The Recommender Systems Model for Smart Cities

Sandeep Tayal, Kapil Sharma

Abstract*: Recommender systems were introduced in the early 1990s. They did not get too much attention and were limited to a narrow domain implemented by only a few companies until the outburst of E-commerce. As online shopping became popular, the recommender system started becoming an integral part of an organization marketing strategy and since then they have completely evolved a lot. This give an opportunity to start with a recommendation System project by collecting information from news of users to provide a best recommendation. The cities become smatter so, this paper review different methods of implementing Recommender systems models for smart cities along with their drawbacks and possible improvements.

***Index Terms*: Content-Based System, Collaborative Filtering, Evaluation, Hybrid System, Recommender System, smart cities**

I. INTRODUCTION

Recommender Systems gained popularity since 2005 when people understood that the Internet meant much more than surfing and socialization. The growth of E-commerce accelerated and by the end of 2007 itself, its sales accounted for 3.4% of total sales. Customers started finding the world of online shopping vast and reliable and organizations saw this as an opportunity to boost their business by getting to know their customers better. User’s feedback collection and analysis were no longer limited to some companies now. Feedbacks marked the valued opinion of the user and could very well be used for generalizing users on demographic fronts. Thus, they not only helped organizations in better responsiveness to the same user in the future but also to the users with the same hidden interests. This led to the nationwide deployment of collaborative filter out. Content-supported and Knowledge-based testimonial near quickly also became popular for bestowal with available data differently.

The recommendations based on user record should be as personalized as possible so that users feel satisfied and connected. Moreover, their implementation should not have a major impact on responsiveness to the user. Consequently, new techniques are being developed to progress the presentation of the recommender system handling large datasets. Various datasets have been made publicly available by companies to find out the best possible techniques to optimize the process. Million Song Dataset by Musixmatch with lyrics, large movie rating dataset by Flixster and MovieLens, product rating dataset by Amazon being some of them. Every year there is a dedicated annual conference (ACM Recommender System Conference) held on the topic that values new research papers on the topic and sought to find more areas to deploy a recommender system.

As of now, recommender systems may be majorly categorized into four classes designed on the approach used for the method specifically the cooperative filtering system, content-based system, knowledge-based and therefore the hybrid technique system.

- Collaborative recommendation System: It provides a recommendation of products or facilities to users based on their likeness in demographics or interest with other people whose preferences and personalized data has been present from past interactions.
- Content-based recommendation System: It provides a recommendation of objects based on their specification to users. Objects having a similar specification to the ones earlier preferred by a user are recommended.
- Knowledge-based recommendation System: It provides recommendations based on clear knowledge near the item classification, and recommendation criteria i.e., which item should be advised in which context.
- Hybrid Method System: It renews content-based filtering & collaborative-filtering approximate to refer most pursuit recommendations to the user.

This paper has been divided into three sections for dealing with collaborative, content and hybrid systems. The present state of the systems and their applications has been carefully analyzed.

II. COLLABORATIVE FILTERING

Collaborative Filtering is an initial technique of recommendation and maximum used in E-commerce systems since its creation. It may be further separated into Memory-based Collaborative filtering and model-based collaborative filtering.

Memory-based Collaborative filtering is the idea that users have similarities and their interests can be predicted by deducing patterns from their earlier interactions and comparing those preferences with other users or inducing possible interests based on their demographic likeness with other users. K-nearest-neighbor prediction algorithm was the first algorithm to be deployed widely in the recommender system. It performs *predict* task i.e. estimate likeliness of user to buy a particular item by neighborhood selection and computation of weighted average.

Neighborhood selection to find nearest related users with same interests to the active user.

Revised Manuscript Received on July 5, 2019.

Sandeep Tayal, Maharaja Agrasen Institute of Technology and Delhi Technological University, Delhi, India,

Kapil Sharma, Delhi Technological University, Delhi, India.

.

