Movie Recommender System using Improvised Cuckoo Search

Puspanjali Mohapatra, Ritesh Kumar Mohapatra, Bibhuranjan Sandhibigraha

***Abstract*: Recommender system is a tool for information filtering that predicts the rating for users and items, on the basis of their likings. Movie recommendation systems provides a mechanism to classify users with similar interests. This makes it an integral part of websites and e-commerce applications. In this research article, a new recommender system has been proposed which makes use of k-means clustering by adopting cuckoo search optimization algorithm applied on the Movielens dataset. Our approach has been explained systematically, and the subsequent results have been discussed. It is also compared with the existing approaches, and the results have been analysed and interpreted.**

***Index Terms*: Recommender System, Movie, Cuckoo Search, K-mean Clustering**

I. INTRODUCTION

Stepping into the era of internet revolution and big data, the internet is show cashed with humongous data of customer's interests. In order to avail all the information efficiently and successfully, the customers are depending more and more on recommender systems. A recommender system is an application which is used for predicting a user's interest on an item based on information about the items, users and their relationship between them. It accumulates information about the preferences of a user on distinct items by following two ways, implicitly or else explicitly [1]. The process of filtering items using opinions of many distinct people is known as Collaborative filtering [2]. The process involves collection of the movie ratings given by each individual user. Then the target user gets the recommended movies based on people who are very much alike with similar interest and taste in the past. A popular technique for segregating users of similar type in recommender system is clustering. It is a unsupervised technique used on a given data for distributing it into equivalent clusters on the basis of some similarity or difference metrics [3]. For improving the movie prediction accuracy, a hybrid clustering and optimization approach is proposed. Such approach overwhelms the limitations of stereotypical collaborative or content-based systems. K-means clustering algorithm is used for clustering and the cuckoo search optimization technique is performed for optimization. As k-means clustering method performs in comparison to other methods of clustering, it is measured on the basis of parameters like computational complexity or time for execution also varying with different number of clusters [4]. When compared with other bio-inspired algorithms, it has been shown that cuckoo search yields better performance. Comparisons of several existing algorithms were made with cuckoo search, it gave good results and found appropriate weights [5-6]. From the experimental results, it depicts that the suggested approach has the ability of generating more accurate movie recommendations in comparison to the available clustering based collaborative filtering methods.

II. RELATED WORK

Many recommender systems are being developed using many diversified approaches which includes collaborative based approach [7], content-based approach [8] and hybrid approach [9]. Many of them are based on collaborative filtering and clustering. The ratings given by the user to rate the movies which are seen by them, the movie recommender system helps to recommend other movies which are not discovered by them by employing collaborative filtering. This technique became so much popular that it has influenced many recommender systems. It is predominantly divided into two types that is memory based and model based. In memory based collaborative filtering, it involves traversal for closest neighbour within the user set for an active user and effectively produces movie recommendations. In this method the flaws are information sparsity and computational complexity. In order to overcome these issues, many researchers tried to introduce relationship between items and users which lead to development of item based collaborative filtering [10]. Result analysis showed that methods involving item could lessen the computational time. Also, it produced reasonable correct prediction and precision. In model based collaborative filtering, [11] it develops a model to recognize the rating patterns collected from the database of ratings given by users that helps to overcomes the limitations like data sparsity and complexity. To solve issues like scalability, many researchers have used techniques of clustering to enhance movie recommendation systems.

III. PROPOSED FRAMEWORK

Several limitations of the collaborative filtering methods are discussed in section II.. To overcome these issues, we propose a hybrid clustering and optimization approach for improving the precision in prediction. We apply K-means as clustering algorithm to the Movie Lens dataset for grouping the users into different clusters. Initially the clusters are randomly chosen, then the users present in it are examined by calculating the differences between user's rating and centroid of the clusters. The user gets allocated to a

Manuscript Received on May 10, 2019.

Prof. Puspanjali Mohapatra, Department of computer science and Engineering, International Institute of Information Technology (IIIT), Bhubaneswar (Odisha), India.

Ritesh Kumar Mohapatra, Department of computer science and Engineering, International Institute of Information Technology (IIIT), Bhubaneswar (Odisha), India.

