© 2016 IJSRSET | Volume 2 | Issue 5 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Service Recommendation Using Rating Inference Approach

Yamini Nikam, M. B. Vaidya

AVCOE, Sangamner, Ahmednagar, Maharashtra, India

ABSTRACT

There is large amount of information available on World Wide Web. But all information is not relevant to a particular user. So this is the place where recommender system is useful. Recommender system guides the user. But traditional recommender systems has some limitations. So, to overcome this limitations we are using a hybrid algorithm which gives accurate result. Which combines the collaborative filtering with sentimental analysis. Therefore, implementing the algorithm distributed will reduce the required computing time. Apache Hadoop is an open-source distributed computing framework that can be composed of a large number of low-cost hardware to run the application on a cluster. It provides applications with a set of stable and reliable interface, Use of single recommended method is difficult to meet the demand of a large amount of data and accuracy requirements.

Keywords : Recommender System, Preferences, Map-Reduce, Top Rank.

I. INTRODUCTION

Collaborative Filtering (CF) is a widely used technique in recommender systems. It uses the database of user preferences, find the similar users which having similar tastes and according to that it provides recommendation to target users [1, 3].

User-log based and rating based are the two main types of CF-based recommender systems classified according to how they collect user preferences. User-log based CF obtains user preferences from implicit votes captured through users' interactions with the system (e.g. purchase histories as in Amazon.com [8]). Ratings based CF makes use of explicit ratings users have given items (e.g. 5-star rating scale as in MovieLens [2]). Such ratings are usually in or can easily be transformed into numerical values (e.g. A to E).

Some review hubs, such as the Internet Movie Database (IMDb), allow users to provide comments in free text format, referred to as user reviews. User reviews can also be considered as type of "user ratings", although they are usually natural language texts rather than numerical values. While research on mining user preferences from reviews, a problem known as sentiment analysis or sentiment classification (e.g. [4, 5, 7, 6]), is becoming increasingly popular in the text mining literature, its integration with CF has only received little research attention.

II. METHODS AND MATERIAL

A. System Architecture

The system architecture consist of following functions

1. Data Preparation
   - POS Tagging
   - Negation Tagging
   - Feature generalization
2. Review Analysis
3. Opinion Dictionary Construction

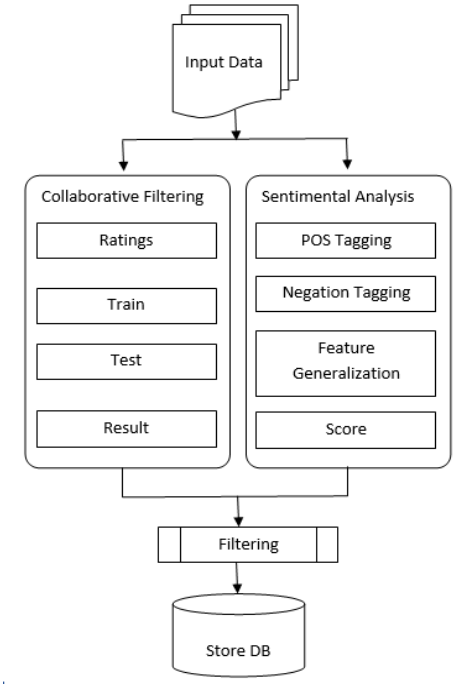
The simple algorithm for this filtering is given below:

Step 1 : Users input a User Name and Movie name.
Step 2 : Calculate the similarities, using
Step 3 : The similar items are passed on as arguments in the recommendation function which considers the items liked and viewed and

predicts the recommendation using the inputs from Step 2.

Step 4 : A weighted average of all these recommendations is calculated

Step 5 : The final recommendation is displayed to the user based on their weighted average



The recommendations are based on items preferred by the user and similar items and all related items. Below is a pseudo code for the same

For each Item i liked
For every User U who liked Item i
For each Item J liked by User U
Record Item (i,j)
Compute the similarity between Item i and Item j

Nearest neighbor (NN) models are often used in collaborative filtering and have been proven quite effective despite their simplicity.

