering*

Collaborative Filtering is a recommendation technique which doesn't rely on domain for content. This approach is applied on data where we cannot describe data by metadata like movies and music.Collaborative filtering technique works by building database of preferences (user-item-matrix) for items by users. It then forms group of users with similar interests by calculating similarities between items or users to make recommendation. This group is called neighborhood [19]. User gets prediction / recommendation about items they have not rated before based on neighborhood. Recommendation produced by CF technique can provide either a numeric value which is called prediction e.g predicted rating a user u will give to item i or Top N items which that user will like the most which is called recommendation.The Collaborative filtering has two sub categories: memory based and model based.

#### *4.2.1.Memory based techniques*

In Memory based technique, a group of users is selected based on similarity with the active user. The items rated by users in past play very critical role in identifying group of people with similar interest.Once the neighborhood is

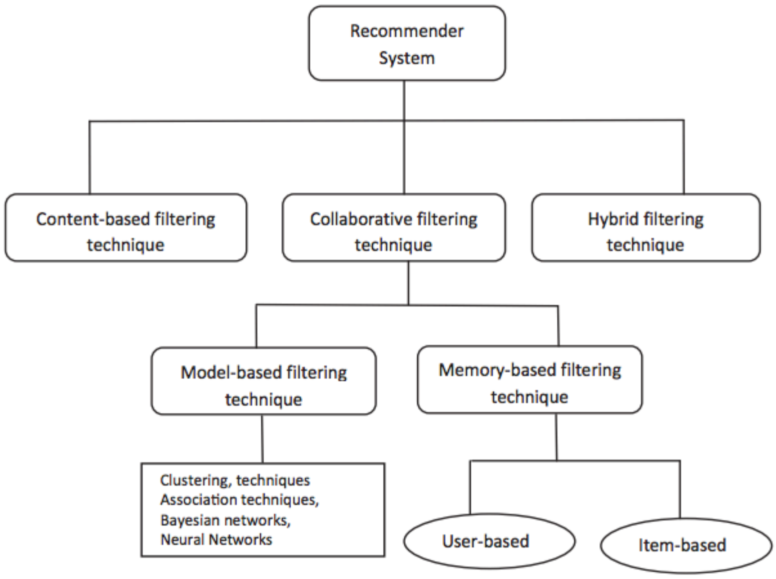



**Figure 2** Recommendation techniques.

identified several algorithms can be applied to combine preferences of neighbors to obtain recommendation.

Memory based CF can be achieved in two ways:

- User Based Collaborative Filtering
- Item Based Collaborative Filtering

User Based Collaborative Filtering technique calculates similarity between users by comparing their ratings on the same item and a weighted average of ratings of the group for that item produces predicted rating of an item, where weight refers to the similarity between users. UBCF builds a model of user similarities by assigning a weight to all users with regards to the similarity with active user, then select k most similar users to active user and prediction is made by taking weighted average of the selected neighbors rating[18].

Item based collaborative Filtering Technique computes predictions using the similarity between items and not the similarity between users. It builds a model of item similarities by retrieving all items rated by an active user from the user-item matrix, it determines how similar the retrieved items are to the target item, then it selects the k most similar items and their corresponding similarities are also determined.Prediction is made by taking a weighted average of the active users rating on the similar item k.

There are many similarity measures but the most popular ones are Pearson correlation similarity and Cosine similarity. Pearson correlation is used to measure the extent to which two variables linearly relate with each other and is defined as [20]:

$$w_{a,u} = \frac{\sum_{i\in I}(r_{a,i}-\bar{r}_a)(r_{u,i}-\bar{r}_u)}{\sqrt{\sum_{i\in I}(r_{a,i}-\bar{r}_a)^2\sum_{i\in I}(r_{u,i}-\bar{r}_u)^2}} \qquad (1)$$

From the above equation, $W_{a,u}$ denotes the similarity between two users a and u, $r_{a,i}$ is the rating given to item i by user a, $r_a$ is the mean rating given by user a while n is the total number of items in the user-item space. Also, prediction for an item is made from the weighted combination of the selected neighbors' ratings, which is computed as the weighted deviation from the neighbors' mean. The general prediction formula is as follows:

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u\in K}(r_{u,i}-\bar{r}_u)\times w_{a,u}}{\sum_{u\in K} w_{a,u}}$$

