Movie Recommendation System Using ML

1st Atharv Nevase
Trinity college of engineering and research

3rd Yash Darekar
Trinity college of engineering and research

2nd Abhay Phadatre
Trinity college of engineering and research.

4th Jayant Borate
Trinity college of engineering and research

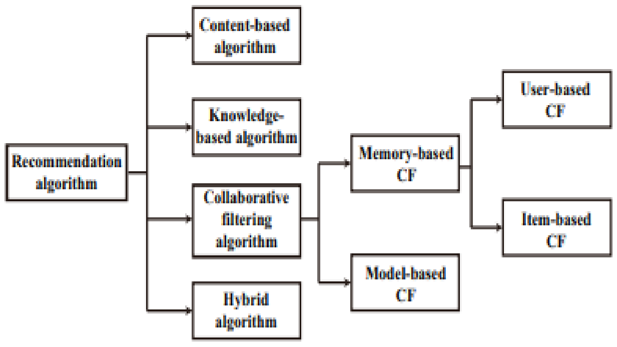
Guided by :- Prof . Ruchira Tare

Abstract -With the eruption of big data, practical recommendation schemes are now very important in various fields, including e-commerce, social networks, and a number of web-based services. Nowadays, there exist many personalized movie recommendation schemes utilizing publicly available movie datasets (e.g., MovieLens and Netflix), and returning improved performance metrics (e.g., Root-Mean-Square Error (RMSE)). However, two fundamental issues faced by movie recommendation systems are still neglected: first, scalability, and second, practical usage feedback and verification based on real implementation. In particular, Collaborative Filtering (CF) is one of the major prevailing techniques for implementing recommendation systems. However, traditional CF schemes suffer from a time complexity problem, which makes them bad candidates for real-world recommendation systems. In this paper, we address these two issues. Firstly, a simple but high-efficient recommendation algorithm is proposed, which exploits users' profile attributes to partition them into several clusters.

I. INTRODUCTION

With the growth of big data generation across various fields, information overload is becoming a critical problem. To address this, a number of Recommendation Systems (RS) have been developed to help consumers find items of interest and decide between products among huge databases. These systems have been applied to various products and services on the internet, including film and video, music, social networking, reading, news, and personalized e-mail and advertising. Three of the domains in which RSs are used most frequently are movies, documents, and product reviews, mainly because of the ease in accessing test data[1–4] . For instance, in the movie domain, MovieLens and Netflix are two online datasets of movie ratings. The movie industry is a prodigious producer of video.

Already by the year 2000, it was reported that more than 4500 movies are released every year around the world, spanning approximately 9000 hours of video. With such a massive amount of choice, there is great demand for technologies that enable viewers to access movies of interest conveniently and therefore facilitate movie distribution. between two parties with the aid of blockchain technology. The smart contract contains all of the deal's specifics and lowers third parties' expectations. Our system contains two smart contracts: one for event creation Two fundamental issues in real movie recommendation systems are often neglected: scalability and practical usage feedback/verification based on real implementation. Collaborative Filtering (CF) is one of the most widely-used algorithms for making rating predictions within an RS. It is based on the core assumption that users who have expressed similar interests in the past will share common interests in the future. Therefore, the idea of collaborative filtering is to identify users in a community that share appreciation for similar things. Intuitively, if two users have rated the same or almost the same items, then they have similar tastes and can therefore be assigned to a group or close neighborhood. A user can receive recommendations for those items that he/she has not rated before, but that have been already positively rated by users in his/her neighborhood. As the number of users and items grows



