International Journal of Research in Advent Technology, Vol.6, No.7, July 2018
E-ISSN: 2321-9637
Available online at www.ijrat.org

A Recommender System Based on Improvised K- Means Clustering Algorithm

Shivani Sharma
Department of Computer Science and Applications, Kurukshetra University, Kurukshetra
Shivanigaur83@yahoo.com

Abstract: Recommender Systems (RS) have the capability to filter information for a particular user to predict whether the user will like the given item for making a choice. From many years, this process has been dependent on data mining and machine learning techniques which have scalability issues when applied to recommendations. In this paper, an improvised K-Means clustering algorithm for RS is used to address the problems of existing K-Means clustering algorithm. The problem with existing K-Means clustering algorithm is that it chooses initial centroid randomly, which leads to inaccurate recommendations and also increases the execution time. The work of this paper shows how the change in selection of centroids improves the quality of recommendations and also decreases the execution time. The improvised approach has been authenticated with extensive set of experiments based on MovieLens dataset. These experiments proved that the improvised approach of RS provides better quality clusters, less execution time than existing algorithm and improves the accuracy of recommendations.

Keywords: Recommender System, Collaborative Filtering, Clustering, RMSE, Execution Time.

1. INTRODUCTION

In recent years, the amount of information existing on the Internet has increased tremendously. Users using the Internet experience difficulty in finding the items of their interest from a huge search space, this is referred as information overload problem. RS solves information overload problem by personalizing the items according to user's interest. Thus RSs are the tools that filter the items from large space of information and provides the items relevant to the user i.e. recommendations. RSs are used in different areas such as e-learning, entertainment, e-commerce, e-business, web-pages, etc. In last decade, many e-commerce websites embedded the RS into their applications and provides a beneficial growth. In general, RS are classified into three main categories as Collaborative Filtering (CF), Content-based Filtering (CBF) and Hybrid Filtering (HF). CF is the most common and effective technique of RS and is adopted by most of the entertainment and e-commerce websites. It recommends to active user the same items that other similar users (neighbors) have liked in the past. The similarity of taste between the different users can be calculated based on the similarity in rating history between users. The feedback of other users helps in finding the similarity between items (item-based) and between users (user-based). Broadly, CF is classified into two main categories which are memory-based and model-based methods. Memory-based method assumes that the user and item past ratings in database must be present in the system memory while the algorithm for recommendation is running. The similarity between different users (or items) are calculated by searching the user and item database and then aggregating the opinions of neighbors as recommendations. Memory-based method contains user-based and item-based approaches depending upon the whether the neighbors are derived by identifying similar users and items. Despite of all these, memory-based faces some disadvantages like sparsity, scalability and cold-start problems. Model-based method works on providing recommendations by learning a pre-defined model. The model can be made from data mining and machine learning techniques. The predictions of ratings are provided for shorter duration of time as it uses off-line computation for training. Model-based methods do not suffer from the drawbacks of memory-based method but it

suffers with the problem like decreasing the performance accuracy. The examples include Clustering, Singular Value Decomposition, Association Rule, decision tree etc. This paper presents the improvised K-Means algorithm proposed in literature and improving the quality of recommendations.

K-Means clustering algorithm is one of the simplest and widely used supervised learning that solves the clustering problem. It was developed by Mac Queen in 1967. This approach is based on partitioning the dataset into k number of clusters. The problem with this algorithm is that it is sensitive to the selection of the initial partition and may converge to local optima. Many improvements are done to enhance the performance of K-Means algorithms. Improvement presented in [1] selects first centroid as random and then calculates the rest of centroid by taking the probability proportional to the maximum distance from all existing centroids. Here in this paper improvement of K-Means algorithm presented in [1] is done.
The proposed work aims of developing new K-Means based recommendation approach and comparing it with existing K-Means as well as with similarity method i.e. Pearson Correlation proposed in literature.

The rest of the paper is organized a follows. In Section II, the related work is presented by giving an overview of different clustering algorithms that have been used for recommendation purposes. In Section III, improvised K-Means algorithm with RS is presented. Section IV describe the experimental setup and the results in detail followed by the conclusion and future scope in Section V.