Cosine similarity is different from Pearson-based measure in that it is a vector-space model which is based on linear algebra rather that statistical approach. It measures the similarity between two n-dimensional vectors based on the angle between them. Cosine-based measure is widely used in the fields of information retrieval and texts mining to compare two text documents, in this case, documents are represented as vectors of terms. The similarity between two items u and v can be defined as follows:

$$w_{a,u} = \cos(\vec{r}_a, \vec{r}_u) = \frac{\vec{r}_a \cdot \vec{r}_u}{\|\vec{r}_a\|_2 \times \|\vec{r}_u\|_2} = \frac{\sum_{i=1}^{m} r_{a,i} r_{u,i}}{\sqrt{\sum_{i=1}^{m} r_{a,i}^2}\sqrt{\sum_{i=1}^{m} r_{u,i}^2}} \qquad (3)$$

*4.2.2.Model based techniques*

Model-based technique [21,22] uses information to create a model that generates the recommendations. Herein, we consider a method model-based if new information from any user outdates the model. Among the most widely used models we have Bayesian classifiers [23], neural networks [24], fuzzy systems [25], genetic algorithms [27,28], latent features [26] and matrix factorization [29], among others. To reduce the problems from high levels of sparsity in RS databases, certain studies have used dimensionality reduction techniques [30]. The reduction methods are based on Matrix Factorization

[31,29,32]. Matrix factorization is especially adequate for processing large RS databases and providing scalable approaches [33]. The model-based technique Latent Semantic Index (LSI) and the reduction method Singular Value Decomposition (SVD) are typically combined [34,35,48]. SVD methods provide good prediction results but are computationally very expensive; they can only be deployed in static off-line settings where the known preference information does not change with time. RS can use clustering techniques to improve the prediction quality and reduce the cold-start problem when applied to hybrid filtering. It is typical to form clusters of items in hybrid RS [36,37]. A different common approach uses clustering both for items and users (bi-clustering) [38,39]. RS comprising social information have been clustered to improve the following areas: tagging [208], explicit social links [179] and explicit trust information [181,70].

## 5. EVALUATION METRICS

The quality of the recommendation system is evaluated using different types of measurement such as accuracy or coverage. The type of metric used for evaluation depends on the type of filtering technique used. Accuracy means the ratio of correct recommendation to the total recommendation. Whereas coverage means the number of items in the search space the RS is able to provide recommendations for.For measuring the accuracy of recommendation system the metrics can be divided into predictive and decision support metrics[40].The suitability of metrics depends on the features and task recommender system wants to provide[41].

In predictive accuracy metric, we evaluate accuracy for recommendation filtering by comparing predictive ratings directly with actual user rating. Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Correlation are usually used as statistical accuracy metrics. MAE is the most popular and commonly used; it is a measure of deviation of recommendation from user's specific value. It is computed as follows[42]:

$$\mathrm{MAE} = \frac{1}{N}\sum\nolimits_{u,i}|p_{u,i} - r_{u,i}| \tag{4}$$

where $p_{u,i}$ is the predicted rating for user u on item i, $r_{u,i}$ the actual rating and N is the total number of ratings on the item set. The lower the MAE, the more accurately the recommendation engine predicts user ratings. Also, the Root Mean Square Error (RMSE) is given by Cotter et al. [43] as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{u,i}(p_{u,i} - r_{u,i})^2} \tag{5}$$

Root Mean Square Error (RMSE) puts more focus on bigger absolute error and the lower the RMSE is, the better the recommendation accuracy.

Decision support accuracy metrics which are popular are Reversal rate, Weighted errors, Receiver Operating Characteristics (ROC) and Precision Recall Curve (PRC), Precision, Recall and F-measure. These metrics help users in selecting items which are of very high quality. These metrics considers prediction procedure as a binary operation which differentiates good items from not so good items. ROC curves are very successful when performing comprehensive assessments of the performance of some specific algorithms. Precision is the ratio of recommended items to the total recommended items, while recall can be defined as the ratio of relevant items which are also a part of the set of recommended items . They are computed as

$$\text{Precision} = \frac{\text{Correctly recommended items}}{\text{Total recommended items}} \tag{6}$$

$$\text{Recall} = \frac{\text{Correctly recommended items}}{\text{Total useful recommended items}} \tag{7}$$