It is performed by calculating a likeness measure among the other users and target user's previous rating for the set of common items commonly using Pearson's correlation coefficient [3,4] or a vector cosine likeness calculate [5,6] and choosing the k most like users to the active users.

The likeness among the target user and its neighbor is formalized by numerical weight.

Pearson coefficient equation:

$$w_{a,u} = \frac{\sum_{i\in I}(r_{a,i}-\overline{r_a})(r_{u,i}-\overline{r_u})}{\sqrt{\sum_{i\in I}(r_{a,i}-\overline{r_a})^2\sum_{i\in I}(r_{u,i}-\overline{r_u})^2}}$$

(Eq. 1[11])

Cosine likeness measure:

$$w_{a,u} = cos(\vec{r_u},\vec{r_a}) = \frac{\vec{r_a}\cdot\vec{r_u}}{\|\vec{r_a}\|_2 \times \|\vec{r_u}\|_2} = \frac{\sum_{i=1}^{m} r_{a,i}r_{u,i}}{\sqrt{\sum_{i=1}^{m} r^2{}_{a,i}}\sqrt{\sum_{i=1}^{m} r^2{}_{u,i}}}$$

(Eq.2[30])

where items set is, I valued through all users, ru, I is rating assumed to item i through user u, r_u is the mean rating taking by user u, $r_{a,i}$ is the rating assumed through active(target) user to item i and m is dimensional space used to represent vector equation.

Normalization and computation of weighted average of ratings by all the proximity of the target user for the specific item to predict user-item likeliness. This approach is then scaled to all items in the domain to begin recommendations to the target user thus accomplishing *recommend* task.

Prediction of likeliness of active user towards item i

$$p_{a,i} = \overline{r_a} + \frac{\sum_{u\in K}(r_{u,i} - \overline{r_u}) \times w_{a,u}}{\sum_{u\in K} w_{a,u}} \quad (\text{ Eq. } 3^{[31]})$$

where $w_{a,u}$ is the likeness among users a and u, and K is the proximity or set of most like users and $\overline{r_u}$ is the average rating given by usher er.

Several extensions to the Pearson coefficient and cosine likely approach suggested and widely calculated over the time e.g. the default voting [5], case amplification [5], weighted majority prediction [7] and imputation-boosted CF [8], Inverse User frequency etc. The default voting and significance weighting deal with improving accuracy in prediction if there are only a few numbers of co-rated objects. Default Voting is because it is safe to give an explicit rating to the items that haven't been rated to overcome the problem of sparse rating matrix. This change evidently improves the accuracy of likeness techniques by a large factor. On the other hand, Significance weighting gets rid of the problem by multiplying likeness weight through a Significance Weighting factor, that undervalues the correlations base on some common rated items during recommend task. Case amplification improves performance by giving more importance to neighbors with higher likeness by introducing the concept of amplification factor and transforming original likeness weight equations.

Error procedures such as mean squared error and mean absolute error accesses accuracy of the forecast ratings by comparing them with actual ratings. Precision (ratio of relevant and retrieved to retrieved) and recall (ratio of relevant and retrieved to relevant) matrices are other evaluation matrices that are widely used.

There are several other likenesses measures available in the literature, with Spearman rank correlation, Kendall's τ mean squared differences, correlation etc.[9,10] Spearman rank correlation is the next most used likeness function after Pearson. It modifies Pearson likeness correlation by advocating the use of item ranks rather than ratings. In Spearman rank correlation items are ranked according to their ratings with highest rated as lowest ranked. Items having a similar rating is given the mean rank of position and then similar computation to Pearson correlation is followed. Herlocker et al. [9] draw light on different variants of neighborhood formation, weighting, and normalization processes, along with their comparisons with other alternatives based on experimental data.

With the advent of big data, Collaborative Filtering approach has been expanded to the application of clustering algorithm on the user-item ratings. This has been done to ensure good results when datasets are very large and parallel processing can be effectively exploited by making use of advanced frameworks. The algorithm specifies clustering of datasets into partitions of a fixed or variable size such that similar users belong to the same partition.

