Model-based Collaborative Filtering using Refined K-Means and Genetic Algorithm

A. Suresh Poobathy[1], C. Manimegalai[2]

[1]Dept. of Mathematics, Pondicherry University Community College, Puducherry, India.
spsureshpoobathy22@gmail.com

[2]Department of Computer Science, KMCPGS, Puducherry, India.
manispb20@gmail.com

ABSTRACT

As cloud computing has emerged as new computing paradigm, more and more web services has been provided on the Internet, thereby how to select a qualified service is becoming a key issue. Several approaches based on Clustering, e.g., K-Means (KMC) Clustering, Fuzzy clustering, Subtractive Clustering has been proposed. KMC is a popular clustering algorithm based on the partition of data. However, it has some limitations, such as its requiring a user to give out the number of clusters at first, and its sensitiveness to initial conditions, and second it can only find linearly separable clusters. In this paper, we have proposed a new context known as Refined K-Means Clustering (KMC) and Genetic Algorithm. Refined KMC is an extension of standard KMC to solve the limitations of standard KMC and provide recommendation. Genetic Algorithm is used to improve the cluster quality than standard KMC and Refined KMC.

Keywords: Recommender Systems, Collaborative Filtering (CF), K-Means Clustering, Refined K-Means Clustering, Genetic Algorithm.

1. Introduction

Collaborative Filtering technology is one of the most successful techniques in Recommender Systems. With the gradual increase of customers and products in electronic commerce, the time-consuming for finding the nearest neighbor becomes too difficult. CF is a type of system used to generate user-specific recommendations based on common items between two similar users. CF is a recommendation technique [9] that identifies similarities between users, based on their ratings in order to select neighbors and compute predictions for the active users.

Clustering is an unsupervised learning algorithm [3]. Applying data clustering algorithms to ratings data in CF will result in a partition of data based on user rating data. Predictions are then computed independently with each partition. Clustering methods have been integrated in several CF based recommender systems in order to reduce dimensionality or to alleviate the sparsity and scalability problems [9]. Various Clustering algorithms such as K-Means clustering, Fuzzy clustering, Subtractive Clustering and K-Modes Clustering are available to partition the data in an efficient way.

Standard K-Means Clustering

K-Means Clustering is an iterative algorithm in which items are moved among set of clusters until the desired set is reached [1]. KMC is used to reduce the search space [8]. The standard KMC [8] method creates k clusters each of which consist of the customers who have similar preferences among themselves. In this method, arbitrarily k customers as the initial center points of the k clusters are selected, respectively. Then each customer is assigned to a cluster in such a way that similarities between the customer and the center of a cluster is maximized.

Then, for each cluster, the mean of the cluster based on the customers who currently belong to the cluster is recalculated. The mean is now considered as the new center of the cluster. After finding new centers, compute the similarity for each customer as before in order to find to which cluster the customer should belong. Recalculating the means and computing the similarity are repeated until a terminating condition is met. The customers within the cluster have greater similarity, but are dissimilar to customers in other clusters thus satisfying the principle of maximizing the intraclass similarity and minimizing the interclass similarity.

There are several similarity algorithms that have been used in the CF recommendation systems: Pearson Correlation, cosine vector similarity, adjusted cosine vector similarity, mean-squared difference and Spearman correlation.

The below pearson’s correlation formula is used to measure the linear correlation between two vectors of ratings as the target item i and the remaining item [7].

$$\mathbf{sim}\,(\mathbf{i},\mathbf{k}) = \frac{\sum_{j=1}^{n} (R_{i,j} - A_i)(R_{k,j} - A_k)}{\sqrt{\sum_{j=1}^{n} (R_{i,j} - A_i)^2 (R_{k,j} - A_k)^2}} \quad \textbf{(1)}$$

Where $R_{i,j}$ is the rating of the target item j by user i, $R_{k,j}$ is the rating of the target item j, by user k, A_i is the average rating of user i, A_j is the average rating of user j and n is the number of items.

2. Related Works

Applying data clustering algorithms to ratings data in CF, it partitions the data into clusters (group). Existing data partitioning and clustering algorithm is used to partitions the set of items based on user rating data. Predictions are then computed independently within each partition. Ideally, partitioning will improve the quality of CF predictions and increase the scalability of CF systems [6].