F-measure defined below helps to simplify precision and recall into a single metric. The resulting value makes comparison between algorithms and across data sets very simple and straightforward

$$F\text{-measure} = \frac{2PR}{P + R} \tag{8}$$

## 6. ADVANTAGES AND DISADVANTAGES

CBF filtering techniques overcomes challenge of CF. If user profile behavior updates , CBF has potential to adjust in a very short span of time. It works for First Rater/Cold Start problem to recommend new items if there are no ratings provided by user. So accuracy is not affected even if user's preferences are not there. CBF can handle data sparsity even if there are few common items by using items metadata as features to find similar items. However, the technique has certain disadvantages as well. Content based filtering technique is highly dependent on metadata. That is, they require rich description of items and very well organized user profile before recommendation can be made to users. This is called limited content analysis. So, the effectiveness of CBF depends on the availability of descriptive data. Content overspecialization is another serious problem of CBF technique. Users are restricted to getting recommendations similar to items they have ordered in past.

Collaborative Filtering has some major advantages over CBF in that it can perform in domains where there is not much content associated with items and where content is difficult for a computer system to analyze (such as opinions and ideal). Also, CF technique has the ability to provide serendipitous recommendations, which means that it can recommend items that are relevant to the user even without the content being in the user's profile. Despite the success of CF techniques, their widespread use has revealed some potential problem such as cold start, data sparsity, scalability and synonymy.