All these techniques are based on user-user based correlation and despite many improvements in algorithm and strategy it still has got some cons. The high increase in complexity of user likeness computation when scaling to large dataset being the most prominent of the problems. One of the major upsurges for Collaborative filtering came up when item-item based collaborative filtering came into the picture. Linden et al. [12] proposed *this* approach where the user's valued items were matched to parallel items more exactly than matching the target user to corresponding users. Sarwar et al. [6] provided the first major contribution by specifying the algorithm that could be take in item-item built collaborative filtering. The strategy was to combine the cosine formula and Pearson coefficient to calculate the object likenesses. Several variants of item-based collaboration have also come up for optimization [13]. The results [14] display that the recommendation algorithms built on similarities of the object perform considerably better than on item-based in most cases. Model-based Collaborative filtering technique provides recommendations by building and analyzing statistical models for user ratings along several well-defined parameters. The earliest model-filtering technique systems compared prediction problem to classification problem casting items as characteristic vectors above users and existing ratings as labels to create a multidimensional structure to better relate users along these dimensions. Dimensionality reduction techniques were used to overcome situations of data scarcity.

To support recommendation systems along with the model-based collaboration many distinct models came up with the passing time. The techniques which were seen most commonly in this proposition was Latent factor and matrix factorization models [15].

Latent model was a good model as it takes in consideration that the likeness was induced parallel between the users and items depending on the lower- dimensional structure which was not visible in the data. And not like other models like neighborhood-based methods which supported enumerated notions between user's likeness, or between items likeness.

For example, a user not only on the basis on likeness to a group of songs give a rating to it but other suggested factors are also there like his taste of music. In an approach of choice ratings of the user with respect to some item, vectors are very useful. And these vectors have hidden dimensions which emerged with users and items therein matrix factorization techniques.

The second most common class of technique that was revealed by Lee and Seung[16] was Non-negative matrix factorization. It was given this name because non-negative constraints are also included. When a user gives a rating, it is seen as the manifestation of different roles. For example, the target user presence was sure by the community in different clusters. Therefore, each user gives ratings which are traced along all the various dimensions to get the sum of users rating to that item. To summarize, though Collaborative Filtering Technique has its problems and still a lot needs to be evolved but its advantages are more profound and better express the approach. An efficient recommender system can very well be used to recommend new information to the users citing his/her hidden interests. It can be used to recommend non-structured data such as art, music etc. These are some of the favorable features that make Collaborate Filtering techniques such as an important class of recommender systems.

III. CONTENT BASED RECOMMENDATION

Content-Based Recommendation System gives a recommendation of an object to the user built on the content information of object instead of correlating different items or users based on statistical comments or feedbacks as in Collaborative filtering approach. The recommendation to a particular user does not have anything to do with the interests of other users of the system. In this approach, user and object profiles are made and continued based on the analysis of these profiles' recommendations are suggested. In this way, these systems learn function for modeling each user's interests. In music recommendation systems, the common characteristics (tone, beats, lyrics etc.) of the songs existing in the profile of the target users are analyzed and then similar songs are recommended.

The advanced algorithms influence be meant higher than the classical data retrieval approaches. they will be adjusted additional swimmingly by reducing weights of corresponding terms or nodes. Naive Thomas Bayes is one such approach to inductive learning and makes use of theorem classifiers for its analysis. It generates a probabilistic model supported applied mathematics knowledge and performs cluster of the offered dataset. it's 2 oftentimes used formulations particularly the variable Bernoulli and also the multinomial model. each these models allocate common values within which text documents are assumed to be generated by associate degree primary generative model, specially a parameterized mixture model. the alternative vital approach makes use of call trees within which internal nodes, branches, and leaves are labeled for various classes. during this technique coaching knowledge is partitioned off into text documents, into subgroups until solely instances of one category are a gift in those subgroups. Partitioning majorly involves text classification supported some feature like presence or absence of a personal word or phrase within the known word. This creates associate degree organized structural read for a far better understanding of user and object profiles. this system is generally used for tiny trees with few tests to avoid a performance hit. alternative adept ways are relevancy technique cluster analysis, neural networks [20] etc.