3 Proposed Recommendation Algorithm Intuitively, we can assume that people who share the same gender, similar age, similar occupations, etc., are likely to share similar tastes in movies, and assign similar ratings to movies. Based on this assumption, we designed a CF algorithm, KM-Slope-VU, in which, according to users' profile attributes, K-means is utilized to partition users into several clusters, and then each cluster produces an opinion leader by calculating the average rating of the items. We also intuit that the historical data of user evaluations of items are naturally correlated to their tastes, and therefore should be utilized to cluster users. Specifically, for each cluster, a virtual opinion leader acts on behalf of other users in the same cluster to rate all items, and then all of virtual opinion leader's ratings are gathered to approximate the real user's rating data. That is, the original user-item rating matrix is replaced with a reduced virtual opinion leader-item rating matrix, meaning that the user-item matrix is compressed and the computational complexity is reduced. Note that the key idea behind the traditional Slope One algorithm is to predict a user's rating for an item based on the score differences between pairs of items. Following this idea, the proposed Weighted Slope One-VU works on the intuitive principle of a popularity differential between items for virtual users in a pairwise fashion, i.e., determines hows much more liked one item is compared to another. One way to measure this differential is to simply subtract the virtual user's average rating of the two items. In turn, this difference can be used to predict real users' ratings of one of those items, given their rating of the other. In the training phase, we first use K-means to find user neighbors based on profile characteristics, such as age, sex, and profession. Secondly, the average rating of each item is calculated under the same cluster to form a virtual user who represents all users in this cluster. For example, if there are users and movies, that is, the original user-item rating matrix is Rmn, and the K-means outputs K clusters, implying that there are K virtual users, then the virtual user-item rating matrix is uvRkn, where uv represents virtual users. Finally, this virtual user-item rating matrix uvRkn is fed into our proposed Weighted Slope One-VU algorithm to infer the recommendation results. In the recommendation phase, our system makes a prediction for users according to the collected user history data. We use the following notations to describe our scheme. The number of elements in a set S is card.S /; ui represents the rating given by user u to item i; vui denotes the rating given by virtual user vu to item i. The subset consisting of all items rated by the user u is S.u/, and Pu;i represents the predicted rating of user u on item i.

II. Related Work

As shown in Fig. 1, existing recommendation algorithms can be divided into four kinds: contentbased, knowledge-based, CF, and hybrid. Among these recommendation algorithms, CF is the most popular technique, based on the core assumption that users who have expressed similar interests in the past will share common interests in the future[5]. CF methods can be model-based or memory-based. Model-based algorithms first construct a model to represent user behavior and, therefore, to predict their ratings. The parameters of the model are estimated using the data from the rating matrix. There are many model-based approaches: Principal Component Analysis (PCA) and SVD are based on algebra[6, 7] , Bayes methods are based on statistics[8]. Matrix factorization for recommender systems has been a special focus of a voluminous amount of research, especially since the Netflix Prize competition was announced. This methodology transforms both items and users to the same latent factor space, thus making them directly comparable. The latent space tries to explain ratings by characterizing both products and users on factors automatically inferred from user feedback. For example, when the products are movies, factors might measure obvious dimensions such as comedy vs. drama, amount of action, or orientation to children; less well defined dimensions such as depth of character development or quirkiness; or completely uninterpretable dimensions[9–11]. A prevalent assumption in constructing matrix approximations is that the partially observed matrix is low-rank. LLORMA[12] significantly relaxes this low-rank assumption; instead of assuming that the user-item matrix M can be globally approximated by a low-rank matrix, this method assumes that the matrix M behaves as a low-rank matrix in the vicinity of certain rowcolumn combinations. Therefore several low-rank approximations of M are constructed, each being accurate in a particular region of the matrix. Then a smoothed convex combination of those low-rank matrices is provided to finally approximate the observed matrix as a weighted sum of low-rank matrices. In brief, unlike standard low-rank matrix approximation techniques achieving consistency in the limit of large data (convergence to the data generating process) by assuming that M is low-rank, this local method achieves consistency without the low-rank assumption. Memory-based approaches can be categorized as user-based or item-based. User-based collaborative filtering algorithms generate recommendations based on the preference of similar users[13, 14]. In contrast to user-based CF, item-based CF approaches recommend items on the basis of information about other items that a user has previously rated[15, 16]. The recommended items for the given user are ranked by the similarities between each candidate item and other items that the user has rated. Since the rating data of each virtual user emerging from a cluster of similar users is used to predict user ratings, our proposed KMSlope-VU method belongs to the class of userbased CF methods. Reference [17] designs a movie recommendation system using data clustering and computational intelligence, designing an algorithm featuring K-means clustering and cuckoo search optimization, and evaluating the recommendation performance on the MovieLens dataset. Reference [18] develops a hybrid clustering model to improve the movie prediction accuracy and make more pertinent recommendations to users. These authors used a combination of K-means Particle Swarm Optimization (PSO) to find initial centers, which is much more accurate and precise than assigning centers randomly. These centers are then used by a fuzzy Cmeans for optimization to form the final clusters, which are directly used for the result calculation process. In order to avoid the premature convergence of K-means clustering, Ref. [19] considers a genetic algorithm as the optimization tool for evolving initial seeds in the first step of the K-means process to identify optimal partitions. Experimental results illustrate that their approach is capable of providing more reliable movie recommendations in comparison with the existing Jiang Zhang et al.: Personalized Real-Time Movie Recommendation System: Practical Prototype and Evaluation 183 cluster-based CF methods. However, these recommendation schemes have two common problems. First, based on a user-item rating matrix, K-means is directly employed to find all like-minded users, called neighbors. However, as the number of users and items increases, the dimension of the user-item rating matrix increases quickly, which slows the operation of the clustering algorithm. Second, considering that the sensitivity selection of initial seeds in K-means could influence the final output and easily fall into local optimum, in order to find the nearest neighbor of the user and improve prediction accuracy in RS, K-means often requires a combination of other heavy-computing intelligence algorithms, such as a genetic algorithm like cuckoo search optimization, to form a hybrid clustering model, which also increases the time complexity. To enable real-time recommendations, it is imperative to solve the scalability problem in a practical movie recommendation system. In response, we propose a simple and scalable KM-Slope-VU recommendation algorithm, with three main features. First, instead of user ratings on items (i.e., user-item matrix), K-means is employed to find several groups of similar users based on their typical profile attributes (age, occupation, etc.), to ensure that the time overhead of clustering does not increase as the number of items increases. Second, each cluster produces a virtual opinion leader (i.e., virtual user) to represent all of the real users in the corresponding cluster, by calculating average rating of the items. Finally, the formed virtual user-item rating matrix is fed into the Weighted Slope One-VU, an improved and lightweight recommendation algorithm based on Slope One, to generate the recommendation results[20]. Compared with the raw user-item matrix, the dimensions of the virtual opinion leader-item matrix are greatly reduced. All of the features above significantly accelerate the classification model training and ensure recommendations can be delivered in real time.

III. LITERATURE-SURVEY

The era of information and communication technology makes the information available on the internet growing rapidly. Recommender Systems are one of the technologies that are widely used to filter information to handle the huge of information. One of the developing information is film. The increasing number of films released every year has led to the development of applications that offer movie streaming services such as Netflix, Yiu, Disney Hotstar, etc. Therefore, movie recommender systems technology is needed to facilitate and provide a good experience when users use these services. The purpose of this study is to conduct a Systematic Literature Review (SLR) to analyze methods against the algorithm developed in building a movie recommender system. SLR method consists of three stages, namely, planning, conducting, and reporting processes. Studies published from 2010 to 2020 were considered. There were 21 main studies in which the collaborative filtering method was used in 16 studies, knowledge-based filtering was used in 2 studies, and hybrid filtering method was used in 3 studies. The results of the SLR process can be concluded that there are advantages and disadvantages to each method developed in building the movie recommender system. However, the model-based collaborative filtering method is one method that can minimize cold start, data sparsity, and scalability problems.

IV. IMPLEMENTATION

We have implemented hybrid movie recommendation system A combination of both algorithms were made and created.

Fig. 3 User Interface

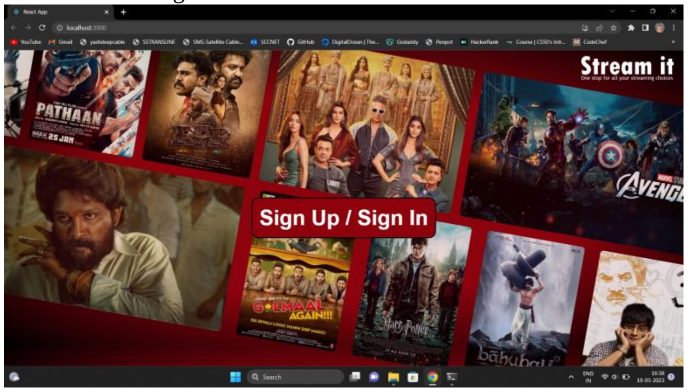


Fig 4: Real Time Result Updation

V. CONCLUSIONS AND FUTURE WORK

In this paper, we developed a novel collaborative filtering approach called Weighted KM-Slope-VU for fast and scalable movie recommendations, and furthermore developed and deployed a personalized movie recommendation site, MovieWatch, to provide users with viewing services and collect user feedback on recommended movies to practically evaluate our proposed algorithm using real-life data. Specifically, we adopted K-means to partition users into several clusters, and then for each cluster conceived a virtual opinion leader to represent all of the users in that cluster. Then, instead of processing the original full user-item rating matrix, a reduced virtual opinion leader-item matrix is processed by the proposed Weighted Slope One-VU recommendation algorithm. Experiments on MovieLens datasets show that our scheme can achieve performance (measured by RMSE) comparable with recommendation algorithms based on matrix factorization, but reduce time complexity in common scenarios. Furthermore, a practical movie recommendation system called MovieWatch was developed, deployed, and opened to the public to collect user feedback on the movies recommended to them.

REFERENCES

[1] [1] Francesco Ricci, LiorRokach, BrachaShapira and Paul B. Kantor - Recommender Systems Handbook; First Edition; Springer-Verlag New York, Inc. New York, NY, USA, 2010.

[2] [2] Tariq Mahmood and Francesco Ricci," Improving recommender systems with adaptive conversational strategies", 20th ACM conference on Hypertext and Hypermedia, pp. 73–82, ACM, July 2009.

[3] [3] Tariq Mahmood, Francesco Ricci, Adriano Venturini and Wolfram Höpken, "Adaptive recommender systems for travel planning", Information and Communication Technologies in Tourism 2008, Proceedings of the International Conference,Innsbruck Austria, pp. 1 - 11, 2008.

[4] [4] X. Su and T. Khoshgoftaar, "A survey of collaborative filtering techniques," Advances in Artificial Intelligence, vol. 2009, pp. 19, August 2009. 35

[5] [5] Yongfeng Zhang, Min Zhang and Yiqun Liu, "Incorporating Phrase-level Sentiment Analysis on Textual Reviews for Personalized Recommendation",Eighth ACM International Conference on Web Search and Data Mining, pp. 435 – 440, February 2015.

[6] [6] P. Resnick and H. R. Varian, "Recommender systems," Communications of the ACM, vol. 40, no. 3, pp. 56–58, 1997.

[7] [7] J. Bennett and S. Lanning, "The Netflix Prize", ACM SIGKDD Explorations Newsletter - Special issue on visual analytics, Vol. 9 Issue 2, pp. 51 – 52, December 2007.

[8] [8] Titov and R. McDonald, "A joint model of text and aspect ratings for sentiment summarization", Annual Meeting of the Association for Computational Linguistics, pp. 308 – 316, June 2008.

[9] [9] P. Lops, M. de Gemmis and G. Semeraro, "Content-based recommender systems: State of the art and trends", Recommender Systems Handbook, pp. 73 - 105, 2011.

[10] [10] X. Ding, B. Liu, and P. S. Yu, "A Holistic Lexicon-Based Approach to Opinion Mining", Web Search and Data Mining, pp. 231 - 239, February 2008.