Feature Augmentation based Hybrid Collaborative Filtering using Tree Boosted Ensemble

Udayabalan Balasingam and Gopalan Palaniswamy
Department of Computer Applications, National Institute of Technology, Tiruchirappalli, India
E-mail: udayabalan@yahoo.com, npgopalan@nitt.edu

Keywords: feature augmentation, collaborative filtering, ensemble modelling, boosting, decision tree, supervised learning

Received: January 8, 2018

Requirements for recommendation systems are currently on the raise due to the huge information content available online and the inability of users to manually filter required data. This paper proposes a Feature augmentation based hybrid collaborative filtering using Tree Boosted Ensemble (TBE), for prediction. The proposed TBE recommender is formulated in two phases. The first phase creates category based training matrix using similar user profiles, while the second phase employs the boosted tree based model to predict ratings for the items. A threshold based filtering is finally applied to obtain precise recommendations for the user. Experiments were conducted with MovieLens dataset and performances were measured in terms of Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The proposed model was observed to exhibit MAE levels of 0.64 and RMSE levels of 0.77 with a variation level of ±0.1. Comparisons with state-of-the-art models indicate that the proposed TBE model exhibits reductions in MAE at 6% to 14% and RMSE at ~0.2.

Povzetek: Avtorji so razvili novo metodo za svetovalne sisteme, temelječo na bogatitvi atributov za filtriranje s pomočjo spodbujevalnega ansambla dreves (TBE).

1 Introduction

Information explosion has led to a huge amount of data being generated online. However, human intellect and perception levels are stable, leading to difficulty in processing all the information available to them [1]. This has led to the formulation of prescription based models that provides automatic recommendations to the users. Such automated recommendations help users to a large extent by categorizing the information and providing the most significant information to the users such that they do not miss them. Systems enabling such automated categorizations and filtering are called recommender systems.

Recommender or a recommendation system is a specialized information filtering model that performs predictions based on the preference a user provides to an item. The preference levels are measured using ratings provided by the user to the item or similar items [2]. User ratings are analyzed and items similar to best rated items are recommended for the users. Recommender systems are not sidelined to predicting products alone. Due to the high online usage levels, such systems have become very popular and are currently used to predict books [3], music, news, research articles and even search queries, jokes and restaurants. Some of the current and most popular recommendation systems were music predictions by Last.fm and Pandora radio. Interests in recommendation systems were sparked by the Netflix challenge that offered a prize of $1 Million for improving their model by 10% [4].

Recommendation systems can be designed in three major aspects [5] namely; collaborative filtering, content based filtering and hybrid recommenders. Collaborative filtering models [6] are based on analyzing the user's behaviors and preferences to provide predictions. Major advantages of using such models are that they are based on available and machine analyzable content. This makes their recommendations more accurate and relatable. However, they suffer from issues like data sparsity, data unavailability (cold start) [7] and data volume [8]. Content based recommenders [9] are based on items that model user's profiles. Predictions are based on the created profiles. Hybrid recommenders [10] are a combination of collaborative and content based recommenders.

This paper proposes a feature augmentation based collaborative filtering mechanism to predict items preferred by users. It uses a model based recommendation approach, where prediction is modelled as a regression problem. Recommendations are usually fine-tuned to users. Hence predictions for one user pertain to that user only. The proposed collaborative filtering architecture is modelled in two phases. The first phase deals with identifying the current user's interests and forming their profile, finding users similar to the current profile and identifying the item vectors pertaining to similar users. Most recommendation systems stop at this level to provide recommendations. The proposed approach moves further by building a training matrix from the item vectors that is passed to the next phase. The second phase uses a boosted tree based ensemble to create a prediction model that is used for the final predictions. Experiments and comparisons indicate the high effectiveness of the proposed model as it exhibits a considerable reduction in the Mean Absolute Error (MAE) and the Root-Mean-Square Error (RMSE) in comparison to existing state-of-the-art models.

The remainder of the paper is organized as follows: section 2 presents the literature review, section 3 presents the problem formulation, section 4 presents a detailed description of the proposed ensemble model, section 5 presents the experimental results and section 6 concludes the work.

2 Literature review

This section discusses some of the recent contributions in the domain of recommendation systems.

An artificial neural network based recommendation system that uses content-based modelling for predictions was proposed by Paradarami et al. in [11]. This work performs model based predictions by utilizing user reviews. Model based predictions are hybridized with ANN to perform enhanced predictions. A similar hybridized recommendation model specifically for e-learning environments was proposed by Chen et al. in [12]. Several hybrid versions of recommenders are currently on raise, like user specific hybrid recommender for offline optimization by Dooms et al. [13], an augmented matrix based hybrid system by Wu et al. [14], a latent factor based recommendation system by Zheng et al. [15] and a linear regression based collaborative recommendation model by Ge et al. [30].

Utilizing metaheuristics for recommendations have currently been on the raise due to the increase in data volume. A cuckoo search based collaborative filtering model was proposed by Katarya et al. in [16]. This model uses k-means clustering for user grouping and cuckoo search for the process of prediction. Other metaheuristic or evolutionary algorithm based recommendation systems include memetic algorithm and genetic algorithm based recommender system by Banati et al. [17], PSO based recommender system [18] and fuzzy ant based recommenders by Nadi et al. [19].

A weighting strategy based recommender that performs genre based clustering was proposed by Fremal et al. [20]. This method analyzes twelve weighting strategies in terms of MAE and RMSE to obtain the best weighting model for effective recommendations. A similar multiple clustering based recommendation model was proposed by Ma et al. in [21]. A trust and similarity based recommender for leveraging multiviews was proposed by Guo et al. in [22]. A prediction system to recommend complimentary products was proposed by McAuley et al. in [23]. This model concentrates in identifying substitutable versions of customers' interests. A coordinate based recommendation system SCoR was proposed by Papadakis et al. [32]. This model uses a combination of matrix factorization and collaborative filtering to improve the prediction process. A user perception based model was proposed by Chen et al. [33]. This is a critiquing based model that considers user's perception of products for the prediction process.

Although several models for recommendations are available, most of them follow the regular filtering mechanisms, resulting in huge data for processing, thereby increasing the computational complexity to a large extent.

3 Problem formulation

The collaborative filtering model has been formulated as a prediction problem, where the proposed Tree Based Ensemble (TBE) model predicts the probable ratings that will be provided by the user for a particular item.

Let CL be the set of customers, where $CL=\{C_1,C_2,C_3...C_m\}$ and P_C be the purchase list of a customer, where the item purchased i_x and corresponding rating given to the item r_x are the mandatory components. All available n items are contained in the items list $I=\{i_1,i_2,i_3...i_n\}$. The ratings are formulated as real numbers R in the interval $[r_{min}, r_{max}]$, where r_{min} and r_{max} are defined by the domain.

The problem is to predict a set of items from I for a customer C such that the customer would have a high probability of purchasing it, pertaining to constraints given in eq 1.

$$Prediction_C = \{i_x | i_x \in I \wedge i_x \notin P_C\} \qquad (1)$$

4 Collaborative filtering using tree based boosted ensemble

Collaborative filtering is the process of predicting a user's interests based on their past behaviors. This paper proposes a model based collaborative filtering approach using a boosted ensemble. Algorithm for the proposed collaborative filtering architecture is given below.

Algorithm:

```
1.  Input user (C) for recommendation
2.  Generate item list $I_C$ from the customer purchase history
3.  Generate rating list $IR_C$ using Eq. 3
4.  Identify similar users (NC) by selecting users with common item list using Eq. 4
5.  Generate rating list by adding the item and ratings given by C and NC
6.  Identify correlation between the rating vectors of C and NC using Eq. 6
7.  Filter items with correlation levels higher than the similarity threshold $\rho_{Thresh}$
8.  Identify categories of the selected items
9.  Normalize categories to obtain the training matrix (T)
10. Create the recommender model by applying T to the boosted tree based ensemble
11. Predict ratings for all the available items
12. Filter n best items with highest ratings as recommendations to the user C
```

The proposed collaborative filtering architecture has been modelled in two major phases namely; profile induced item matrix creation using feature augmentation and ensemble based predictions. The first phase collects, filters and integrates data corresponding to the user for whom the recommendation is to be made. The second phase creates a boosted ensemble, trains it using the created training data and provides predictions.

4.1 Profile induced item matrix creation using feature augmentation

Recommendation systems are built for heterogeneous users. Every user's requirements is distinct, however there also exists slight similarities with other users in the system [24]. Hence it is important to identify and build appropriate profile for the current user for the model to be trained upon. Effectiveness of predictions depends entirely on the quality of the training data built at this phase. The process of building the user's profile is performed using Feature Augmentation. Feature Augmentation is the process of computing a set of features to be passed to the subsequent phase for evaluation.

The initial phase of this process is to generate an item list from the purchase history of the customer under analysis. This is given by

$$I_C = \{i_x | i_x \in I \wedge i_x \subset P_C\} \quad (2)$$

Where i_x is an item from the set of all items purchased by the customer.

The ratings pertaining to the item list I_C are integrated to obtain the user's preferences.

$$IR_C = \{(i_x, r_x) | (i_x, r_x) \subset P_C\} \quad (3)$$

Where r_x is the rating corresponding to item i_x.

The mere factor that the customer has purchased an item does not guarantee the person's affinity towards the product. Hence affinity levels for the product are obtained by integrating the ratings.

The next step is to identify users with interests similar to the current user *C*. Identifying similarities begins by identifying the commonalities existing between *C*, the current user, paired with all the other existing users (NC). This is given by

$$I_{Common} = I_C \cap I_{NC} \quad \forall \quad 1 \le NC \le m \ \wedge \ NC \ne C \quad (4)$$

Where I_C and I_{NC} correspond to items purchased by customers *C* and *NC*.

The common items identified in I_{Common} are integrated with their corresponding ratings from *C* and *NC*, to create the ratings matrix, which is given by,

$$IR_{Common} = \{(i_x, r_c, r_{nc}) | (i_x, r_c) \subset P_C \ \wedge \ (i_x, r_{nc}) \subset P_{NC}\} \quad (5)$$

Where r_c is the rating given to product i_x by customer *C* and r_{nc} is the rating given to product i_x by customer *NC*.

Correlation of ratings r_c and r_{nc} between the current customer *C* and every other customer *NC* is determined to identify the similarity levels between the two customers. Similarities are identified between two rating vectors, and a similarity identification model is used for the process [25]. Some of the common similarity measures are Euclidean distance, Minkowski distance and Pearson correlation [2]. Distance based measures requires the input vectors to be standardized prior to operations, while major advantage of a correlation based model is that they operate based on cosine similarities, hence do not require standardized values. This avoids the additional overhead of standardizing the input data. Hence the proposed TBE model uses Pearson correlation as the similarity measure identifier. TBE model uses all the items identified as common (entire population) to obtain the similarity, which is given by

$$\rho_{C,NC} = \frac{cov(R_C, R_{NC})}{\sigma_{R_C} \sigma_{R_{NC}}} \quad (6)$$

Where R_C and R_{NC} are the rating vectors corresponding to *C* and *NC* , the numerator calculates the covariance of R_C and R_{NC}, the denominator calculates the product of standard deviations of R_C and $R_{NC.}$

The final item set is obtained by filtering item data satisfying the similarity threshold (ρ_{Thresh}). The similarity threshold is domain and data dependent. The proposed TBE model sets a similarity threshold of 0.5 for analysis. Items corresponding to users with satisfied thresholds are considered for building the training matrix. The selection criteria for items is given by

$$I_{Selected} = \{i_x | i_x \in IR_{Common} \wedge i_x \in I_{NC} \wedge \rho_{C,NC} < \rho_{Thresh}\} \quad (7)$$

Item categorization plays a vital role in determining the details pertaining to items. Training data for TBE is constructed with the item categorization details, rather than the actual items. Broader and highly specific categorizations tend to provide more accurate results. The proposed model also deals with integrating items falling under multiple categories. Categories pertaining to items under $I_{Selected}$ are obtained. Item categorizations tend to be nominal rather than numeric. Hence they are normalized with 1-of-n encoding to obtain the numeric training data matrix (*T*) for training the ensemble model.

4.2 Ensemble based Predictions

Model based recommenders utilize a machine learning model to predict recommendations for a user. The proposed TBE model builds a boosted tree ensemble for prediction.

Ensemble modelling [26] is the process of incorporating multiple models for prediction, rather than relying on the results of a single model. Boosting is a machine learning ensemble aimed to reduce bias and variance in the prediction system to provide an effective prediction model. It is a supervised learning approach operating by creating a set of weak learners to form a single strong learner. This work uses decision trees to build the model for recommendations based on the training data [27].

The proposed boosting model operates by iteratively training the algorithm based on the resultant errors from previous iterations.

Let *DT(x)* be the base decision tree used for training. The process of prediction is given by

$$y' = DT(x) \quad (8)$$

Where *y'* is the prediction given by the decision tree model *DT*. However, being a weak learner, the predictions by *DT* will constitute errors *e*, which can be given by

$$e = y' - y \quad (9)$$

Where *y* is the actual solution and *y'* is the predicted solution.

The next level prediction model is built by integrating the error component *e* into the prediction model. This is given by

$$y'' = DT(x) + e \qquad (10)$$

Similarly, the next level error is given by

$$e' = y - y'' \quad (11)$$

The next model training incorporates *e'* into the training process. This is iteratively performed until the error *e* reaches an acceptable threshold.

Training data for the recommendation problem is modelled with category based training matrix constructed from item ratings obtained from similar users. Rating corresponding to the item vector is incorporated as the class label for the training matrix (*T*). The training matrix is passed to the boosted decision tree and the trained model is obtained. The process of training matrix creation and prediction is repeated for each user individually on every recommendation requirement, to obtain result pertaining to an individual user.

Test data is obtained by considering the items not contained in the purchase list of customer *C*. The formulation of test data is given by

$$I_{Test} = \{i_x | i_x \in I \land i_x \notin P_C\} \quad (12)$$

Categories corresponding to I_{Test} are obtained and 1-of-n encoding is applied to obtain the test matrix. TBE is formulated as a regression model. Hence an error level of 0.001 is set as the acceptable error limit for the TBE model.

The results provide probable user ratings for each item. Recommendations can be provided by sorting the results in decreasing order and providing the top *n* rated products as probable recommendations.

5 Experimental results

The proposed TBE model is implemented using Python and uses MovieLens data [28, 29] for analysis. MovieLens is a benchmark dataset used to validate recommendation systems. The dataset pertaining to 1Million reviews is considered for evaluation. It contains details pertaining to 6040 users and provides reviews for 3952 movies. Ratings are provided on a 5 point scale. The dataset also contains categorizations of movies in terms of genres. A single movie sometimes belongs to multiple genres, providing scope for multiple options. The proposed model operates by considering movies as items and genres as the categorization parameters.

Recommendation models are usually measured in terms of Root-Mean-Square Error (RMSE) and Mean Absolute Error [13, 31].

$$MAE = \frac{1}{N}\sum_{i=1}^{n} |y_i - y'_i| \qquad (13)$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(y_i - y'_i)^2} \quad (14)$$

Where y_i and y_i' are the actual and the predicted ratings for the *N* test reviews. MAE measures the effectiveness of the predictions. Smaller MAE values exhibit better predictions. RMSE depicts the stability of the predictions, in other words, the prediction variance. Low MAE values represent a good predictor, while high RMSE values indicate high variability in predictions.

Performance of the proposed model is measured in terms of RMSE and MAE. Scalability of the model is measured by sampling the data from 100K reviews, moving up to 1 Million reviews. Exhibited results were attained by performing 1000 iterations and identifying the mean of the obtained predictions.

Mean Absolute Error (MAE) corresponding to datasets of various sizes is shown in figure 1. It could be observed that the proposed TBE model exhibits similar MSE values exhibiting low fluctuations irrespective of the data size. The best MAE was observed to be 0.51, and average MAE was observed to be 0.64, with fluctuation levels of ±0.1. This exhibits the stability of the proposed model irrespective of the data size.

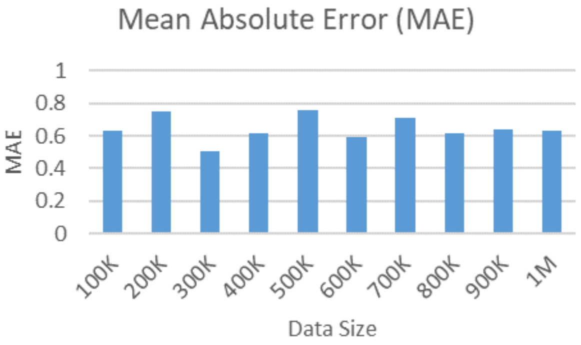

Figure 1: Mean Absolute Error Analysis for Varied Sized Input Data.

Root-Mean-Square Error (RMSE) corresponding to datasets of various sizes is shown in figure 2. It could be observed that the proposed TBE model exhibits similar RMSE values exhibiting low fluctuations irrespective of

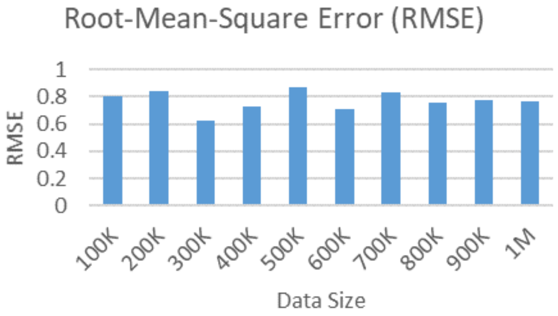

Figure 2: Root-Mean-Square Error Analysis for Varied Sized Input Data.

the data size. The best RMSE was observed to be 0.62, and average RMSE was observed to be 0.77, with fluctuation levels of ±0.1. This exhibits the low variability in prediction levels of TBE.

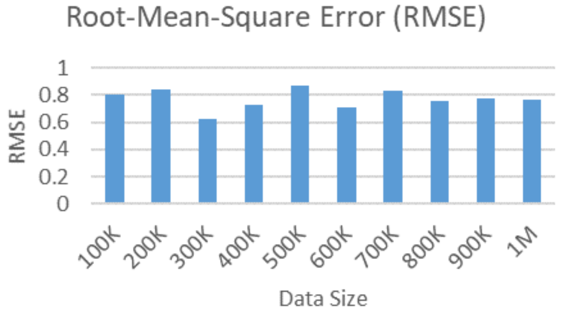

Figure 2: Root-Mean-Square Error Analysis for Varied Sized Input Data.

Enhanced performance of the proposed model is attributed to the two major factors, feature augmentation and the tree based boosted ensemble. Feature augmentation is based on the user's profile. Hence the input data contains several attributes depicting the user's profile. This results in the model being highly fine-tuned towards the user's requirements. This enables better predictions. further, usage of the boosting model ensures that every wrong prediction increases the weight of the instance. This enables even rarely found instances to have a significant impact on the final prediction process. These factors enable enhanced results in the proposed model.

The actual performance values for TBE is tabulated and shown in table 1. Best performances are shown in bold. Moderate MAE and RMSE values indicate effectively reduced error levels and low variability levels in predictions. Time requirements for the proposed model exhibits linear time requirements by the TBE model.

Dataset	MAE	RMSE	Time(s)
100K	0.630135	0.806763	9.197
200K	0.746131	0.841447	20.662
300K	**0.506511**	**0.625448**	51.306
400K	0.617964	0.724338	88.696
500K	0.757933	0.871471	94.607
600K	0.590549	0.705942	112.125
700K	0.710233	0.835888	105.766
800K	0.618823	0.753463	108.144
900K	0.641097	0.77633	113.657
1M	0.629914	0.763647	118.935

Table 1: Performance Levels on Data with Varied Sizes.

Comparison of the proposed model is performed with the weighted strategy based model (SW I, MLR and CM II) proposed by Fremal et al. [20] and K-Means and Cuckoo Search based model proposed by Katarya et al. [16]. Both these models are recent and also considers the MovieLens data for their prediction process. Hence this work considers the two models for comparison.

A comparison on MAE values of the proposed model with SW I, MLR, CM II and K-Means Cuckoo is shown in figure 3. It could be observed that the proposed TBE model exhibits lowest MAE levels, exhibiting reduction levels in the range of 6% to 14%.

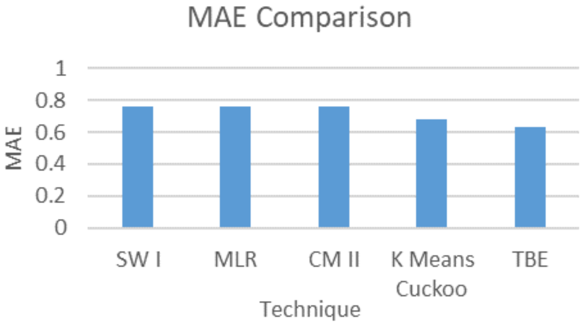

Figure 3: Mean Absolute Error Comparison.

A comparison on RMSE values of the proposed model with SW I, MLR, CM II and K-Means Cuckoo is shown in figure 4. It could be observed that the proposed TBE model exhibits lowest RMSE levels (0.76), while the other models exhibits RMSE levels >0.9. This exhibits that the variance levels of the proposed model is low compared to the other models. Low variability levels indicate high prediction reliability of the proposed TBE model.

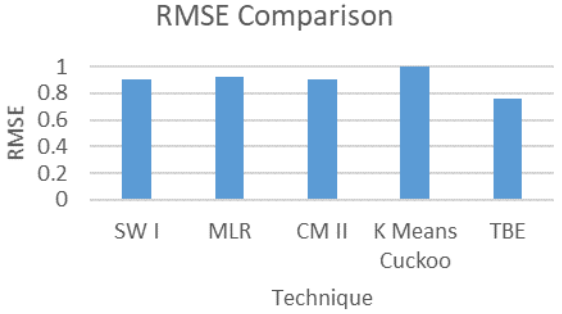

Figure 4: Root-Mean-Square Error Comparison.

6 Conclusion

Recommendations have become one of the major requirements in the current information rich world. However, the voluminous data available for the recommendation engines poses a huge challenge. This paper proposes a feature augmentation based tree bagging ensemble model, TBE for recommendations. TBE, being an iterative model uses a weak classifier, hence the computational complexities pertaining to TBE was observed to be very low. Further, the repeated data filtering process in the first phase reduces data to a large extent. This further reduces the computational complexities, hence speeding up the prediction process to a considerable extent. This aids in handling large datasets effectively, indicating enhanced scalability levels.

The major advantage of TBE is that it uses the available data for the current user and user's similar to the current user. Hence TBE has the capability to identify even hidden patterns from the available data. Further, this process of prediction solves the data sparsity issue that affects collaborative filtering approaches. Limitations of the proposed model are that cold start problem has not been handled. Future extensions of the proposed model will be designed by incorporating user's demographic data, which can enable solving the cold start issue.

References

[1] Vickery, A., and Vickery, B. C. (2005). *Information science in theory and practice*. De Gruyter. https://doi.org/10.1515/9783598440083

[2] Ricci, F., Rokach, L., and Shapira, B. (2011). *Introduction to recommender systems handbook* Boston, MA: Springer US . pp. 1-5. https://doi.org/10.1007/978-0-387-85820-3_1

[3] Karlgren, Jussi. "A digital bookshelf: original work on recommender systems"

[4] Lohr, S. (2009). A $1 million research bargain for Netflix, and maybe a model for others. *The New York Times*, *22*.

[5] Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering, 17* (6), 734–749. https://doi.org/10.1109/TKDE.2005.99

[6] Prem Melville and Vikas Sindhwani, (2010) Recommender Systems, *Encyclopedia of Machine Learning,* pp. 1-9.

[7] Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 253-260. https://doi.org/10.1145/564376.564421

[8] Sarwar, B. M., Karypis, G., Konstan, J., & Riedl, J. (2002). Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In *Proceedings of the fifth international conference on computer and information technology*, Vol. 1.

[9] Mooney, R.J., and Roy, L. (1999). Content-based book recommendation using learning for text categorization. *In Workshop Recom. Sys.: Algo. and Evaluation.* https://doi.org/10.1145/336597.336662

[10] Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). *Recommender systems: an introduction*. Cambridge University Press. https://doi.org/10.1017/CBO9780511763113

[11] Paradarami, T.K., Bastian, N.D. and Wightman, J.L., (2017). A hybrid recommender system using artificial neural networks. *Expert Systems with Applications*, vol. 83, pp.300-313. https://doi.org/10.1016/j.eswa.2017.04.046

[12] Chen, W., Niu, Z., Zhao, X., and Li, Y. (2014). A hybrid recommendation algorithm adapted in e-learning environments. *World Wide Web,* doi: 10. 1007/s11280-012-0187-z .vol. 17(2), pp.271-284. https://doi.org/10.1007/s11280-012-0187-z

[13] Dooms, S., Pessemier, T., and Martens, L. (2015). Offline optimization for user-specific hybrid recommender systems. *Multimedia Tools and Applications,* vol. 74(9) , pp.3053-3076. https://doi.org/10.1007/s11042-013-1768-2

[14] Wu, M.L., Chang, C.H., and Liu, R.Z. (2014). Integrating content-based filtering with collaborative filtering using co-clustering with augmented matrices. *Expert Systems with Applications*, vol. 41(6),pp.2754-2761. https://doi.org/10.1016/j.eswa.2013.10.008

[15] Zheng, Y. (2014). Semi-supervised context-aware matrix factorization: Using contexts in a way of "latent" factors. *In Proceedings of the 29th annual ACM symposium on applied computing* . In SAC New York, NY, USA: ACM .vol.14, pp.292-293. https://doi.org/10.1145/2554850.2555195

[16] Katarya, R. and Verma, O.P., (2016). An effective collaborative movie recommender system with cuckoo search. *Egyptian Informatics Journal*. https://doi.org/10.1016/j.eij.2016.10.002

[17] Banati, H. and Mehta, S., (2010). A Multi-Perspective Evaluation of ma and ga for collaborative Filtering Recommender System. *International journal of computer science & information Technology (IJCSIT)*, vol. *2*(5), pp.103-122. https://doi.org/10.5121/ijcsit.2010.2508

[18] Alam, S., Dobbie, G. and Riddle, P., (2011), May. Towards recommender system using particle swarm optimization based web usage clustering. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* . Springer, Berlin, Heidelberg.pp.316-326. https://doi.org/10.1007/978-3-642-28320-8_27

[19] Nadi, S., Saraee, M., Bagheri, A. and Davarpanh Jazi, M., (2011). FARS: Fuzzy ant based recommender system for web users. *International Journal of Computer Science Issues*, vol.8 (1), pp.203-209. https://doi.org/10.20533/ijmip.2042 .4647.2011.0001

[20] Frémal, S. and Lecron, F., (2017). Weighting strategies for a recommender system using item clustering based on genres. *Expert Systems with Applications*, vol.77, pp.105-113. https://doi.org/10.1016/j.eswa.2017.01.031

[21] Ma, X., Lu, H., Gan, Z., and Zhao, Q. (2016). An exploration of improving prediction accuracy by constructing a multi-type clustering based recommendation framework. *Neurocomputing,* vol.191, pp. 388–397. https://doi.org/10.1016/j.neucom.2016.01.040

[22] Guo, G., Zhang, J., and Yorke-Smith, N. (2015). Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems. *Knowledge-Based Systems,* vol. 74 , pp.14–27.

https://doi.org/10.1016/j.knosys.2014.10.016

[23] McAuley, J., Pandey, R., and Leskovec, J. (2015).a Inferring networks of substitutable and

complementary products. *In Proceedings of the twenty first ACM SIGKDD international conference on knowledge discovery and data mining* . In KDD15 New York, NY, USA: ACM. pp.785-794. https://doi.org/10.1145/2783258.2783381

[24] Kunaver, M., & Požrl, T. (2017). Diversity in recommender systems–A survey. *Knowledge-Based Systems*, Vol. 123, pp. 154-162. https://doi.org/10.1016/j.knosys.2017.02.009

[25] Sorensen, S. (2012). Accuracy of similarity measures in recommender systems. In Proceedings of the 29th SemiAnnual Computer Science Senior Seminar Conference (Minnesota, USA.

[26] Rokach, L. (2010). "Ensemble-based classifiers". *Artificial Intelligence Review*. Vol. **33**, pp. 1–39. https://doi.org/10.1007/s10462-009-9124-7

[27] Breiman, L. (1997) "Arcing The Edge"

[28] https://grouplens.org/datasets/movielens/

[29] Harper, F.M., and Konstan, J.A. (2015). The MovieLens datasets: History and context. ACM *Transactions on Interactive Intelligent Systems*, doi: 10.1145/2827872 .vol. 5(4), pp.19:1-19:19. https://doi.org/10.1145/2827872

[30] Ge, X., Liu, J., Qi, Q., & Chen, Z. (2011, July). A new prediction approach based on linear regression for collaborative filtering. In *2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)* (Vol. 4, pp. 2586-2590). IEEE. https://doi.org/10.1109/FSKD.2011.6020007

[31] Ge, X., Liu, J., Qi, Q., & Chen, Z. (2011). A new prediction approach based on linear regression for collaborative filtering. IEEE *Eighth International Conference* In *Fuzzy Systems and Knowledge Discovery (FSKD)*, Vol. 4, pp. 2586-2590. https://doi.org/10.1109/FSKD.2011.6020007

[32] Papadakis, H., Panagiotakis, C., & Fragopoulou, P. (2017). SCoR: a synthetic coordinate based recommender system. *Expert Systems with Applications*, *79*, 8-19. https://doi.org/10.1016/j.eswa.2017.02.025

[33] Chen, L., Yan, D., & Wang, F. (2019). User perception of sentiment-integrated critiquing in recommender systems. *International Journal of Human-Computer Studies*, *121*, 4-20. https://doi.org/10.1016/j.ijhcs.2017.09.005