Mody University International Journal of Computing and Engineering Research
2 (2), 2018, 57-60
http://www.publishingindia.com/micer

Efficiency Booster Techniques for Recommendation System

Neetu Singh[1*] and V. K. Jain[2]

[1]Student, Dept. of Computer Science, Mody University of Science and Technology, Sikar, Rajasthan, India. Email: nits.tanwar29@gmail.com
[2]Dean, School of Engineering and Technology, Mody University of Science and Technology, Sikar, Rajasthan, India. Email: dean.cet@modyuniversity.ac.in
*Corresponding Author

Abstract: Recommendation Systems (RS) work as guides, it is guiding users to find products of their interest. It's a fact that with the increase in RS deployment, traditional methods need modifications. Many techniques and different approaches have been developed to generate an effective recommendation. This is interesting as different application's scenarios could have a fittest solution. This article throws light on techniques to turn the traditional methods more useful for real-world scenario to boost the productivity of RS. Finally, proposed a similarity fusion method for increasing efficiency of a recommendation system.

Keywords: Algorithm, Efficiency, Recommender systems, Similarities.

I. Introduction

Recommender Systems (RS) work as guides, it is guiding users to find products of their interest. The exponential evolution of the web technology along with growing footsteps of online commerce applications has resulted in the expansion of recommender system. It is a custom-made provider of information to recognize item sets that will be of concern to a particular user. Recommender systems are the base for the future of the smart webs. The systems produce better experience for user by making information retrieval easier [1] and divert users from queries typing phase towards hit it off suggested links. No one is untouched by real-life recommender systems. They are doing amazing work, when browsing for music, movies, news or books. These engines are mandatory for websites like Amazon, Myntra or Netflix. On the basis of different approaches used for development of recommender systems such as demographic, content, or historical information [2,3,4], user centered collaborative filtering came out as the most widespread and promising means for structuring recommender systems till date. Yet it is most successful but unluckily, the linear growth of its computational complexity with the count of customers can grow to be several millions as witnessed in commercial applications. To overcome this scalability issues, item-item similarity is used instead of user-user similarity. It generates user-item matrix to recognize relations among the diverse items, and use them to produce recommendations [5,6]. The progress of RS has shown the significance of hybrid techniques, which is the combination of different techniques in order to produce the benefits of each of them. A survey on hybrid recommenders has been discussed in [7]. While the current surveys focus on the most apt methods used and best suited algorithms of the recommendation system, our paper instead tries to focus on how to increase the efficiency of the particular RS: from a traditional phase to modified one.

The balance of the paper is structured by different sections: section II will outline the recommendation system. The recommended similarity fusion algorithm have been elaborated in section III and finally the concluding remarks and proposed future work in section IV.

II. Recommendation System Process

Recommendation process can be break into three modules:

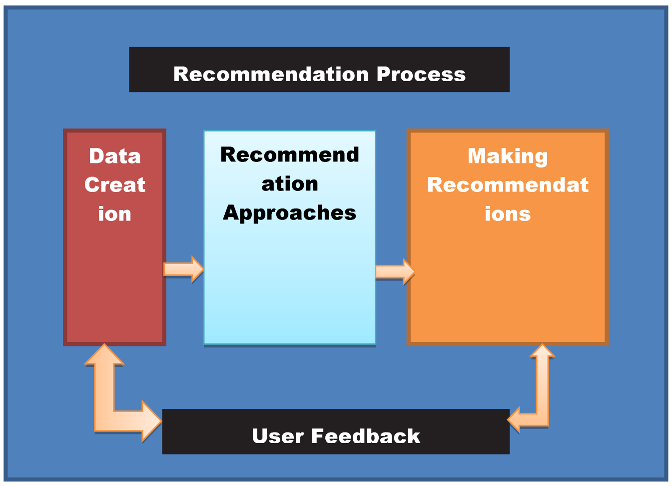

Fig. 1: Recommendation System Process

A. Phase 1: Data Creation

Phase 1 is all about creating user-item matrix and considering item & user ontology. Data creation is very important in RS.

Few techniques consider only basic information like rankings or one can say ratings while others act on extra knowledge like social as well personal constraints, and real time activities in scenario of distributed systems.

Basically for recommendations, there should be something common between users or the items. That means base consist of item, user and the tractions between them. In collaborative filtering, we have preferences; these are in form of user ID, item ID and the user preference for the item.

Efficiency Booster Techniques:

- *Improvement of Rating Prediction*:

Information gathering is a very important step for RS. To increase efficiency of RS, we need improvement in ratings and intelligent dealing of problems like cold start problem. For example quality of movie recommenders can be significantly enhanced by considering the account information obtainable in other similar organizations. This appears apparent for the big organizations like Netflix and Amazon prime movies. Both have lots of item in common. The latest exploration in this area has been dedicated on rating improvement valuations delivered by the recommendation system [8], [9].

