ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 5, Issue 5, May 2016

Hybrid Content-Based Filtering Recommendation Algorithm on Hadoop

Mohit G. Deshpande, Ashish V. Muddebihalkar, Ajay B. Jadhav, Pritish C. Patil,

Shrikant Kokate,

Department of Computer Engineering, Pimpri Chinchwad College OF Engineering

***Abstract*— In last two decades, because of revolution in internet technologies, the amount of information available online is increased exponentially. Recommender Systems (RS) are one of the innovations in this revolution. Collaborative Filtering (CF) is most commonly used techniques in developing a recommender system. This paper aims to develop a hybrid model based on CF which addresses the scalability of RS over a distributed environment and also the cold start problem associated with a RS. For testing the algorithm we have used Hadoop Framework, most popularly used distributed computing environment which supports batch processing.**

***Index Terms*— Hadoop, Hybrid recommendation algorithm, MapReduce**

I. INTRODUCTION

During the last two decades because of the revolution in internet technologies, the amount of information available online has increased exponentially. Recommender systems are of one the innovations in this revolution, which addresses the problem of filtering the data/information that is more likely to individual user's interest. In a typical recommender system, user profiles are maintained and used to predict ratings for items that have not been considered before [1].

Recommender systems (RS) are software tools and techniques which provide suggestions for items to be of use to user [2, 3, 4, and 5].

Here, an "item" is a general term which is used to denote, what the system recommends to users. Any Recommender System is designed to focus on a particular type of item and accordingly is the GUI, the Core recommendation technique used to generate recommendations, etc. are customized. In the following sub-section we discuss the types of RS.

Over last two decades, more than 200 research articles were published about recommender systems. In the previous works [5], four important classes of recommendation techniques, based on the information source as follows,

A. Collaborative: Collaborative systems generate recommendations based only upon the information of rating profiles of different users. Examples include [6, 7, 8, 9]

B. Content-based: Content based recommender systems provide recommendations based on two parameters, [10, 11, 12, 13]

- Features associated with the items.
- Rating that a user has given them. .

C. Demographic: A Demographic recommendation system provides recommendations based on the demographic profile of the users. And the recommended items could be from different demographic classes by combining the ratings of the users from those classes. [14, 15]

D. Knowledge-based: A Knowledge-based recommendation system is used to recommend the items based on inferences from the user's needs and preferences. This knowledge sometimes contains explicit functional knowledge about how certain features meet the user needs.

II. PROPOSED WORK

The proposed work gives design of a scalable feature-combination based hybrid recommendation system, based on map-reduce framework. Apache Hadoop [20] is an open-source framework for distributed computing that can be composed of large number of commodity hardware systems to run on an application or cluster.

Hadoop implements Google's Map-Reduce programming model Map Reduce [19]. We leverage these computing abilities of Hadoop to achieve scalability and more accuracy in our recommendations.

There exist several methods to combine collaborative filtering with context based techniques but probably not all of them will lead to the same prediction accuracy. A well-thought-out hybridization approach is critical for the success of the two-part recommender engine. Due to the fact that collaborative filtering is a well-established way of predicting preferences, and due to the fact as well that it is very well supported by Apache Mahout, we want to use this technique for our online module.

However in a collaborative-based recommender, items co-rated by a pair of users may be very few. Hence in this case

correlation between two users is not reliable. Content-based information of each user is exploited to detect similarities between them.

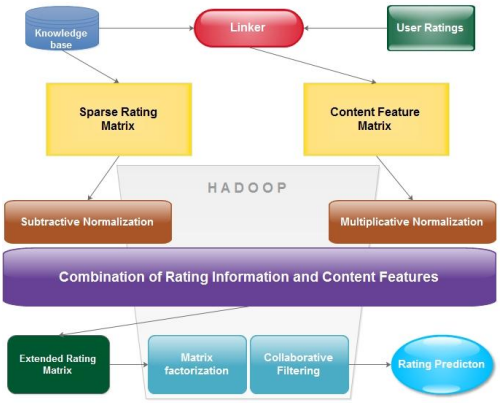


Figure 1: Architecture of Hybrid RS

The architecture diagram of our recommender system gives the basic idea of the operation of our system.

III. Collaborative and Content Based Filtering

COLLABORATIVE FILTERING (CF):

This technique perhaps the most studied and also the most widely-used recommendation approach in practice. Key characteristic of CF: It predicts the utility of items for a user based on the items previously rated by other like-minded users.