Personalized recommendation systems can help people to find interesting things and they are widely used with the development of electronic commerce. Many recommendation systems employ the CF technology [7], which has been proved to be one of the most successful techniques in recommender systems in recent years. With the gradual increase of customers and products in electronic commerce systems, the time consuming nearest neighbor CF search of the target customer in the total customer space resulted in the failure of ensuring the real time requirement of recommender system.

At the same time, it suffers from its poor quality when the number of records in the user database increases. Sparsity of source data set is the major reason causing the poor quality to solve the problems of scalability and sparsity in the CF, a personalized recommendation approach joins the user clustering technology and item clustering technology. Users are clustered based on users' ratings on items and each users cluster has a cluster center. Based on similarity between the target user and cluster centers, nearest neighbors of target user can be found and smooth the prediction where necessary. Then, the proposed approach utilizes the item clustering CF to produce the recommendations. The recommendation joining user clustering and item clustering CL is more scalable and more accurate than the traditional one.

At the basis of analyzing products rated by target user, forecasting the value of unrated items, then items with higher forecasting values are selected [2]. The rating data can be collected by directed or concealed way, such as analyzing user's click-stream and behaviour.

The rating data of user forms one mxn matrix, R(m,n). In the matrix, row m represents user m, column n represents item n. the element $R_{i,j}$ in i^{th} row and j^{th} column represents the rating data of user i about item j.

Collaborative system generates recommendations at the basis of similarity among users. CF algorithm is a successful method, widely applied in many e-Commerce systems, such as recommending movies or news for user. CF algorithm evaluates current user's near neighbors according to the product rating data of user. Through neighbors' rating data, the current user's evaluation for a new product can be forecasted, then, the recommendation for current user can be gained. This kind of recommender also has defaults, such as data sparsity, new user and new product problem.

GA searches the optimizing solution through simulating nature evolution process. According to adaptive learning process and parallel process, the complex un-structure problem can be tackled by less computing cost.

GA is an incrementally approach which is used to provide a possible solution to a classic problem: given an array of numbers, find a partitioning among items and produce the best fitness value [5].

3. Refined K-Means Clustering

Since the magnitudes of users and products are huge, the rating matrix is typical sparse space. Standard KMC work poor in this application field. Hence, Refined KMC is introduced to improve the quality goodness by means of using the Restraint function and a similarity measure.

Restraint Function

While measuring their similarity, three kinds of situations will be encountered.

(1) Regarding item j, all of the user I and k have rated, so, the similar degree may be compared;
(2) Regarding item j, the user i has rated, but user k does not has, this time if calculating their similarity on time j, the similar degree will be reduced. Obviously, these rating data are different. The user k has not rated item j, may be he has not used this product, but not regard rating data is 0 on item j;
(3) Regarding item j, both of user i and k have not rated, in this situation, if calculates their similarity on item j, the similar degree will be increased. Because two users' rating data are not the identical value – 0, in majority situation.

If cannot restraint the latter two kinds of situations, similarity evaluating will be out of true, thus the quality of clustering will be drop. Hence, it is necessary to introduce a restraint function. When comparing two users, those components which having rating data be selected to calculate the similar degree.

The restraint function is a decision equation is as follows:

$$R(R_{i,j}) = \begin{cases} 1 & \text{if user i rated item j} \\ 0 & \text{if user i not rated item} \end{cases} \qquad (2)$$

The KMC starts from random partitioning, the algorithm repeatedly (i) computes the current cluster centers (i.e. the average vector of each cluster in data space) and (ii) reassigns each data item to the cluster whose center is closest to it. It terminates when no more reassignments take place.

Similarity

When calculating the distance between of users i and k, restraint function R is applied, causes the computing in those item which user rated. So, the function can be used to revise Euclidean metric for solving the problem of sparsity of user rating data. The below equation is applied to measure user's similarity.

$$d\,(i,k) = \sqrt{\sum_{j=1}^{n} R\,(R_{i,j})R\,(R_{kj})(R_{i,j} - R_{k,j})} \qquad (3)$$

Prediction

Suppose C is set of users, P is set of products, R is set of rating. Rating values describe attitudes or preferences of users about products. In fact, different method constructs different forecasting function E, to predict those rating values of un-rating items. It is,

$$E : C\,X\,P \rightarrow R \qquad (4)$$

Then, the product with the highest predicting rating value is recommended to user c.

$$\forall\, c \in C, p_c = \arg\max e(c,p) \qquad (5)$$

For k recommenders, the forecasting result of recommender k is vector

$$E_k = (\,e_{1k}, e_{2k}, \ldots \ldots \ldots \ldots \ldots \ldots, e_{Nk}\,) \qquad (6)$$

e_{ik} is attitude of current user about product i. N is the number of un-rating items.