- *Dividing Datasets:*

CF requires huge amount of computations. So, better performance one can divide for original large dataset into smaller one, which do not include the unwanted or extra information like presence of semantic information, and after that we can apply the appropriate algorithms to attain better results.

- *Data Dimensionality*:

Other method is reduction in data dimensionality, for example principle component analysis and singular value decomposition [10], is usually employed to enhance the performance of RS. A hybrid recommendation method involving two-stage data handling and processing, dealing with content specifications, describing items and tendering user behavioral data is discussed in [11].

B. Phase 2: Recommendation Approches

After completion of phase 1, we got a clear idea about what parameters are suitable for getting similarity measures between items and users. So we need only that information that is meaningful and could be passed to recommender. Mahout Library facilitates us with similarity models namely: Pearson correlation similarity, Euclidean distance similarity, Spearman correlation similarity, Cosine measure similarity and Tanimoto coefficient similarity. Last one is set of operations generally represented by binary value and is effective only when user expressed some preference for an item. Pearson similarity and cosine similarities are quite related to each other. Major difference between two is that cosine similarity works on centred data. Euclidean distance between two vectors became Euclidean similarity when we get negative proportion to this distance. Spearman similarity is same as Pearson correlation similarity except that value xi and yj are replaced with their relative ranks. The detailed explanations are in the javadoc API of Mahout [12]. In this phase, requirement is to calculate the 'n' neighbor of the current user 'u' and pass size of neighbourhood as parameter in Item based Recommender. KNN algorithms are used for the purpose. Size of the neighbourhood is important for the accuracy of the algorithm. The utmost similar users are taken (the number is limited by the neighbourhood size).

Efficiency Booster Techniques:

- *Modification in Similarity Model:*

The general methods to measure similarity such as cosine [13], Pearson correlation coefficient [14], mean squared difference [15] are actually insufficient to catch the actual similar users, especially in cold start problem. Well, this is one of the desired research topics for researchers and lots of research is going on. To increase the accuracy, some new methods have recommended by researchers for similarity measures as PIP (Proximity-Impact-Popularity) [16], [17] discussed the weighted Pearson correlation coefficient for similarity.

- *Modification in Algorithm:*

Some change in algorithm, also leads to efficiency improvement [18], proposes a new personal technology algorithm e-commerce recommender system.

C. Phase 3: Making Recommendations

Finally, recommender generates some ranks based on preferences and similarities. A list of most suitable items for recommendation has been generated and passed to user. Good recommenders suggest best suited items to user [19].

Efficiency Booster Techniques:

Creating a balance between presenting rank and diversity: The traditional method of ranking was dependent upon ranks, later on it was turned to semantic information. But fact is, still recommenders sometime recommends diverse set of items, for example the recommendations for top-3 ranked movie recommender for a user, watching Star Wars sequels may get a broader sight of the highly graded movies according to the recommender may instead of a fact that user may like this series. Say, result includes a movie of Star Wars, but also other movies like Star Trek or E.T. So for improving the recommendation system, one can focus on the balance of ranking Vs. diversity by applying some weights or tags or Priority-medoids [20-22].

III. Proposed Similarity Fusion Algorithm

The similarity has been calculated taking for the items that user with alike liking opted previously and on the basis of this similarity measure, the recommendations to the active users will be given. Traditionally, any similarity model namely: Pearson correlation similarity, Euclidean distance similarity, Spearman correlation similarity, Cosine measure similarity and Tanimoto coefficient similarity can be used. But in our proposed method, we used fusion similarity measure. It is the combination of the two similarity measure namely Pearson and Euclidean. Pearson is a range of numbers between -1 to +1. According to this similarity, the two series share the linear relationship, when we measure the tendency of two numbers move proportionally. Whereas, in Euclidean distance users are seen as points in multidimensional space and the coordinates of these points are seen as preferences value. The Euclidean similarity measure is inversely proportional to the Euclidean distance. But in fusion method, similarity of the common item by adding both similarity measures that in turn increase the similarity measure of item, hence help recommender to place most apt item plan to the customer. It clearly affects recommender accuracy.

Algorithm is as follows:

For every item I1 in item ontology

For every user U in user ontology who bought I1

For each item I2 bought by user U

Record that a user bought both I1 and I2

For each item I2

Calculate the similarity S1 between I1 and I2 using Pearson

Calculate the similarity S2 between I1 and I2 using Euclidean

Calculate the Improvised similarity between I1 and I2 by adding S1 and S2

IV. Conclusion and Future Work

CF is a striking topic for scholars owing to its competency to handle surplus information efficiently. Many techniques and different approaches have been developed to generate an effective recommendation. This is interesting as different application's scenarios could have a fittest solution. In the current research paper, the different techniques that are capable to enhance the efficiency of RS. The proposed similarity fusion technique is assumed to provide enhanced forecast accuracy. We intend to extend the recommended algorithm in a real-world recommender system and evaluate its efficiency.