Content-Based Recommendation System also faces the problem of increased complexity when scaled to large datasets. Optimal threshold algorithm overcomes this problem. The algorithm [21,22] is an improved form of adaptive filtering technique [23] which analyses existing data and likeness distribution of user profile to decide threshold value and files having likeness over this threshold are only familiar inform user profiles. This technique greatly growing the system correctness and efficiency. These systems like Collaborative filtering systems also face the problem of accuracy-speed trade-off. Effective human-machine interaction is one remedy to this problem. They also face linguistic problems as user profiles with words from different languages need to be handled. Mart´ınez et al. [24] offered a adaptable language method to improve user profile compatibility.

To summarize Content-Based Recommendation is a relatively new recommendation approach but has well-defined areas of application. Unlike Collaborative filtering technique, it could not face the cold-start problem and is more effective in case of sparse datasets. Moreover, it can even recommend new objects to the target user by exploiting likeness in user's profiles and object specification.

IV. HYBRID SYSTEM

Both Collaborative Filtering techniques and Content-Based Techniques having some cons. The former face problems like early rater and sparsity of datasets while subjective domain and expressive content description are some of the weaknesses of the latter. So most real systems combine these approaches to overcome the shortcomings of each and provide the best results. There are many methods employed in this process like weighted, switched, feature combination and cascade.

In the weighted technique, predictions by different recommendations approach available [25,26,27] in the system are combined linearly after independent computation to provide optimal suggestion [28,29]. In switching technique, many recommendation techniques are deployed in the system. Hybridization strategy is made sensitive to a confidence level of results given by one technique and accordingly decision is made to switch to other technique [30,32]. In feature combination technique, collaborative information is treated as basically extra information data related through each record and content-based method are used over these augmented data sets.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

In the cascade technique, hybridization is done in stages. Firstly, one method is working to produce coarse dataset speciation and then second method is employed optionally depending upon the differentiation level of previous results. Hybrid systems can also deploy any of these methods to merge different recommendation and data analysis techniques. There are many models based on this approach. Music recommendation model which uses collaborative filtering and the audio analysis technology [33] for recommending users and web recommender system combining collaborative filtering with the domain ontology are some examples [34].

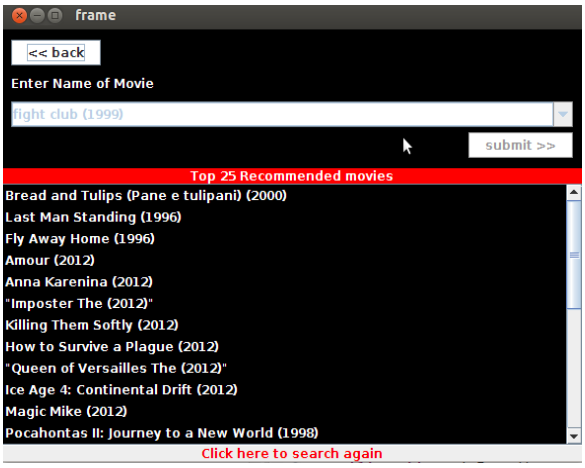
To summarize a hybrid recommender system are most reliable and fast as they are created for specific purposes based on the situation by combining the best possible choice of available approaches or techniques. They generally have specialized algorithms that are deployed in these systems for improving accuracy as well as speed.