Hadoop is used to calculate the similarity. The output of the Hadoop Map phase i.e. UserID and corresponding ItemID are passed to reduce phase. In reduce phase, output has been generated and sorted according to UserID. Output again has been stored in HDFS.

The basic idea behind it is for every pair of movies X and Y, find all the people who rated both X and Y. Use these ratings to form a Movie X vector and a Movie Y vector. Then, calculate the correlation between these two vectors. Now when someone watches a movie, you can now recommend him the movies most correlated with it. Our task is to find similarity between pair of item using correlation formula. Similarity(X, Y) = Correlation(X, Y) X and Y are items

$$Corr(X,Y) = \frac{n\sum xy - \sum x \sum y}{\sqrt{n\sum x^2 - (\sum x)^2}\sqrt{n\sum y^2 - (\sum y)^2}}$$

B. Algorithm

1. For pair of items find the users rated both the items X and Y

2. Form two vectors X and Y

U1	1	2
U2	5	4
U3	4	5
U4	3	2
U5	3	4

3. Calculate correlation between X and Y using the formula

Similarity(X, Y) =correlationi(X, Y)

We are using chaining of two MapReduce job. Output of the first job will work as input to the second.

Step-1 In first step dataset DATA file is given as input to the first MapReduce job. After completion it will generate output1 file which will work as input to MapReduce job-2.

Step-2 MapReduce job-2 will wait for the completion of MapReduce job-1, its output-1 will work as input to the MapReduce job-2.MapReduce job-2 will generate final output.

MapReduce Job-1: Work of the first MapReduce job is to collect the entire user rated both the items.

MapReduce Job-2: second MapReduce job will find the similarity between items using correlation formula.

III. RESULTS AND DISCUSSION

This **data set** consists of:

- 100,000 ratings (1-5) from 943 users on 1682 movies.
- Each user has rated at least 20 movies.
- Simple demographic info for the users (age, gender, occupation, zip)

The data was collected through the MovieLens web site (movielens.umn.edu) during the seven-month period from September 19th, 1997 through April 22nd, 1998. This data has been cleaned up - users who had less than 20 ratings or did not have complete demographic information were removed from this data set.

Results Analysis

1. Impact on parameters

Table 1. For dataset size 12500 reviews

User Count	117		
Item Count	1682		
Rating Count	12441		
Rating Density	6.32%		
Recommendation Type	**MAE**	**RMSE**	**AvgP**
Simple Recommendation	1.0138	1.2529	0.6797
Collaborative Recommendation	0.8099	1.0347	0.7767
SO+ Collaborative Recommendation	0.7952	1.0254	0.7683
Hadoop Recommendation	0.7689	0.9865	0.7715

Table 2. For dataset size 25,000 reviews

User Count	253		
Item Count	1682		
Rating Count	24890		
Rating Density	5.85%		
Recommendation Type	**MAE**	**RMSE**	**AvgP**
Simple Recommendation	1.0643	1.2971	0.6939
Collaborative Recommendation	0.8499	1.0837	0.8048
SO+ Collaborative Recommendation	0.7944	1.0248	0.804
Hadoop Recommendation	0.7755	0.9988	0.8157

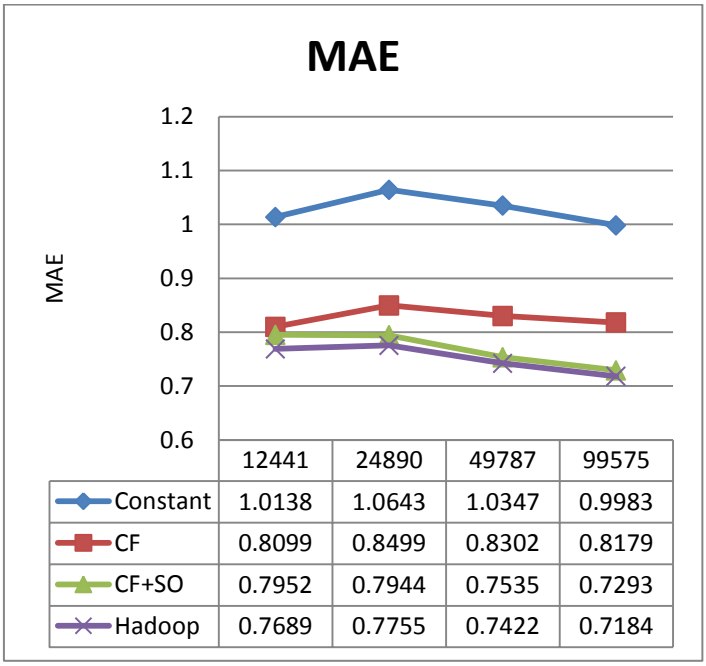
Table 3. For dataset size 50,0000 reviews

