LITERATURE REVIEW: High performing Movie Recommender System using Parallel Computing

Raghad Rowaida
School of Computer Science
Carleton University
Ottawa, Canada K1S 5B6
raghadrowaida@cmail.carleton.ca

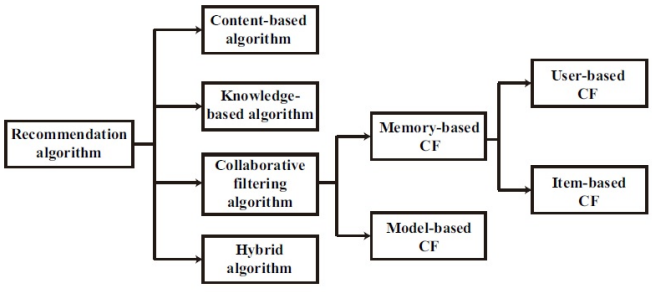
October 2, 2019

1 Introduction

Building an efficient Recommender system is a challenge. As, Recommender systems are associated with big data, and their efficient use. In October 2006, Netflix organized a competition to find the best algorithm for user rating prediction. Several high performing approaches has been proposed. Still many approches are being introduced to improve recommendation performance. Parallel computing is highly associated with these vast amount of data processing. Frameworks such as, Apache Hadoop, Apache Mahout, are being used to build good recommendation systems. The purpose of this project is to study some of the exiting techniques to build an efficient recommendation system, and building a recommendation system of our own.

2 Literature Review

According to [1], algorithmic approaches to build a recommendation system can be classified in to below categories:



- Content based algorithm mostly focuses on user profile information and object profile information. Then try to match them accordingly. This one is not that efficient for movie recommendation system.
- Collaborative filtering (CF) methods considers past history of users' preferences, activities, and behaviors and recommend items based on the similarities to other users. It is so far the most efficient system of them all. [1][2]

- Hybrid is a composition of Content-based and CF method.

Collaborative Fitting itself is a vast area of recommendation system. Several algorithms have been developed to get the best outcome of it. Principle Component Analysis (PCA), Singular Value Decomposition (SVD), Matrix Factorization are some of them. Among them, Matrix Factorization holds very high accuracy [2]. This method has been further extended to get even more better result. CF works on both binary, and non binary grades[3].

Dimension reduction[4] by using SVD to find out neighborhood cluster based on radius distance D is one way to do it.

Matrix Factorization for recommendation: The idea is to construct a bipartite graph of user and item based on their past rating. Then find the co-clusters of the graph based on cluster of user who liked the cluster of movies. when a uses becomes members of this co-cluster, then any non connected movie node will be the recommended movie for the user.

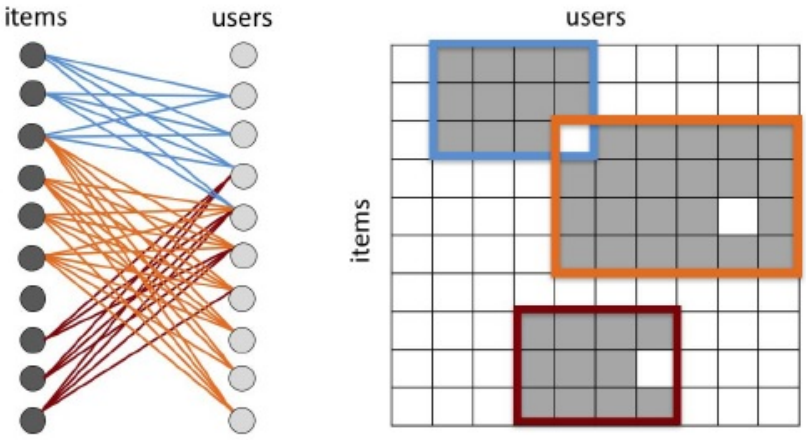


Figure 1: A bi-partitite graph of items and users with three co-clusters. On the right, the bi-partite graph and the co-clusters are shown in matrix format. The white spots inside the co-clusters represent good recommendations.

Step one is to calculate latent factor between user and item. The model assumes that there exist K co-clusters. The affiliation strength of a user u with each of the K co-clusters is given by a K-dimensional vector f_u and the affiliation strength of an item i with the K co-clusters by a K-dimensional vector f_i.

Let $f_{i,k}$ denote the kth entry of f_i, and similarly, let $f_{u,k}$ denote the kth entry of f_u, where k in {1..K} and denotes the co-cluster. The larger the affinity between an item i and a co-cluster k, the larger is the value of $f_{i,k}$. When there is no affinity between an item i and a co-cluster k, we have $f_{i,k} = 0$. Given a history of the ratings the users have expressed for the items, algorithm iteratively updates the f_i and f_u.Matrix Factorization is used to find the optimal cluster from these vectors.

There are some major issues[5] that can effect the performance of a recommendation

system. Those are, scalability, cold start, sparsity. Several new and improved method have address this problems with solutions.

Now the involvement of parallel computing in the above procedures can be in many forms. Mostly the Matrix Factorization process is divided into individual operations which are computed in parallel. Both CPU and GPU computing is possible for the calculation. Different big data mining frameworks like Hadoop[6][7] can be used in case of Content-based analysis, Data pre-processing in CF, or in Hybrid approaches. The knowledge-based recommender algorithms are working for the 'surprise' factor[8], that provides rich recommends outside of users' expectation. Machine learning is also playing a role in recommender systems[9].

References

[1] Jiang Zhang, Yufeng Wang, Zhiyuan Yuan, and Qun Jin. *Personalized Real-Time Movie Recommendation System: Practical Prototype and Evaluation.* Tsinghua Science and Technology, vol. 25, no. 2, pp. 180-191, April 2020.

[2] Kubilay Atasu, Thomas Parnell, Celestine Dunner, Michail Vlachos, and Haralampos Pozidis. *High-Performance Recommender System Training using Co-Clustering on CPU/GPU Clusters.* 2017 46th International Conference on Parallel Processing, Bristol, 2017, pp. 372-381.

[3] Azar Yossi , Nisgav Aviv, Patt-Shamir Boaz. *Recommender systems with non-binary grades.* Proceedings of the twenty-third annual ACM symposium on Parallelism in algorithms and architectures (SPAA '11). ACM, New York, NY, USA, 245-252. DOI: https://doi.org/10.1145/1989493.1989528

[4] H. Zarzour, Z. Al-Sharif, M. Al-Ayyoub and Y. Jararweh. *A new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques.* 2018 9th International Conference on Information and Communication Systems (ICICS), Irbid, 2018, pp. 102-106.

[5] S. Jain, A. Grover, P. S. Thakur and S. K. Choudhary, *Trends, problems and solutions of recommender system.* International Conference on Computing, Communication Automation, Noida, 2015, pp. 955-958.

[6] G. Wei-wei and L. Feng. *Application Research of Hadoop's Weibo Recommendation System Prototype Based on Customer Dynamic Behavior.* 2018 3rd International Conference on Smart City and Systems Engineering (ICSCSE), Xiamen, China, 2018, pp. 562-566.

[7] S. Saravanan. *Design of large-scale Content-based recommender system using hadoop MapReduce framework.* 2015 Eighth International Conference on Contemporary Computing (IC3), Noida, 2015, pp. 302-307.

[8] A. Al-Doulat. *Surprise and Curiosity in A Recommender System.* 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA), Aqaba, 2018, pp. 1-2.

[9] J. Lund, Y. Ng. *Movie Recommendations Using the Deep Learning Approach.* 2018 IEEE International Conference on Information Reuse and Integration (IRI), Salt Lake City, UT, 2018, pp. 47-54.