V. IMPLEMENTATION

Different kinds of recommender systems are used extensively for specialized purposes. Collaborative filtering can be put to use in many applications. We are hereby discussing how it can be used to make a movie recommender system. We are using MovieLens dataset for this purpose which can be downloaded from their official website. The dataset comprises of many files of which we have used “movies.csv” and “ratings.csv” majorly to create a simple application that suggests similar movies to a user based upon the historic rating records. Recommending movies similar to a particular movie was done using Item-based Collaborative filtering. The likeness between movies was found out by using the Pearson Coefficient relation. Map Reduce programming technique was used to create the application. Movies were input from the user and then likeness of input movie with every other movie in the dataset was calculated and the top 25 results were filtered and displayed to the user. Internally the process performed the mapping from movie name to movie Id or vice versa. Four maps reduce tasks were performed by application for the purpose. These are:

<u>Mapper 1</u>

Scan the records of the dataset ratings.csv and divide it into key-value pairs.
Input- Record(uid,mid,rating,timestamp) where uid=unique Id of user, mid=Id of movie
Output- key(uid) value(mid,rating)
<u>Reducer 1:</u>
Input the sorted keys and list of corresponding values and output each key-value pair.
Input -key(used) value (List<value (mid, rating)>)
Output-key(uid) value (mid, rating)
<u>Mapper 2:</u>
Scan the records from output file of the first job and divide value into a pair of items each having desired target item (whose likeness needs to be found out) and every other item rated by the given user.
Input- Record(uid,List<mid,rating>)
Output- key(midn,targetmid) value(ratingn,targetrating) where midn=movie id of nth movie, targetmid=movie id of target movie ratingn and targetrating-rating given by user to nth movie and target movie respectively.
<u>Reducer 2:</u>
Input all the movie pairs and their corresponding ratings and output the Pearson correlation between these movies.
Input- key (midn, targetmid) List<value (ratingn, targetrating)>
Output-key (mind, targetmid) value (Pearson coefficient)
<u>Mapper 3:</u>
Scan the records from the output file of the first job and divide it into key-value pairs with keys being the Pearson correlation and output being an nth movie.
Input- Record (midn, targetmid, Pearson coefficient)
Output-key (Pearson coefficient) value(midn)
<u>Reducer 3:</u>
Input all the key-value pairs from preceding mapper sorted with respect to key and obtain the value of the last 25 pairs and prepare the corresponding list for reducer 4.
Input- key (Pearson coefficient) value(midn)
Output-key(midn) value (Pearson coefficient) of last 25 pair
<u>Mapper 4:</u>
Scan the records of the dataset movies.csv and divide it into key-value pairs.
Input- Record(id,name,genre) where id=id of movie,
name= name of the movie, genre=genre of movie
Output-key(id) value(name)
<u>Reducer 4:</u>
Input key value from Mapper 4 and a list of 25 items from mapper 3. Find the name of the corresponding movie with its presence in that list.
Input- key(id) value(name)
Output-key(index) value(name) for items with its present in list where index refers to index of movie id in the list.
This simple application can, therefore, be used by movie enthusiasts to know the movies that would like if they enjoy a movie. It may also be worked to predict movie rating by a target user. The output of the application for the popular movie “Fight Club” is shown below.



Enter Name of Movie

fight club (1999)

Top 25 Recommended movies

```
Bread and Tulips (Pane e tulipani) (2000)
Last Man Standing (1996)
Fly Away Home (1996)
Amour (2012)
Anna Karenina (2012)
"Imposter The (2012)"
Killing Them Softly (2012)
How to Survive a Plague (2012)
"Queen of Versailles The (2012)"
Ice Age 4: Continental Drift (2012)
Magic Mike (2012)
Pocahontas II: Journey to a New World (1998)
```

Click here to search again

Fig 1. Top 25 movies similar to Fight Club

In this way, this application can be used by movie enthusiasts to take smart decisions about whether to watch a movie or not.

VI. CONCLUSION AND FUTURE SCOPE

This paper analyzed different approaches of a recommendations system model for smart cities each having their own pros and cons. The best recommendation approach is evidently found out to be the use of Hybrid System that can be customized and scaled according to the problem. They can be programmed for the better functioning with respect to speed-accuracy tradeoff.