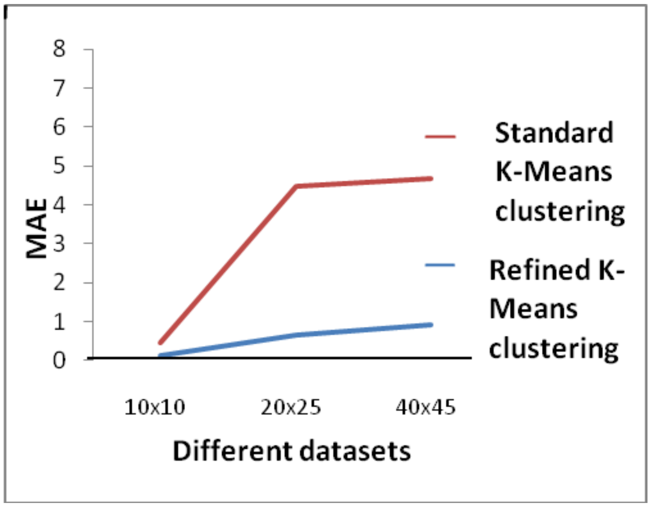
Quality Metrics

Several metrics have been proposed for assessing the accuracy of CF methods. These metrics evaluate the accuracy of a prediction algorithm by comparing the numerical deviation of the predicted ratings from the respective actual user ratings. Some of them frequently used are Mean Absolute Error (MAE) and Root Mean Squared error (RMSE).

MAE

Mean Absolute Error (MAE) is applied to evaluate recommendation quality. Let forecasting rating set is { p_1, p_2, p_3, ... ,p_n }, the real rating set is { q_1, q_2, q_3, ... ,q_n }, then

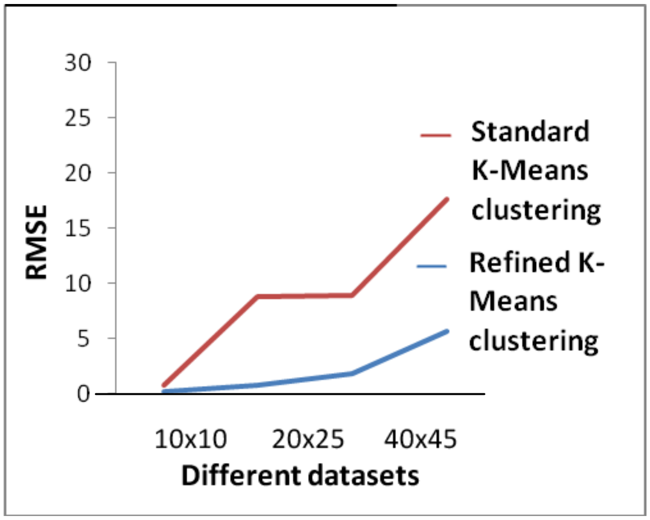
$$MAE = \frac{\sum_{i=1}^{n} |\,p_i - q_i\,|}{n} \qquad (7)$$



RMSE

Root Mean Squared Error (RMSE) is applied to evaluate recommendation quality. Let forecasting rating set is { p_1, p_2, p_3, ... ,p_n }, the real rating set is { q_1, q_2, q_3, ... ,q_n }, then

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\,p_i - q_i\,)}{n}} \qquad (8)$$



4. Genetic Algorithm

GA is randomized search and optimization techniques guided by the principles of evolution and natural genetics, having a large amount of implicit parallelism. In GA [3], the parameters of the search space are encoded in the form of strings (called chromosomes). A collection of such strings is called a population. Initially, a random population is created, which represents different points in the search space.

An objective and fitness function is associated with each string that represents the degree of goodness of the string. Based on the principle of survival of the fittest, a few of the strings are selected and each is assigned a number of copies that go into the mating pool. Biologically inspired operators like cross-over and mutations are applied on these strings to yield a new generation of strings. The process of selection, crossover and mutation continues for a fixed number of generations or till a termination condition is satisfied.

The basic reason for refinement is, in any clustering algorithm the obtained clusters will never give 100% quality. There will be some errors known as mis-clustered. These kinds of errors can be avoided by using our refinement algorithm.

The cluster obtained from the KMC is considered as input to our refinement algorithm. Initially a random point is selected from each cluster; with this a chromosome is build. Like this an initial population with 10 chromosomes is build. For each chromosome the entropy and f measure is calculated as fitness value and the global minimum is extracted.

With this initial population, the genetic operators such as reproduction, crossover and mutation are applied to produce a new population. While applying crossover operator, the cluster points will get shuffled means that point can move from one cluster to another. From this new population, the local minimum fitness value is calculated and compared with global minimum. If the local minimum is less than the global minimum then the global minimum is assigned with the local minimum, and the next iteration is continued with the new population. Otherwise, the next iteration is continued with repeated for N number of iterations.

String Representation

Here the chromosomes are encoded with real numbers; the number of genes in each chromosome is equal to the number of clusters. Each gene will have 5 digits for vector index. For example, data set containing 5 clusters, so a simple chromosome may looks like as follows:

00100 10010 00256 01875 00098

Here, the 00098 represents, the 98th instance is available at first cluster and the second gene says that the 1875 instance is at second cluster. Once the initial population is generated now it is ready to apply genetic operators.

Reproduction (Selection)

The selection process selects the chromosomes from the mating pool directed by the survival of the fittest concept of natural genetic systems. In the proportional selection strategy adopted in this article, a chromosome is assigned a number of copies, which is proportional to its fitness in the population, that go into the mating pool for further genetic operations.

Roulette Wheel Selection is one common technique that implements the proportional selection strategy.

Crossover

Crossover is probabilistic process that exchanges information between two parent chromosomes for generating two child chromosomes. In this work, single point crossover with a fixed crossover probability of p_c is used. For chromosomes of length l, a random integer, called the crossover point, is generated in the range [l,l-1]. The portions of the chromosomes lying to the right of the crossover point are exchanged to produce two offspring.

Mutation

Each chromosome undergoes mutation with a fixed probability p_m. For binary representation of chromosomes, a bit position (or gene) is muted by simply flipping its value. Since we are considering real numbers in this work, a random position is chosen in the chromosome and replace by a random number between 0-9.

After genetic operators are applied, the local minimum fitness value is calculated and compared with global minimum. If the local minimum is less than the global minimum then the global minimum is assigned with the local minimum, and the next iteration is continued with the new population. The cluster points will be repositioned corresponding to the chromosome having global minimum. Otherwise, the next iteration is continued with the same old population. This process repeated for N number of iterations.

Entropy

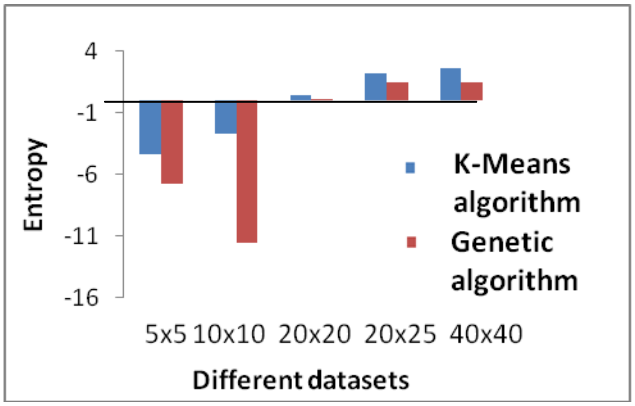
Entropy is a measure of quality of the cluster. The best entropy is obtained when each cluster contains exactly on data point. For each cluster, the class distribution of the data is calculated first, i.e., for cluster j we compute p_{ij}, the “probability” that a member of cluster j belongs to class i. Then using this class distribution, the entropy of each cluster j is calculated using the standard formula

$$E_j = -\sum_i p_{ij} \log(p_{ij}) \qquad (9)$$

where the sum is taken over all classes. The total entropy for a set of clusters is calculate as sum of the entropies of each cluster weighted by the size of each cluster:

$$E_{cs} = -\sum_{j=1}^{n} \frac{n_j E_j}{n} \qquad (10)$$

where n is the size of cluster j, m is the number of clusters, and n is the total number of data points.



F-measure

The second external quality is the F measure, a measure that combines the precision and recall ideas from informational retrieval. We treat each cluster as if it were the result of a query and each class as if it were the desired set of data points for a query. We then calculate the recall and precision of that cluster for each given class. More specifically, for cluster j and class i

$$\text{Recall}\,(i,j) = \frac{n_{ij}}{n_i} \qquad (11)$$

$$\text{Precision}\,(i,j) = \frac{n_{ij}}{n_j} \qquad (12)$$

where n_{ij} is the number of members of class i in cluster j, n_j is the number of members of cluster j and n_i is the number of member of class i.

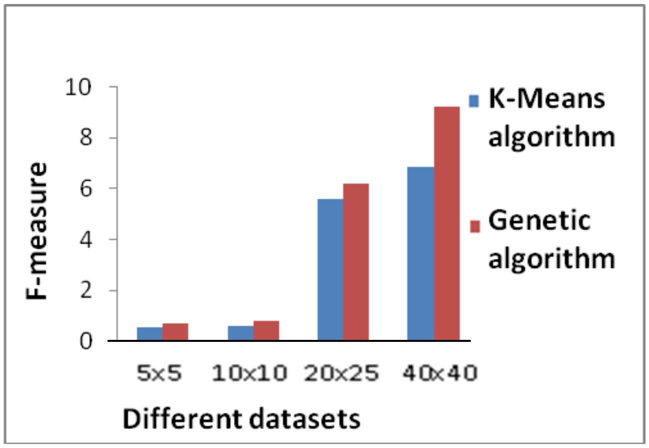
The F measure of cluster j and class i is then given by

$$F\,(i,j) = \frac{(2*\text{Recall}\,(i,j)*\text{Precision}\,(i,j))}{(\text{Precision}\,(i,j) + \text{Recall}\,(i,j))} \qquad (13)$$

The overall value for the F measure is computed by taking the weighted average of all values for the F measure as given by the following.

$$F = \sum \frac{n_i}{n} \max\{F\,(i,j)\} \qquad (14)$$

Where the max is taken over all clusters at all levels, and n is the number of data items.



5. Conclusion

In this work, a new framework called Genetic algorithm (GA) is proposed to improve the cluster quality from KMC and Refined KMC using GA. It is found that our proposed work achieves better results, and also outcome of Recommendation CF is taken as input to the GA and external quality measures, such as Entropy and F-Measure values are checked. It is found that the application of GA improve the quality of traditional KMC and Refined KMC.

In future, the same method has to be tested with huge volume of dataset and also desired to generate a new dataset using more quality factors such as Failure rate, Throughput, etc.

6. References

i. *Chittu V., N. sumathi, "A Modified Genetic algorithm K-Means Clustering", Global Journal of Computer Science and Technology, Vol. 1, Issue 2, Version 1.0 February 2011.*

ii. *Gediminas Adomavicius and YoungOk Kwon. "New Recommendation Techniques for Rating Systems". IEEE Intelligent, May/June: 48-55 2007.*

iii. *Kohrs, A., Merialdo, B.: "Clustering for Collaborative Filtering Applications". In Computational Intelligence for Modelling, Control & Automation. IOS Press, 1999.*

iv. *Lyle H. Ungar, Dean P. Foster, "Clustering Methods for Collaborative filtering".*

v. *Matthijs Dorst, "Solving partition problem using Genetic Algorithm".*

vi. *M. O. Corner and J. Herlocker. "Clustering Items for Collaborative Filtering". In Proceedings of the ACM SIGIR Workshop on Recommender Systems, Berkeley, CA, August 1999.*

vii. *SongJie gong, "A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering".*

viii. *Taek-Hun Kim, Young-suk Ryu, Seok-In Park, and Sung-Bong Yang, "An Improved Recommendation Algorithms in collaborative Filtering".*

ix. *Vreixo Formoso, Fidel Cacheds, Victor Carneiro, "Algorithms for Efficient Collaborative Filtering".*