This approach includes both,

1. User-based methods
2. Item-based methods

ITEM-BASED METHOD: The item-based approach works by comparing items based on their pattern of ratings across users. The similarity of items i and j is computed as follows:

$$sim(i,j) = \frac{\sum_{\mathbf{u}\in U}(r_{\mathbf{u},i} - \bar{r}_{\mathbf{u}})(r_{\mathbf{u},j} - \bar{r}_{\mathbf{u}})}{\sqrt{\sum_{\mathbf{u}\in U}(r_{\mathbf{u},i} - \bar{r}_{\mathbf{u}})^2}\sqrt{\sum_{\mathbf{u}\in U}(r_{\mathbf{u},j} - \bar{r}_{\mathbf{u}})^2}}$$

This similarity measure is based on how much the ratings by common users for a pair of items deviate from average ratings for, also known as Pearson (correlation) based similarity.

After computing the similarity between items we select a set of k most similar items to the target item and generate a predicted value of user u's rating,

$$p(\mathbf{u},i) = \frac{\sum_{j\in J} r_{\mathbf{u},j} \times sim(i,j)}{\sum_{j\in J} sim(i,j)}$$

Where, J is the set of k similar items.

CONTENT-BASED FILTERING:

Perform item recommendations by predicting the utility of items for a particular user based on how "similar" the items are to those that he/she liked in the past. E.g. In a movie recommendation application, a movie may be represented by such features as specific actors, director, genre, subject matter, etc.

The user's interest or preference is also represented by the same set of features, called the user profile.

Recommendations are made by comparing the user profile with candidate items expressed in the same set of features. The top-k best matched or most similar items are recommended to the user. The simplest approach to content-based recommendation is to compute the similarity of the user profile with each item.

IV. Hybridization Technique Used

Content-based and collaborative methods have complementary strengths and weaknesses. Combine methods to obtain the best of both.

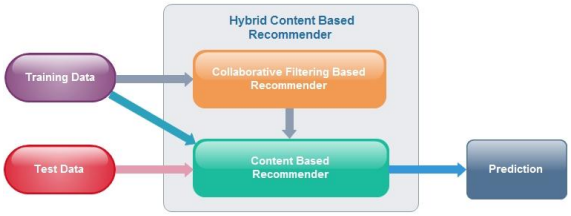


Figure 2: Overview of Hybridization Technique

Collaborative filtering uses the correlation between user ratings to make predictions. Such correlation is very meaningful when users have many rated items in common. In large datasets with many items this is not however always the case. Also, the lack of access to the content of the items prevents similar users from being matched unless they have rated the exact same item.

In collaboration through content, rated items as well as the content of the items are being used to construct profile of an user. This selection of terms which gives the description of content of the items is done using content-based techniques. The weights of terms indicate how important they are to the user.

Our approach consists of building a bi-dimensional model called user-item extended matrix. The model is the combination of a user-items features matrix and an item -users features matrix.

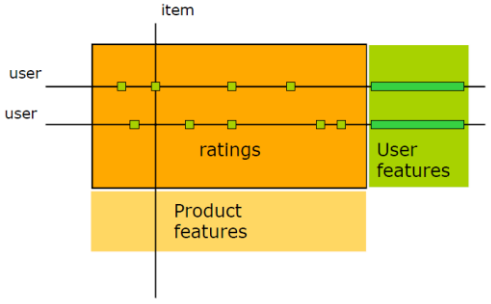


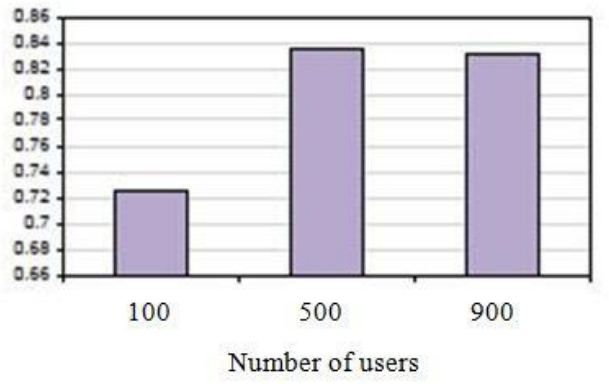
Figure 3: User-Item Matrix

All Rights Reserved © 2016 IJARCET

V. EVALUATION

In this section we are going to discuss briefly the evaluation process of the hybrid recommender engine that we have built. Our approach for such a recommender was to combine content features with ratings. We decided to use the rating data derived from Movie-Lens and adapt it to our context. Movie-Lens started out in 1997 and by now contains more than 10,000 movies rated by approximately 70,000 users. There exist several different Movie-Lens datasets that were made anonymous and published for research matters (i.e. 100K, 1M, 10M). We decided to use the 100K data set which contains anonymous ratings from 943 users given on 1682 movie items. Once the rating data was obtained we have assigned randomly user Ids and item ids from the dataset to the users and item we had created in our model. We also plotted the content features. We have considered for this thesis two features which were tag names and category names. For users we considered the tag names and category name they used to describe their profile. For the items we considered the tag names and category name used in their description.