User Count	445		
Item Count	1682		
Rating Count	49787		
Rating Density	6.65%		
Recommendation Type	**MAE**	**RMSE**	**AvgP**
Simple Recommendation	1.0347	1.2718	0.6817
Collaborative Recommendation	0.8302	1.0456	0.802
SO+ Collaborative Recommendation	0.7535	0.9693	0.8085
Hadoop Recommendation	0.7422	0.9493	0.8115

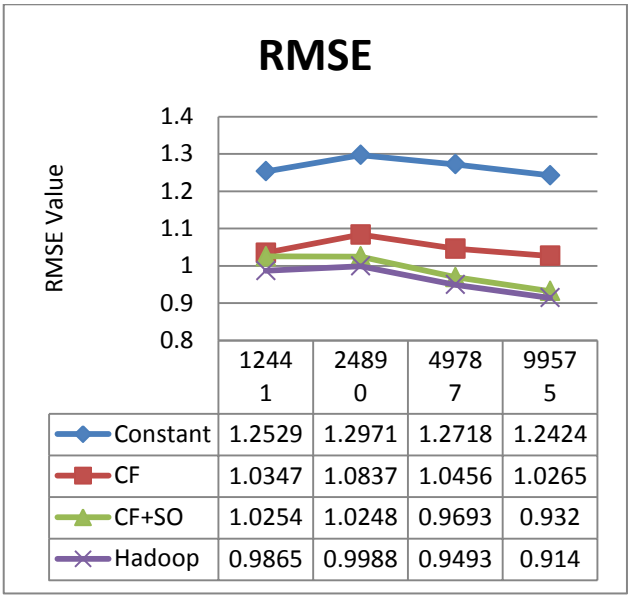
Table 4. For dataset size 1,00,000 reviews

User Count	938		
Item Count	1682		
Rating Count	99575		
Rating Density	6.31%		
Recommendation Type	**MAE**	**RMSE**	**AvgP**
Simple Recommendation	0.9983	1.2424	0.6641
Collaborative Recommendation	0.8179	1.0265	0.7981
SO+ Collaborative Recommendation	0.7293	0.932	0.8119
Hadoop Recommendation	0.7184	0.914	0.8149

- **MAE**



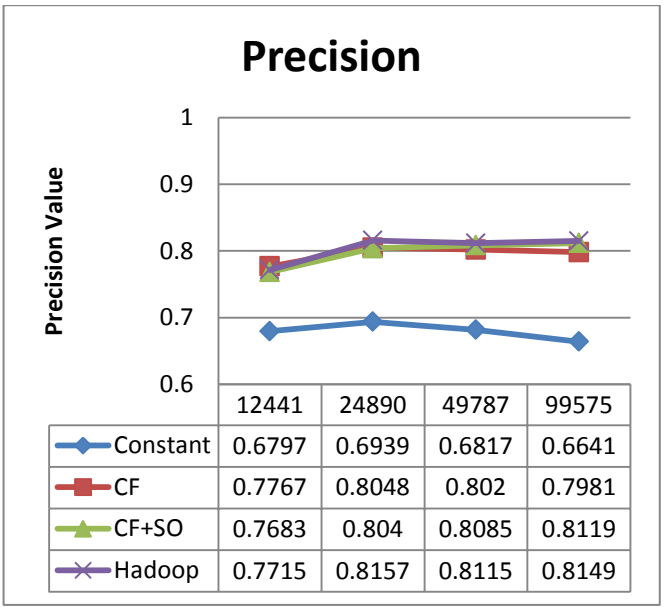
- **RMSE**



	12441	24890	49787	99575
Constant	1.2529	1.2971	1.2718	1.2424
CF	1.0347	1.0837	1.0456	1.0265
CF+SO	1.0254	1.0248	0.9693	0.932
Hadoop	0.9865	0.9988	0.9493	0.914

- **Precision**

It is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved.