2. RELATED WORK

Sobia Zahra et al. [1] proposed novel centroid selection approaches for K-Means clustering based recommender system. The proposed algorithm showed that the centroid selection in K-Means could improve the performance and save cost. The described algorithm had the ability to exploit underling data correlation structure that improve accuracy of recommendations as it did not choose random centroids. The proposed approach was tested on five datasets from movies, books and music domains. Tapas Kunango et al. [2] proposed an efficient K-Means clustering algorithm with its analysis and implementation. The algorithm required the kd-tree as the only data structure. They established a practical effectiveness of the filtering algorithm in two ways. First, offered a data-sensitive study of the algorithm running time which showed that the algorithm run faster as the distance between clusters increases. Second offered a number of experimental studies both on syntactically generated data and on real datasets in color quantization, data compression and image segmentation. Unnati R. Raval et al. [3] explained the implementation and improvisation of K-Means clustering algorithm. Their approach was based on two approaches i.e. to initially select the centroids of the cluster and assigning data points to nearest cluster by using equation for calculating mean and distance between two data points. F.O. Isinkaye [4] described the recommender systems principles, methods and evaluation. This paper explored the different characteristics, prediction techniques of the RS. It explained two traditional recommendation techniques and emphasized their strengths and challenges with various kind of hybridization approaches used to improve performance. Feng Xie et al. [5] described item similarity learning methods for collaborative filtering recommender systems. They proposed a series of item similarity learning methods by minimizing squared prediction error using stochastic gradient descent. The explained method of similarity inherited the interpretability of items based collaborative filtering that produce accurate recommendations as compared to others matrix factorization methods. Yong Wang et al. [6] proposed a hybrid user similarity model for collaborative filtering. The methods was based on Kullback- Liebler divergence which was used as a weight to correct the output of n adjusted proximity significance similarity model. In the approach, user preference factor and an asymmetric factor were considered to distinguish the rating preference between different users and improve the consistency of the proposed model. , O. J Oyelade et al. [7] proposed an application of K-Means clustering algorithm for prediction of student's academic performance. In this approach K-Means clustering algorithm was combined with deterministic model to analyze student's performance of institute in Nigeria. They provided a simple and qualitative approach to compare the predictive power of clustering algorithm and the Euclidean distance as a measure of similarity distance. ShyrShen Yu et al. [8] proposed two improved k-means algorithms. Tri-level K-Means algorithm was used when the data in a cluster were considerably different, a cluster center cannot

alone surely describe each item in the cluster. Noisy data, outliers, and data with quite different values in the same cluster may decrease the performance of pattern matching systems. The bi-layer K-Means algorithm could deal with the above problems. Meanwhile, a genetic-based algorithm was provided to derive the fittest parameters used in the tri-level and bi-layer K-Means algorithms. Experimental results demonstrated that both algorithms could provide much better accuracy of classification than the traditional K-Means algorithm.

Problem Formulation: The approaches based on K-Means clustering have been introduced to solve the issues in recommendations. The selection of cluster centroids in the user item rating matrix has been widely used in the literature, but it is not a reasonable approach. The proposed work aims to improve the quality of clusters and recommendations by using new centroid selection approaches and examining how they affect the quality of recommendations.

3. IMPROVISED K-MEANS ALGORITHM

To address the problem stated in the previous section an improvised K-means is proposed in Figure 1. The dataset is a rating matrix R_{ij} of dimensions m×n containing the ratings of users for items where m is the number of users and n is the number of items. The algorithm chooses k as the number of clusters in which the users are clustered according to similarities in their ratings. A new matrix C_{ij} with dimensions k×n is defined for calculating centroids, initialize its values with zero.

The rating matrix R_{ij} is sorted row wise and divided into k number of parts. These k parts are the clusters. The centroid of each cluster is updated in k×n matrix by calculating the mean of each cluster.