Bibhuranjan Sandhibigraha, Department of computer science and Engineering, International Institute of Information Technology (IIIT), Bhubaneswar (Odisha), India.

*Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication*

particular cluster, if their difference is smallest. But at this moment, we cannot verify that every user is allocated to a cluster which has the closest centroid. Therefore, the distance from each user is compared to each cluster's centroid and then according to least distances relocation happens. This process will continue until no further relocations happens. If no relocations happen after a point, then this would be the completion of clustering.

After K-means clustering algorithm is executed, then cuckoo search optimization is applied for optimizing the results. A fitness function is developed for the clusters which enhances each user to centroid of cluster distances. This function changes the existing centroids to new ones for a no of iterations. Then the users are classified by calculating the least distance to any centroid. Cuckoo Search algorithm is explained by the subsequent rules [12]:

1. Each cuckoo puts one egg at any given moment and dumps its egg in arbitrarily picked nest.
2. The best nests with the top quality of eggs are carried over to the next generations.
3. Various existing host nest is fixed, and the egg laid by a cuckoo is uncovered by the hosts with a probability p [0, 1]

In this framework, we have taken into account an association in which a user is referred to as an egg and the nest as a cluster. Fig 1 is a flowchart which exhibits the step-wise flow of the process involved. The complete flow of proposed algorithm can be visualized in following three phases.

1. Dividing the data into a predefined fixed number of clusters using conventional K-Means algorithm
2. Optimizing these clusters using proposed modified Cuckoo Search algorithm
3. Measuring different performance parameters to compare with other approaches

The detailed steps of first phase i.e. dividing the data using K-means clustering is given below

1. Select K users (points) into the space that are to be clustered. These points constitute the opening set of centroids.
2. Assign each of the remaining user to the cluster which has the nearest centroid.
3. After all users are assigned, compute the locations of centroids for each cluster.
4. Repeat steps two and three until the centroids gets fixed. This segregates users into a group (clusters) from which the measurement to be decreased can be determined.

After these steps we have fixed the number of clusters. Let the leftover items are in a pool. The following steps are executed to optimize the cluster.

1. A centroid for each cluster is determined. Afterwards distances from each user to its centroid is calculated and user with maximum distance is marked as 'worst user'.
2. Select a random cluster and a random user from pool
3. Calculate the distance of the user to the centroid of the selected cluster using Euclidean distance method
4. If this distance is less than the distance of worst user than remove that 'worst user' from respective cluster and add current user to that cluster otherwise simply add current user to the cluster
5. Repeat the above process until the pool is empty

After these processes all the users are placed in particular clusters and ready to be processed further. Then, the ratings which the user will give are predicted.