	12441	24890	49787	99575
Constant	0.6797	0.6939	0.6817	0.6641
CF	0.7767	0.8048	0.802	0.7981
CF+SO	0.7683	0.804	0.8085	0.8119
Hadoop	0.7715	0.8157	0.8115	0.8149

IV. CONCLUSION

Now-a-days the recommender system is continuously expanding, so the number of users and items also growing exponentially. So, there is need of algorithms which performs better on large data sets. So, to solve the traditional recommender systems limitations and to improve performance here the proposed rating inference approach integrates sentiment analysis and CF. Such approach transforms user preferences expressed as unstructured, natural language texts into numerical scales that can be understood by existing CF algorithms. It improves speed and solves high recommendation performance under large data sets. Experiments show that the improved parallel hybrid recommendation algorithm compared with the former one improves the speed and reduces the time consumption.

V. REFERENCES

[1] Kai Yu, Xiaowei Xu, Jianhua Tao, Martin Ester,Hans-Peter Kriegel "Instance selection techniques for memory based collaborative filtering", Proc. Second SIAM International Conference on Data Mining (SDM'02)

[2] Prem Melville and Vikas Sindhwani, "Recommender Systems", Encyclopedia of Machine Learning, 2010.

[3] John S. Breese, David Heckerman, and Carl Kadie (1998). "Empirical analysis of predictive algorithms for collaborative filtering". In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence (UAI'98).

[4] Good, N., Schafer, J. B., Konstan, J. A., Borchers, A., Sarwar, B., Herlocker, J., et al. (July 1999). "Combining collaborative filtering with personal agents for better recommendations". In Proceedings of the sixteenth national conference on artificial intelligence (AAAI-99), Orlando, Florida (pp. 439-436).

[5] Xiaoyuan Su, Taghi M. Khoshgoftaar, "A survey of collaborative filtering techniques", Advances in Artificial Intelligence archive, 2009.

[6] 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.

[7] Montaner, M.; Lopez, B.; de la Rosa, J. L. (June 2003). "A Taxonomy of Recommender Agents on the Internet". Artificial Intelligence Review 19 (4): 285–330. doi:10.1023/A:1022850703159.

[8] Greg Linden, Brent Smith, and Jeremy York, "Amazon.com Recommendations" IEEE Computer Society, 1089-7801/03 Feb-2003.

[9] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? Sentiment classification using machine learning techniques”, in Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 79–86, (2002).
[10] P.D. Turney, “Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews”, in Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 417–424, (2002).
[11] M. Hu and B. Liu, “Mining and summarizing customer reviews”, in Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177,(2004).
[12] B. Pang and L. Lee, “Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales”, in Proceedings of the 43rd Annual Meeting of the Association for Computation Linguistics, pp. 115–124, (2005).
[13] L. Terveen, W. Hill, B. Amento, D. McDonald, and J. Creter, “Phoaks: A system for sharing recommendations”, Communications of the ACM, 40(3), 59–62, (1997).
[14] Schelter, S., Bod en, C., et al., "Scalable similarity-based neighborhood methods with Map Reduce". Proc. 6th ACM conference on Recommender systems, Dublin, Ireland, pp. 163- 170,2012.
[15] Hadoop: Open source implementation of MapReduce,http : //lucene.apache.org/hadoop/.
[16] Lammel, R.: Google's MapReduce Programming ModelRevisited. Science of Computer Programming 70, 1-30, 2008.
[17] Zhao, W., Ma, H., et al.: Parallel K-Means Clustering Based on MapReduce. In: CloudCom 2009. LNCS, vol. 5931, pp. 674- 679,2009.
[18] K. Shvachko, H. Kuang, S. Radia, and R. Chans1er, "The Hadoop distributed file system," Mass Storage Systems and Technologies, IEEE/ NASA Goddard Conference, vol. 0, pp. 1-10,2010.