## 7. REFERENCECS

1. P. Resnick and H. R. Varian, "Recommender systems," Communications of the ACM, vol. 40, no. 3, pp. 56–58, 1997

2. J. Ben Schafer, Joseph Konstan, John Riedi , "Recommender systems in E-commerce"
3. B. N. Miller, J. A. Konstan, and J. Riedl, "PocketLens: toward a personal recommender system," ACM Transactions on Information Systems, vol. 22, no. 3, pp. 437–476, 2004.
4. Anand Rajaraman , Jeffrey David Ullman, "Mining of Massive Datasets", Cambridge University Press, New York, NY, 2011
5. Acilar AM, Arslan A, "A collaborative filtering method based on Artificial Immune Network.", Exp Syst Appl 2009;36(4):8324–32.
6. Chen LS, Hsu FH, Chen MC, Hsu YC, "Developing recommender systems with the consideration of product profitability for sellers", Int J Inform Sci 2008;178(4):1032–48.
7. Jalali M, Mustapha N, Sulaiman M, Mamay A. WEBPUM: a web-based recommendation system to predict user future movement. Exp Syst Applicat 2010;37(9):6201–12.
8. K. Lang, NewsWeeder: learning to filter netnews, in: Proceedings 12th International Conference on Machine Learning, 1995, pp. 331–339.
9. Lieberman H. Letizia: an agent that assists web browsing. In: Proceedings of the 1995 international joint conference on artificial intelligence. Montreal, Canada; 1995. p. 924–9
10. Chen LS, Hsu FH, Chen MC, Hsu YC. Developing recommender systems with the consideration of product profitability for sellers. Int J Inform Sci 2008;178(4):1032–48.
11. C. Sammut, G.I. Webb (eds.), Encyclopedia of Machine Learning and Data Mining, DOI 10.1007/978-1-4899-7687-1
12. P Melville, V Sindhwani -Recommender systems, 2011-Springer
13. J. Bobadilla,F. Ortega,A. Hernando,A. Gutiérrez,"Recommender systems survey",Elsevier,July 2013
14. Xiaoyuan Su and Taghi M. Khoshgoftaar,"A Survey of Collaborative Filtering Techniques", Hindawi Publishing Corporation,Volume 2009, Article ID 421425,19 pages.
15. Frankowski, D., Cosley, D., Sen, S., Terveen, L., Riedl, J.: You are what you say: privacy risks of public mentions. In: SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 565–572. ACM, New York, NY, USA (2006). DOI http://doi.acm.org/10.1145/1148170.1148267
16. Marko Balabanovi and Yoav Shoham, "Fab: Content-based , Collaborative recommendation", Communications of the Association for Computing Machinery, 40(3):66-72,1997
17. Raymond J. Mooney and Loriene Roy, "Content-based book recommending using learning for text categorization", In proceedings of the 5th ACM Conference on Digital Libraries, pages 195-204, San Antonio, TX, June 2000.
18. Melville, P., Sindhwani, V.: Recommender system, Encyclopedia of machine learning, pp. 1–9. (2010)
19. Herlocker JL, Konstan JA, Terveen LG, Riedl JT. Evaluating collaborative filtering recommender systems. ACM Trans Inform Syst 2004;22(1):5–53.
20. Resnick, P., Iacovou, N., Sushak, M., Bergstrom, P., & Reidl, J. (1994a). GroupLens: An open architecture for collaborative filtering of netnews. In Proceedings of the 1994 computer supported cooperative work conference, New York. New York: ACM. ACM Press New York, NY, USA
21. G. Adomavicius, A. Tuzhilin, 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 (6) (2005) 734–749
22. X. Su, T.M. Khoshgoftaar, A survey of collaborative filtering techniques, Advance in Artificial Intelligence 2009 (2009) 1–19.
23. S.B. Cho, J.H. Hong, M.H. Park, Location-based recommendation system using Bayesian user's preference model in mobile devices, Lecture Notes in Computer Science 4611 (2007) 1130–1139.
24. H. Ingoo, J.O. Kyong, H.R. Tae, The collaborative filtering recommendation based on SOM cluster-indexing CBR, Expert Systems with Applications 25 (2003) 413–423.
25. ]R.R. Yager, Fuzzy logic methods in recommender systems, Fuzzy Sets and Systems 136 (2) (2003) 133–149
26. J. Zhong, X. Li, Unified collaborative filtering model based on combination of latent features, Expert Systems with Applications 37 (2010) 5666–5672.
27. L.Q. Gao, C. Li, Hybrid personalizad recommended model based on genetic algorithm, in: International Conference on Wireless Communication, Networks and Mobile Computing, 2008, pp. 9215–9218
28. Y. Ho, S. Fong, Z. Yan, A hybrid ga-based collaborative filtering model for online recommenders, in: International Conference on e-Business, 2007, pp. 200–203
29. X. Luo, Y. Xia, Q. Zhu, Incremental collaborative filtering recommender based on regularizad matrix factorization, Knowledge-Based Systems 27 (2012) 271–280.
30. B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Application of dimensionality reduction in recommender system – a case study, in: ACM WebKDD Workshop, 2000b, pp. 264–272.
31. X.N. Lam, T. Vu, T.D. Le, A.D. Duong, Addressing cold-start problem in recommendation systems, in: Conference On Ubiquitous Information Management And Communication, 2008, pp. 208–211
32. X. Luo, Y. Xia, Q. Zhu, Applying the learning rate adaptation to the matrix factorization based collaborative filtering, Knowledge Based Systems 37 (2013) 154–164.
33. G. Takács, I. Pilászy, B. Németh, D. Tikk, Scalable collaborative filtering approaches for large recommender systems, Journal of Machine Learning Research 10 (2009) 623–656
34. M.G. Vozalis, K.G. Margaritis, Using SVD and demographic data for the enhancement of generalized collaborative filtering, Information Sciences 177 (2007) 3017–3037
35. S. Zhang, W. Wang, J. Ford, F. Makedon, Using singular value decomposition approximation for collaborative filtering, in: IEEE International Conference on E-Commerce Technology, 2005, pp. 1–8
36. S.K. Shinde, U. Kulkami, Hybrid personalizad recommender system using centering–bunching based clustering algorithm, Expert Systems with Applications 39 (1) (2012) 1381–1387.
37. Z. Yao, Q. Zhang, Item-based clustering collaborative filtering algorithm under high dimensional sparse data, in: International Joint Confeence on Computational Sciences and Optimization, 2009, pp. 787–790.
38. R.L. Zhu, S.J. Gong, Analyzing of collaborative filtering using clustering technology, international colloquium on computing, in: ISECS International Colloquium on Computing, Communication, Control, and Management, 2009, pp. 57–59.
39. T. George, S. Meregu, A scalable collaborative filtering Framework base don co-clustering, in: IEEE International Conference on Data Mining (ICDM), 2005, pp. 625–628.

40. Sarwar B, Karypis G, Konstan J, Reidl J. Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web, ACM, Hong Kong, China; 2001. p. 295.
41. Friedman N, Geiger D, Goldszmidt M. Bayesian network classifiers. Mach Learn 1997;29(2–3):131–63.
42. Claypool M, Gokhale A, Miranda T, Murnikov P, Netes D, Sartin M. Combining content- based and collaborative filters in an online newspaper. In: Proceedings of ACM SIGIR workshop on recommender systems: algorithms and evaluation, Berkeley, California; 1999
43. Sarwar B, Karypis G, Konstan J, Reidl J. Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web, ACM, Hong Kong, China; 2001. p. 295.