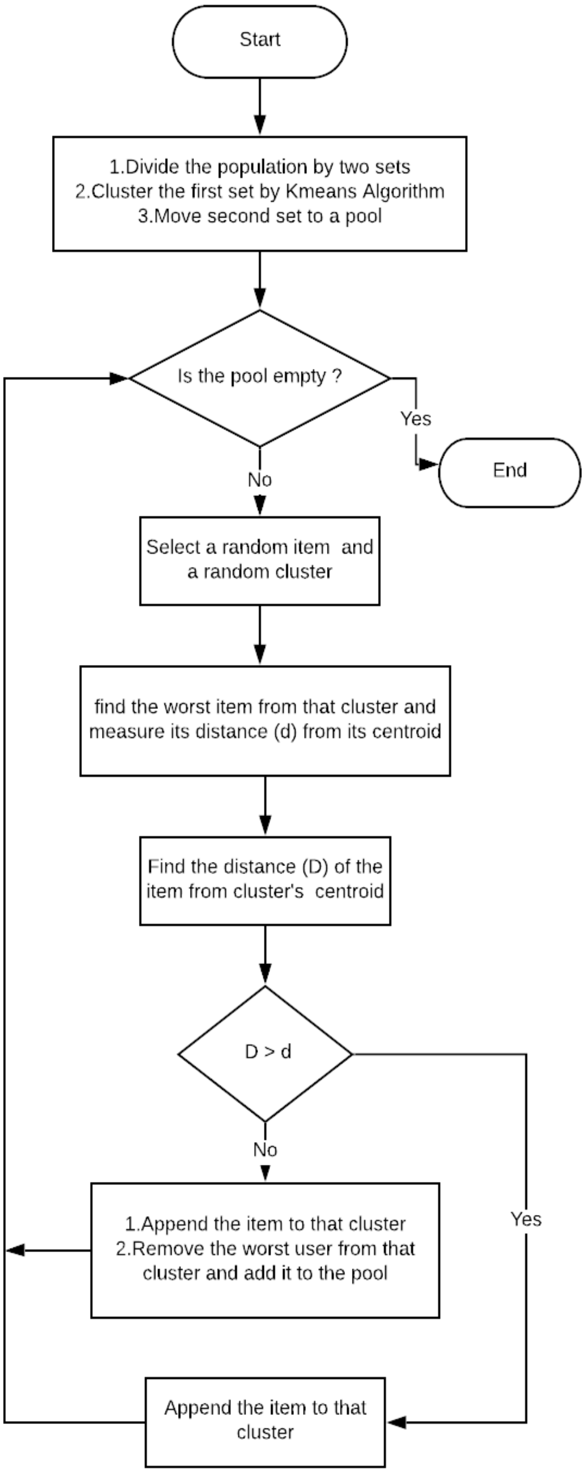
Start
1.Divide the population by two sets
2.Cluster the first set by Kmeans Algorithm
3.Move second set to a pool
Is the pool empty ?
Yes
End
No
Select a random item and a random cluster
find the worst item from that cluster and measure its distance (d) from its centroid
Find the distance (D) of the item from cluster's centroid
D > d
No
Yes
1.Append the item to that cluster
2.Remove the worst user from that cluster and add it to the pool
Append the item to that cluster

Fig. 1: Complete Process of Proposed Algorithm

IV. RESULTS AND ANALYSIS

For our experiment we have chosen the 100k Movielens data-set

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

(https://grouplens.org/datasets/movielens/100k/) which is available publicly. It contains 100,000 ratings provided by 943 users to 1642 movies ranging from 1 to 5. If a user has not rated the movie, it is left blank.

The data is extracted with the help of python modules. The empty fields are filled with zeros in the phase of preprocessing. The refined data is converted to a matrix with rows represented as users, columns represented as movies, and fields represented as ratings given by users to particular movies. To check the performance of Recommendation system we have calculated various performance measures such as MAE, root means squared error (RSME), and SD under different number of clusters on same dataset. All results are tabulated, various graphs are plotted for better understanding of different parameters and to explore the relationship between them under different conditions.

A. Mean Absolute errors (MAE)

After step by step simulation of the algorithm MAE for Movielens dataset is calculated as

$$MAE = \frac{\sum |R_{ij} - r_{ij}|}{M}$$

where M is the total number of expected movies, r_{ij} is the original rating and R_{ij} is the predicted value by a user i on item j.

For different number of clusters Mean absolute error is calculated and result is shown in Table 1. From the pattern of the result we have concluded the following conclusion.

1. cluster numbers have been changed in a range of 3 to 63 in an interval of 4 and MAE drops from 0.7756 to 0.6457.
2. Increasing the number of clusters, the closeness or similarity increases between the elements which tends to go into the cluster.
3. The increased number of clusters makes not only the calculation and prediction easier but also increases accuracy.
4. As the total number of users is constant, but no. of clusters is increased step wise . Therefore, every user has more flexibility to get located to any one of the clusters. It may happen that some of the clusters will have few number of users.
5. Hence, the deviation between computed value and original value decreases with the increase in no of clusters

The experimental result is represented in graphical form in fig. 2 which displays the variation of MAE w.r.t the no. of clusters.