References

[1] J. Liu, T. Zhou, and B. Wang, "Research process of personalized recommendation system," *Progress in Natural Science*, vol. 19, no. 1, pp. 1-15, 2009.

[2] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," UAI, 1998.

[3] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: An open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW'94),* pp. 175-186, Chapel Hill, North Carolina, USA, 22-26 October 1994.

[4] J. B. Schafer, J. Konstan, and J. Riedl, "Recommender systems in e-commerce," in *Proceedings of the 1st ACM Conference on Electronic Commerce (EC'99)*, pp. 158-166, Denver, Colorado, USA, 03-05 November 1999.

[5] B. Mobasher, H. Dai, T. Luo, M. Nakagawa, and J. Witshire, "Discovery of aggregate usage profiles for web personalization," in *Proceedings of the WebKDD Workshop*, 2000.

[6] L. H. Ungar, and D. P. Foster, "Clustering methods for collaborative filtering," in *Workshop on Recommendation Systems at the 15th National Conference on Artificial Intelligence*, 1998.

[7] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331-370, 2002.

[8] S. Renaud-Deputter, T. Xiong, and S. Wang, "Combining collaborative filtering and clustering for implicit recommender system," in *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, pp. 748-755, 2013.

[9] J. Chen, Y. Chen, X. Du, C. Li, J. Lu, S. Zhao, and X. Zhou, "Big data challenge: A data management perspective," *Frontiers of Computer Science*, vol. 7, no. 2, pp. 157-164, 2013.

[10] M. Jahrer, A. Toscher, and R. Legenstein, "Combining predictions for accurate recommender systems," in *Proceedings of the ACM Conference on Knowledge Discovery in Databases (KDD'2010),* 2010.

[11] N. Lathia, S. Hailes, L. Capra, and X. Amatriain, "Temporal diversity in recommender systems," in *Proceedings of the SIGIR 2010*, Geneva, Switzerland, 19-23 July 2010.

[12] Mahout, Apache Software Foundation, 01 June 2018. [Online]. Available: https://mahout.apache.org/. [Accessed 11 September 2018].

[13] Y. Li, C. X. Zhai, and Y. Chen, "Exploiting rich user information for one-class collaborative filtering," *Knowledge and Information Systems*, vol. 38, no. 2, pp. 277-301, 2014.

[14] R. Bambini, P. Cremonesi, and R. Turrin, "A recommender system for an IPTV service provider: A real large-scale production environment," in F. Ricci *et al*.

(eds.), *Recommender Systems Handbook*, Springer, New York, NY, 2011.

[15] D. Singh, and C. K. Reddy, "A survey on platforms for big data analytics," *Journal of Big Data*, Springer open journal, pp. 1-8, 2014.

[16] V. D. Blondel, A. Decuyper, and G. Krings, "A survey of results on mobile phone datasets analysis," *EPJ Data Science*, pp. 4-10, 2015.

[17] G. Miritello, L. Rubén, M. Cebrian, and E. Moro, "Limited communication capacity unveils strategies for human interaction," *Scientific Reports,* vol. 3, article no. 1950, 2013.

[18] A. Szwabe, M. Ciesielczyk, and T. Janasiewicz, "Semantically enhanced collaborative filtering based on RSVD," in *Proceedings of the International Conference on Computational Collective Intelligence*, pp. 10-19, 2011.

[19] Y. Guo, and G. Deng, "An improved personalized collaborative filtering algorithm in e-commerce recommender system," *International Conference on Service Systems and Service Management*, pp. 1582-1586, IEEE, 2007. Y. Wang, B. Liang, W. Ji, S. Wang, and Y. Chen, "An improved algorithm for personalized recommendation on MOOCs," *International Journal of Crowd Science*, vol. 1, no. 3, pp. 186-196, 2017.

[20] Y.-A. de Montjoye, Z. Smoreda, R. Trinquart, C. Ziemlicki, and V. D. Blondel, "D4D-Senegal: The second mobile phone data for development challenge," 2014. ArXiv:1407.4885.

[21] M. Deshpande, and G. Karypis, "Item-based top-n recommendation algorithms," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 143-177, 2004.

[22] N. Singh, P. Kumar, and A. K. Dahiya, "RWYW: Recommend what you want - A recommender for mobile plans," *International Journal of Innovations & Advancement in Computer Science*, vol. 7, no. 2, pp. 135-144, 2018.