Algorithm: Improvised K-Means for Calculating Centroids
// Calculate the Centroids of k Number of Clusters

Figure 1: Improvised K-Means Clustering

Begin

1. Input a rating matrix R_{ij} // where i = 1…m and j = 1…n, m is the number of users and is the number of items.
2. Define k number of clusters. // where k is the number of clusters to be formed
3. Initialize centroid matrix C_{ij} // where i = 1…k and j = 1…n.
4. Sort the rating matrix R_{ij} on i = 1 and j as multi key sorting.
5. Split the sorted rating matrix R_{ij} into k equal parts except last one.
   // calculating the centroid of clusters
6. For i = 1 to k
   6.1. For j = 1 to n //where n is the number of items
      6.1.1. $C_j = \frac{\sum_{l=1}^{p} R_{lj}}{p}$
   6.2. End For;
7. End For;
8. For i = 1 to k
   8.1. For j = 1 to p //where p is the number of users in the cluster
      8.1.1. For l = 1 to p
         //Calculate the similarity between user j and l for all items
         8.1.1.1. $sim_{jl} = \frac{1}{1+\sqrt{\sum_{k=1}^{n}(R_{jk}-R_{lk})^2}}$
      8.1.2. End For;
      8.1.3. Sort the similarities of user j in descending order.
   8.2. End For;
9. End For;

End

Now the Algorithm described in Figure 2 is used for making recommendations to active user where active user is the user for which the recommendations of any item is made and predicting the rating for that item. For this purpose a rating matrix R_{ij} of dimensions m×n is used where m is the number of users and n is the number of items. From this rating matrix R_{ij}, two new matrices are formed i.e. training matrix and testing matrix. Training matrix T_{ij} have dimensions m×n, it is formed by making the ratings of active users zero for that item whose rating will be predicted whereas testing matrix TS_{ij} is of dimensions r×s where r is number of active users and s is the number of items with its ratings. This testing matrix is formed by taking the actual ratings of active users for the predicting item from the rating matrix R_{ij}.

Algorithm: Proposed Recommender System

//calculate the recommendations and RMSE for active user

Figure 2: Recommender System

Begin

//Initialize rsum with zero

1. Input training matrix T_{ij} // where i = 1…m and j = 1…n, m is the number of users and n is the number of items.
2. Input testing matrix TS_{ij} // where i = 1…r, j = 1…s, r is number of active users and s is the number of items with their ratings.
3. For i = 1 to r

//Calculate average rating of user T_{i1} for n items

3.1. $avg\_rating_i = \frac{\sum_{j=1}^{n} T_{ij}}{n}$

// Find the similarity between user T_{i1} and its neighbors using improvised K-Means clustering by passing training matrix T_{ij}, also calculate the top N number of neighbors of T_{i1}

3.2. topN = N // where N is the top N neighbors of T_{i1}

3.3. For k = 1 to topN

// Initialize arrays sumdiff and simab with zeros

// $usersim_{ik}$ contains the similarity between T_{i1} and nb_k

3.3.1. $nb\_avg\_rating_k = \frac{\sum_{j=1}^{n} T_{kj}}{n}$

3.3.2. sel_rating_k = rating of neighbor nb_k for item TS_{12} from training matrix T_{ij}

3.3.3. $wdiff_k = usersim_{ik} * (sel\_rating_k - nb\_avg\_rating_k)$ // usersim is the similarity between user u_i and neighbor nb_k

3.3.4. $sumdiff_k = sumdiff_{k-1} + wdiff_k$

3.3.5. $simab_k = simab_{k-1} + usersim_{ik}$

3.4. End For;

3.5. $pred\_rating_i = avg\_rating_i + \frac{\sum_{j=1}^{topN} sumdiff}{\sum_{j=1}^{topN} simab}$

3.6. $rdiff_i = (TS_{i3} - pred\_rating_i)^2$

3.7. $rsum_i = rsum_{i-1} + rdiff_i$

4. End For;
5. RMSE $= \sqrt{\frac{\sum_{i=1}^{r} rsum}{r}}$ // RMSE is the Root Mean Squared Error

End

4. EXPERIMENTAL RESULTS

a. Dataset

For the experiments, dataset is made up of data captured from film recommendation websites, which are usually used in the field of RS, they facilitate to measure the scalability of algorithm as well as to benchmark improvised algorithm with some of the state of art algorithms. In this paper, MovieLens 100K ratings are used. The dataset is collected from the MovieLens website (http://www.grouplens.org/datasets/movielens/100k/) with 1, 00000 ratings. This dataset is well known for making recommendations. The number of users are 943 and 1682 are movies. In this dataset the ratings are provided on 5-star scale and only those users are selected who have provided ratings to at least 20 movies. The dataset is split into two parts that is training set and testing test. Training set contains the matrix which is to be trained and testing set contains the actual rating of any user with respect to any item.