Statistical accuracy metrics evaluate the accuracy of a system by comparing the numerical recommendation scores against the actual user ratings for the user-item pairs in the test dataset. Mean Absolute Error (MAE) between ratings and predictions is a widely used metric. MAE is a measure of the deviation values. Results are as follows,



VI. CONCLUSION

The approach presented in this paper for the implementation of the recommender system was a hybrid method that combines content based information and collaborative based information. The evaluation the Hybrid approach has proven that it performs better than single collaborative based recommenders and single content based recommenders. In that regards, we believe that our approach is practical for real life applications.

During the design process of the recommender engine, we opted for a semi- structured structure for our data. Even though for our work this structure in itself was enough, we believe that in order to perform better in real life applications, the data manipulated by the recommender engine needs to be described formally following ontology. This will be the future scope of this proposed algorithm.

ACKNOWLEDGMENT

We would like to thank our project guide Prof. Shrikant Kokate for his enormous co-operation and guidance.The technical guidance provided by him was more than useful and made the project successful. We would also like to thank our Department of Computer Engineering, Pimpri Chinchwad College of Engineering.

REFERENCES

1. GAUCH, SUSAN, MIRCO SPERETTA, ARAVIND CHANDRAMOULI and ALESSANDRO MICARELLI: User Profiles for Personalized Information Access. In BRUSILOVSKY, PETER, ALFRED KOBSA and WOLFGANG NEJDL (editors): The Adaptive Web, volume 4321 of Lecture Notes in Computer Science, pages 54–89. Springer, 2007.
2. Mahmood, T., Ricci, F.: Improving recommender systems with adaptive conversational strategies. In: C. Cattuto, G. Ruffo, F. Menczer (eds.) Hypertext, pp. 73–82. ACM (2009).
3. Resnick, P., Varian, H.R.: Recommender systems. Communications of the ACM 40(3), 56–58 (1997)
4. Burke, R.: Hybrid web recommender systems. In: The Adaptive Web, pp. 377–408. Springer Berlin /Heidelberg (2007)
5. Burke, R.: Hybrid Recommender Systems: Survey and Experiments. UMUAI 12 (4), 331-370. (2002)
6. Goldberg, D., Nichols, D., Oki, B, & Terry, D. Using collaborative filtering to weave an information tapestry. 35(12):61–70. CACM. (1992)
7. Hill, W., Stead, L., Rosenstein, M. and Furnas, G.: Recommending and evaluating choices in a virtual community of use. In: CHI '95: Conf. Proc. on Human Factors in Computing Sys., Denver, CO, 194-201. (1995)
8. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In: Proc. of the Conf. on Comp. Supp. Coop. Work, Chapel Hill, NC, 175-186. (1994)
9. Schwab, I. and Kobsa, A.: Adaptivity through Unobstrusive Learning. Künstliche Intelligenz 16(3): 5-9. (2002)
10. Chen, L. and Sycara, K.: WebMate: A personal agent for browsing and searching. In Proc.of the 2nd Intl Conf. on Autonomous Agents (Agents'98), 132-139, New York. ACM Press. (1998)
11. Jennings, A. and Higuchi, H.: A User Model Neural Network for a Personal News Service. UMUAI 3, 1-25. (1993)
12. Lang, K.: Newsweeder: Learning to filter news. In: Proc. of the 12th Intl Conf. on Machine Learning, Lake Tahoe, CA, 331-339. (1995)
13. Pazzani, M., Muramatsu, J., and Billsus, D.: Syskill & Webert: Identifying Interesting Web Sites. In Proc. of the 13th Natl Conf. on AI, 54-61. (1996)
14. Krulwich, B.: Lifestyle Finder: Intelligent User Profiling Using Large-Scale Demographic Data. AI Magazine 18 (2), 37-45. (1997)
15. Pazzani, M. J.: A Framework for Collaborative, Content-Based and Demographic Filtering. AI Review 13 (5/6), 393-408. (1999)

16. Zhi-Dan Zhao; Ming-Sheng Shang, "User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop," in Knowledge Discovery and Data Mining, 2010. WKDD '10. Third International Conference on , vol., no., pp.478-481, 9-10 Jan. 2010 doi: 10.1109/WKDD.2010.54

17. Ghuli, P.; Ghosh, A.; Shettar, R., "A collaborative filtering recommendation engine in a distributed environment," in Contemporary Computing and Informatics (IC3I), 2014 International Conference on , vol., no., pp.568-574, 27-29 Nov. 2014 doi: 10.1109/IC3I.2014.7019592

18. Kunhui Lin; Jingjin Wang; Meihong Wang, "A hybrid recommendation algorithm based on Hadoop," in Computer Science & Education (ICCSE), 2014 9th International Conference on , vol., no., pp.540-543, 22-24 Aug. 2014 doi: 10.1109/ICCSE.2014.6926520

19. Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters. Communications of the ACM, 2005,51(1):107 113.

20. Hadoop: Open source implementation of MapReduce, http://lucene.apache.org/hadoop/.

All Rights Reserved © 2016 IJARCET