Table 1
MAE VALUES FOR DIFFERENT NO. OF CLUSTERS

S. No	No of clusters	Mean absolute error
1	3	0.7805
2	7	0.7542
3	11	0.7275
4	15	0.7074
5	19	0.6877
6	23	0.6707
7	27	0.6505
8	31	0.6342
9	35	0.6159
10	39	0.6019
11	43	0.5830
12	47	0.5789
13	51	0.5643
14	55	0.5390
15	59	0.5372
16	63	0.5381
17	67	0.5391

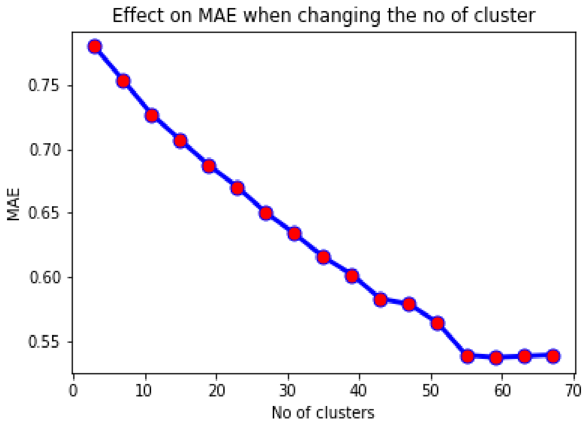

Fig. 2: Effect of MAE with changing no of clusters

B. Standard Deviation (SD)

The SD is calculated for 100k Movie-lens dataset. The mathematical formula for SD is given as follows.

Standard deviation =

$$\frac{\sum_{i=1}^{k}\{\sum_{j=1}^{no\ of\ items\ in\ i}(\sum_{l=1}^{m}\sqrt{\frac{(expect\ l - mean\ K_n l^2)}{m}})\}}{no\ of\ items\ in\ i}$$

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Table 2
SD VALUES FOR DIFFERENT NO. OF CLUSTERS

S. No	No of clusters	Standard Deviation
1	3	0.2265
2	7	0.1993
3	11	0.1788
4	15	0.1742
5	19	0.1698
6	23	0.1658
7	27	0.1599
8	31	0.1336
9	35	0.1308
10	39	0.1303
11	43	0.1354
12	47	0.1199
13	51	0.1223

Like MAE when cluster numbers are increased standard deviation decreased as mentioned in Table 2. When cluster numbers (k) = 3 SD is found to be 0.2265 which deliberately comes down to 0.1223 when we gradually increase the k value to 51 at an interval of 4. From all observations it is clear that optimal results are obtained when we increase our no of cluster from 27 to 31 (applied only for 100k movielens dataset). As mentioned earlier users can be allocated more accurately due to increased number of choices with increased number of clusters and less number of users per clusters as total users are fixed. This is reflected in the results shown in table 2. For better visualization this result is represented in tabular and graphical form in figure 3.

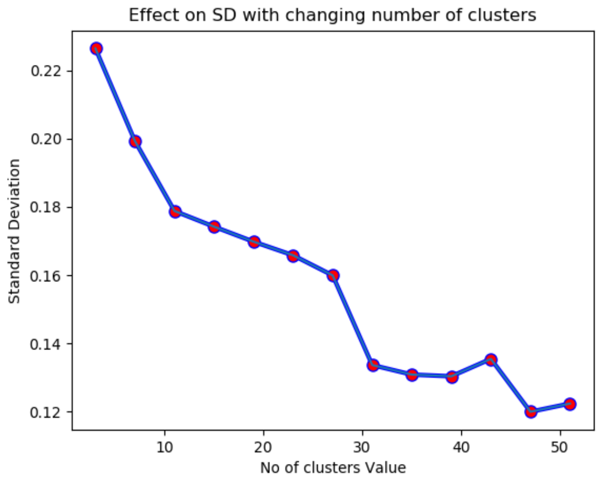

Fig. 2: Effect of SD with changing no of clusters

C. Root Mean Square Error (RSME)

The Root Mean Square Error is a parameter that calculates the deviation between values estimated by the model and the original values ascertained from the environment.

In Table 3, it can be seen that like MAE and SD, the value of RMSE also decreases gradually when total cluster numbers are increased. The increase in the number of clusters results in increasing more similar users for establishing this typical behaviour. The mathematical formula for RMSE is given by

$$\sqrt{\sum (\text{predicted rating} - \text{actual rating})^2/n}$$

The experimental results are tabulated in table 3 and represented graphically in fig. 4.