b. Metrics

Root Mean Square Error (RMSE): It is a commonly used measure of the differences between values predicted by an estimator and the values actually observed. The RMSE represents the differences between predicted values and observed values. These individual differences are called residuals when the calculations are performed over the data sample that was used for estimation, and are called predicted errors when computed out-of-the sample. The RMSE helps to aggregate the magnitudes of the errors in predictions for various times into a single measure of predictive power. RMSE is a measure of accuracy, to compare forecasting errors of different models for a particular dataset and not between datasets, as it is a scale dependent.

RMSE is a square root of the average of the squared error and is given as

$$RMSE = \sqrt{\frac{\sum(predictive\ rating - actual\ rating)^2}{number\ of\ test\ cases}} \ldots (1)$$

Actual rating is provided in the user-item matrix in the test rating dataset of MovieLens. Prediction rating is given by [9]

$$P_{u,i} = \overline{R_u} + \frac{\sum(R_{u,i} - \overline{R_u}) * sim_{u,a}}{\sum sim_{u,a}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Where $P_{u,i}$ is the prediction rating of user u for item i. $\overline{R_u}$ is the mean rating of user u, $R_{u,i}$ is the rating of user u for item i and $sim_{u,a}$ is the similarity between user u and other user a.

c. Performance Comparison in terms of recommendation RMSE and Execution Time

Recommender Systems based on Pearson Correlation similarity (RS1), existing K-Means algorithm [1] (RS2) and improvised K-Means (RS3) are compared with each other on the basis of execution time and RMSE.

The improvised K-Means is better than the existing algorithm proposed in [1] in terms of making the clusters, calculating centroids values and finding the neighbors of any particular user.

The results over MovieLens dataset in Figure 3 and Figure 4 shows that the RS3 outperforms in terms of execution time and RMSE. The table below shows the values of these approaches. The values from Table 2 and Table 3 shows that the clustering approach is better than the similarity approach for making recommendations. And the clustering approach using improvised K-Means is best suited for RS as it takes less execution time and also gives less value of RMSE.

Table 1: Execution Time of Algorithms

Recommender Systems	
RS1	Recommender System Based on Pearson Correlation Similarity
RS2	Recommender System Based on K-Means clustering
RS3	Recommender System Based on improvised K-Means clustering

Table 2: Execution Time of Algorithms

Execution Time	
RS1	15.780 seconds
RS2	10.845 seconds
RS3	4.6388 seconds

Table 3: RMSE of Algorithms

RMSE	
RS1	1.8022
RS2	1.7089
RS3	1.6837

Figure 3: Execution Time of different approaches of recommendations

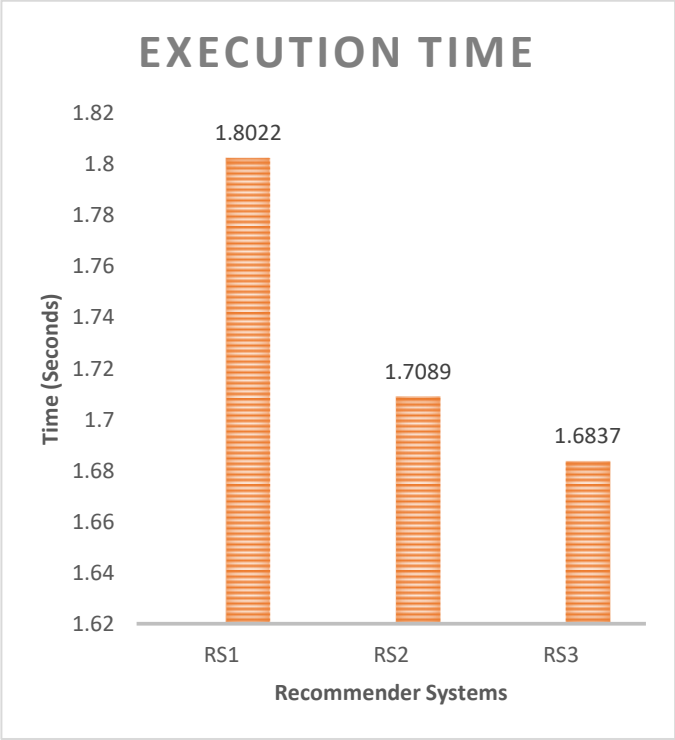
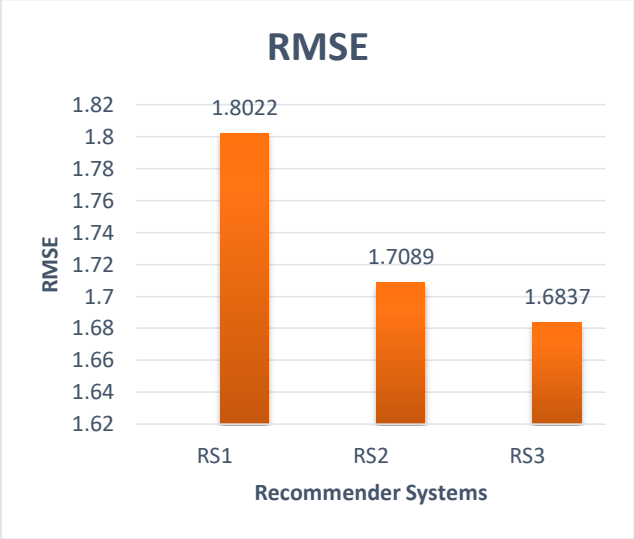


Figure 4: RMSE of different approaches of recommendations



5. CONCLUSION AND FUTURE SCOPE

The main purpose of this work is that if centroids of K-Means algorithm are efficiently selected for partitioning the dataset into clusters then this can give benefits in recommendations like finding the best similar users, recommending the movies of user interest effectively and saving time of execution. This work, presents a comparative study of Recommender systems based on best similarity metric i.e. Pearson Correlation, Existing K-Means algorithm and the improvised K-Means algorithm and their consequent impact upon accuracy and execution time.

A limitation of K-Means algorithm is that it highly depends on k number of clusters and k must be predefined. Developing some statistical methods to compute the value of k, depending on the data distribution, is suggested for future research.

REFERENCES

[1] Zahra, S.; Ghazanfar, M. A.; Khalid, A.; Azam, M. A.; Naeem, U.; Bennett, A. P. (2015): Novel Centroid Selection Approaches for K-Means Clustering Based Recommender Systems. Information Sciences.

[2] Kanungo, T.; Mount, D. M.; Netanyahu, N. S.; Piatko, C. D.; Silverman, R.; Wu, A. Y. (2002): An Efficient K-Means Clustering Algorithm: Analysis and Implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, 881-892.

[3] Raval, U. R.; Jani, C. (2015): Implementing and Improvisation of K-Means Clustering. International

Journal of Computer Science and Mobile Computing, vol. 4, pp. 72-76.

[4] Isinkaye, F. O.; Folajimi, Y. O.; Ojokoh, B. A. (2015): Recommendation Systems: Principles, Methods and Evaluation. Egyptian Informatics Journal, pp. 261-273.

[5] Xie, F.; Chen, Z.; Shang, J.; Huang, W.; Li, J. (2015): Item Similarity Learning Methods for Collaborative Filtering Recommender Systems. IEEE 29th International Conference on Advanced Information Networking and Applications, pp. 896-903.

[6] Wang, Y.; Deng, J.; Gao, J.; Zhang, P. (2017): A Hybrid User Similarity Model for Collaborative Filtering. Information Sciences.

[7] Oyelade, O. J.; Oladipupo, O. O.; Obagbuwa, I. C. (2010): Application of K-Means Clustering algorithm for prediction of Students' Academic Performance. International Journal of Computer Science and Information Security, vol. 7, pp. 292-295.

[8] Yu, S. S.; Chu, S. W.; Wang, C. M.; Chan, Y. K.; Chang, T. C. (2017): Two Improved K-Means Algorithms. Applied Soft Computing.

[9] Gong, S. (2010): A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering. Journal of Software, vol. 5, pp. 245-252.