The domain of the recommender system is expanding at a fast pace with new possible optimizations getting suggested every day. The paper has been written with utmost care to mention all the presently deployed techniques regarding the subject but may need to be reviewed in the future to mention further advancements in this field.

REFERENCES

1. D. Goldberg, D. Nichols, B. Oki, and D. Terry.Using collaborative filtering to weave an information tapestry. *Communications of the Association of Computing Machinery*, 35(12):61–70, 1992.
2. F. J. Martin, “RecSys ’09 industrial keynote: Top 10 lessons learned to develop deploying and operating real-world recommender systems,” in ACM RecSys’09, pp. 1–2, ACM, 2009.
3. Resnick P, Iacovou N, Suchak M, Bergstrom P, and Riedl J, GroupLens: An open architecture for collaborative filtering of netnews. Proc. ACM Conf. on Computer-Supported Cooperative Work, 1994, Chapel Hill, pp. 175–186.
4. Shardanand U and Maes P, Social Information Filtering: Algorithms for Automating ‘Word of Mouth’. Proc. Conf. Human Factors in Computing Systems. Denver, 1995: 210–217.
5. Breese JS, Heckerman D, and Kadie C, Empirical Analysis of Predictive Algorithms for Collaborative Filtering. Proc. 14th Conf. Uncertainty in Artificial Intelligence. Madison, 1998:43–52.
6. Sarwar B, Karypis G, Konstan J, and Riedl J, Item-Based Collaborative Filtering Recommendation Algorithms. Proc. 10th Int’l WWW Conf., Hong Kong, 2001: 1–5.
7. Atsuyoshi Nakamura and Naoki Abe. Collaborative filtering using weighted majority prediction algorithms. In ICML ’98: Proceedings of the Fifteenth International Conference on Machine Learning, pages 395–403, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.
8. Xiaoyuan Su, Taghi M. Khoshgoftaar, Xingquan Zhu, and Russell Greiner. Imputation-boosted collaborative filtering using machine learning classifiers. In SAC ’08: Proceedings of the 2008 ACM symposium on Applied computing, pages 949–950, New York, NY, USA, 2008. ACM.
9. J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In Proceedings of 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 230–237, Berkeley, CA, 1999. ACM Press.
10. Xiaoyuan Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. Advances in Artificial Intelligence, 2009:1–20, 2009.
11. Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, 2005.
12. Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 7(1):76–80, 2003.
13. Badrul Sarwar, George Karypis, Joseph Konstan, and John Reidl. Item-based collaborative filtering recommendation algorithms. In WWW ’01: Proceedings of the 10th international conference on World Wide Web, pages 285–295, New York, NY, USA, 2001. ACM.
14. Deshpande M and Karypis G, Item-Based Top-N Recommendation Algorithms. ACM Trans. Information Systems, 2004, 22(1): 143–177.
15. Robert Bell Yehuda Koren and Chris Volinsky. Matrix factorization techniques for recommender systems. In IEEE Computer, volume 42 (8), pages 30–37, 2009.
16. D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative matrix factorization. In Nature, 401 (788), 1999.
17. Baeza-Yates R and Ribeiro-Neto B, Modern Information Retrieval. Addison-Wesley, Wesley Press, 1999.
18. Salton G, Automatic Text Processing. Addison-Wesley, 1989.
19. Rocchio, J.: Relevance Feedback Information Retrieval. In: G. Salton (ed.) The SMART retrieval system - experiments in automated document processing, pp. 313–323. Prentice-Hall, Englewood Cliffs, NJ (1971)
20. Mooney RJ, Bennett PN, and Roy L, Book Recommending Using Text Categorization with Extracted Information. Proc. Recommender Systems Papers from 1998 Workshop, Technical Report WS-98-08, 1998.
21. Robertson S and Walker S, Threshold Setting in Adaptive Filtering. J. Documentation, 2000, 56: 312–331.
22. Zhang Y and Callan J, Maximum Likelihood Estimation for Filtering Thresholds. Proc. 24th Ann. Intel ACM SIGIR Conf., New Orleans, 2001: 294–302.
23. Zhang Y, Callan J, and Minka T, Novelty and Redundancy Detection in Adaptive Filtering. Proc. 25th Ann. Intel ACM SIGIR Conf., Tampere, 2002: 81–88.
24. Mart´ınez L, P´erez LG, and Barranco M, A multigranular linguistic content-based recommendation model: Research Articles. International Journal of Intelligent Systems, 2007, 22(5): 419–434.
25. Claypool M, Gokhale A, Miranda T, Murnikov P, Notes D, and Sartin M, Combining Content-Based and Collaborative Filters in an Online Newspaper. Proc. ACM SIGIR ’99 Workshop Recommender Systems: Algorithms and Evaluation, Berkeley 1999.
26. Pazzani M, A Framework for Collaborative, Content-Based, and Demographic Filtering. Artificial Intelligence Rev., 1999, 13(5-6): 393–408.
27. Schein AI, Popescul A, Ungar LH, and Pennock DM, Methods and Metrics for Cold-Start Recommendations. Proc. 25th Ann. Intel ACM SIGIR Conf., Tampere, 2002: 253–260.
28. Billsus D and Pazzani M, User Modeling for Adaptive News Access. User Modeling and User-Adapted Interaction, 2000, 10(2-3): 147–180.
29. Getoor L and Sahami M, Using Probabilistic Relational Models for Collaborative Filtering. Proc. Workshop Web Usage Analysis and User Profiling, San Diego, 1999.
30. Basu C, Hirsh H, and Cohen W, Recommendation as Classification: Using Social and Content-Based Information in Recommendation. Papers from 1998 Workshop, Technical Report WS-98-08, AAAI Press 1998: 714–720.
31. Marko Balabanovic and Yoav Shoham. Fab: Content-based, collaborative recommendation. *Communications of the Association for Computing Machinery*, 40(3):66–72, 1997.
32. Christakou C, Vrettos S, and Stafylopatis A, A hybrid movie recommender system based on neural networks. International Journal on Artificial Intelligence Tools, 2007, 16(5): 771–792.
33. Yoshii K, Goto M, Komatani K, Ogata T, and Okuno HG, An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model. IEEE
34. Girardi R and Marinho LB, A domain model of Web recommender systems based on usage mining and collaborative filtering. Requirements Engineering, 2007, 12(1): 23–40.

AUTHORS PROFILE

Sandeep Tayal was born in Delhi, India in 1985. He is Pursuing a Doctorate Degree in Information Technology under the Department of Information Technology at Delhi Technological University (DTU), Delhi, India. He has obtained his Bachelor of Technology and Master of Technology degree in Information Technology and Computer Science and Engineering from Guru Gobind Singh Indraprastha University (GGSIPU), Delhi, India in 2008 and 2011 respectively. Presently, he is working as an Assistant Professor in the Department of Computer Science and Engineering at Maharaja Agrasen Institute of Technology, Rohini Sector- 22, Delhi, India. His research interest includes Big Data and Machine Learning.

Dr. Kapil Sharma was born in Haryana, India in 1977. He has completed his Doctors Degree in Computer Science and Engineering under the Faculty of Engineering and Technology at the M. D. University, Rohtak (Haryana), India in 2011. He has obtained his Bachelor of Engineering and Master of Technology Degrees in Computer Science & Engineering and Information Technology from M. D. University, Rohtak, and IASE, Rajasthan, India in 2000, and 2005 respectively. he was worked as a Faculty of the Department of Information Technology at Guru Premsukh Memorial College of Engineering, Budhpur, Delhi, India. Presently, he is working as Professor and Head of Information Technology Department at Delhi Technological University (DTU), Delhi, India. His research interests include Software Engineering, Reliability, Big Data and Machine Learning.