Table 3
RSME VALUES FOR DIFFERENT NO. OF CLUSTERS

S. No	No of clusters	RMSE
1	3	0.9801
2	7	0.9553
3	11	0.9288
4	15	0.9107
5	19	0.8955
6	23	0.8767
7	27	0.8592
8	31	0.8460
9	35	0.8299
10	39	0.8175
11	43	0.8012
12	47	0.7879
13	51	0.7764
14	55	0.7601
15	59	0.7589
16	63	0.7612
17	67	0.7627

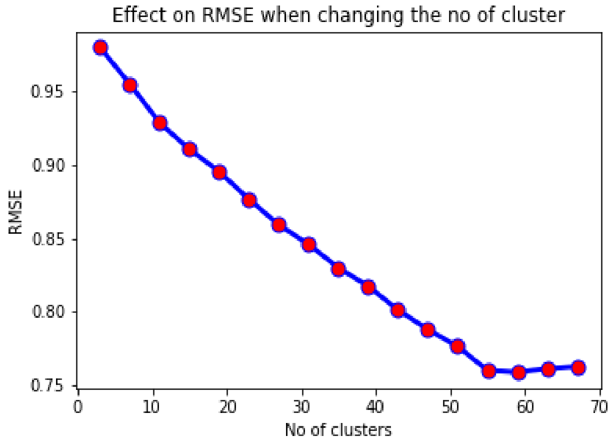

Fig. 3: Effect of RMSE with changing no of clusters

V. CONCLUSION AND FUTURE SCOPE

We proposed a hybrid approach of k-means clustering and cuckoo search optimization algorithm that has the ability to attain an improved movie recommendation system when applied to movie lens dataset. The performance was analyzed using metrics like MAE, RMSE and SD. From, the result analysis, we can infer that the approach proposed has higher accuracy and it has the ability of generating accurate movie recommendations. The only limitation to this approach is that if the initial partition of the clustering is not a reliable one, then at that point efficiency may decrease. As far as future work is concerned, we

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

can use the demographic features and sentiments may be taken into account including various other optimization algorithms.

REFERENCES

1. J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, “Recommender system application developments: a survey,” Decision Support Systems, vol. 74, pp. 12–32, 2015.
2. Y. Shi, M. Larson, and A. Hanjalic, “Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges,” ACM Computing Surveys (CSUR), vol. 47, no. 1, p. 3, 2014.
3. A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, and A. Bouras, “A survey of clustering algorithms for big data: Taxonomy and empirical analysis,” IEEE transactions on emerging topics in computing, vol. 2, no. 3, pp. 267–279, 2014.
4. T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, “An efficient k-means clustering algorithm: Analysis and implementation,” IEEE Transactions on Pattern Analysis & Machine Intelligence, no. 7, pp. 881–892, 2002.
5. A. H. Gandomi, X.-S. Yang, and A. H. Alavi, “Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems,” Engineering with computers, vol. 29, no. 1, pp. 17–35, 2013.
6. M. Hatami and S. Pashazadeh, “Improving results and performance of collaborative filtering-based recommender systems using cuckoo optimization algorithm,” International Journal of Computer Applications, vol. 88, no. 16, 2014
7. Z. Huang, D. Zeng, and H. Chen, “A comparison of collaborativefiltering recommendation algorithms for e-commerce,” IEEE Intelligent Systems, vol. 22, no. 5, pp. 68–78, 2007.
8. R. C. Bagher, H. Hassanpour, and H. Mashayekhi, “User trends modeling for a content-based recommender system,” Expert Systems with Applications, vol. 87, pp. 209–219, 2017.
9. P. Melville and V. Sindhwani, “Recommender systems,” Encyclopedia of Machine Learning and Data Mining, pp. 1056–1066, 2017.
10. J. Basilico and T. Hofmann, “Unifying collaborative and content-based filtering,” in Proceedings of the twenty-first international conference on Machine learning. ACM ,2004, pg 9
11. C. C. Aggarwal, “Model-based collaborative filtering,” in Recommender systems. Springer, 2016, pp. 71–138.
12. Yang, Xin-She, and Suash Deb. "Multiobjective cuckoo search for design optimization." *Computers & Operations Research* 40.6 (2013): 1616-1624

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication