Improved Item Based Collaboration Filtering New recommendation System

Manmohan singh[1], Jay Prakash Maurya[2],Anil Bhawarkar[3],
Sudhir Nagwanshi[4],Ratna Nayak[5]
[1] Asst. Prof. Department of CSE BIST Bhopal 462021 M.P. (India), manmohan_sati@yahoo.co.in
[2] Department of CSE BIST Bhopal 462021 M.P. (India), jpeemaurya@gmail.com
[3] Department of CSE BIST Bhopal 462021 M.P. (India), bhawarkar_2006@in.com
[4] Department of CSE BIST Bhopal 462021 M.P. (India), sidhu4u12@gmail.com
[5]Department of CSE BIST Bhopal 462021 M.P. (India), nayak.ratna15@gmail.com

Abstract: Recommender systems apply data analysis techniques to the problem of helping users find the items they would like to purchase at E-Commerce sites by producing a predicted likeliness score or a list of top-N recommended items for a given user. We apply improved K-mean algorithms method on preprocessed data. Finally we proposed a method that can increase accuracy based on previous K-mean. Recommender system applies knowledge discovery techniques to the problem of making personalized recommendation for information. Products or services during a live interaction. This system especially the k-nearest neighbor collaborative filtering based once, these are producing high recommendations performing many recommendations per second for millions of users and items and achieving high coverage in the face of data sparsely. In traditional collaborative filtering system the amount of work increases with the number of participants in the system..

Keywords:-K-means,filtering,E-commerce

INTRODUCTION:-

Recommender systems apply knowledge discovery techniques to the problem of making personalized recommendations for information, products or services during a live interaction. These systems, especially the k-nearest neighbor collaborative filtering based ones, are achieving widespread success on the Web. The tremendous growth in the amount of available information and the number of visitors to Web sites in recent years poses some key challenges for recommender systems [Aggarwaletal.1999].Theseare: producing high quality recommendations, performing many recommendations per second for millions of users and items and achieving high coverage in the face of data sparsely [Aggarwal et al., 1999]. In traditional collaborative filtering systems the amount of work increases with the number of participants in the system. New recommender system technologies are needed that can quickly produce high quality recommendations, even for very large-scale problems. To address these issues we have explored item-based collaborative filtering techniques. Item-based techniques first analyze theuser-itemmatrixto identify relationships between different items, and then use these relationships to indirectly compute recommendations for users. We look into different techniques for computing item-item

[ISSN 2250 - 3765]

similarities (e.g., item-item correlation vs. cosine similarities between item vectors) and different techniques for obtaining recommendations from them (e.g. weighted sum vs. regression model). Finally, we experimentally evaluate our results and compare them to the basic k-nearest neighbor approach. Our experiments suggest that item-based algorithms provide dramatically better performance than user-based algorithms, while at The same time providing better quality than the best available user-based algorithms.

Collaboration Filtering:-

Collaborative Filtering systems can produce personal recommendations by computing the similarity between your preference and the one of other people .One of the most promising such technologies is collaborative filtering [shardanand et al.,1995].Collaborative Filtering works by building a database of preferences for items by users. A new user, Neo, is matched against the database to discover neighbors, which are other users who have historically had similar taste to Neo. Items that the neighbors like are then recommended to Neo, as he will probably also like them. Collaborative Filtering has been very successful in both research and practice, and in both information filtering applications and E-commerce applications. However, there remain. Important research questions in overcoming two fundamental challenges for Collaborative Filtering recommender systems.

The first challenge is to improve the scalability of the Collaborative Filtering algorithms. These algorithms are able to search tens of thousands of potential neighbors in real-time, but the demands of modern systems are to search tens of millions of potential neighbors. Further, existing algorithms have performance problems with individual users for whom the sithas large amounts of information. For instance, if a site is using browsing patterns as indications of content preference, it may have thousands of data points for its most frequent visitors. These ``long user rows" slow down the number of neighbors that can be searched per second, further reducing scalability.

The second challenge is to improve the quality of the recommendations for the users. Users need recommendations they can trust to help them find items they will like. Users will "vote with their feet" by refusing to use recommender systems that are not consistently accurate for them. In some ways these two challenges are in conflict, since the less time an algorithm spends searching for neighbors, the more scalable it will be, and the worse its quality. For this reason, it is important to treat the two challenges simultaneously so the solutions discovered are both useful and practical. In this paper we address these issues of recommender systems by applying different approach Item-Based algorithms. The bottleneck in conventional Collaborative Filtering algorithms is the search for neighbors among a large .user population of potential neighbors [Her locker et al., 1999]. Item-based algorithms avoid this bottleneck by exploring relationships between items first, rather than the relationships between users. Recommendations for users are computed by finding items that are similar to other items the user has liked. Because the relationships between items are relatively static, item-based algorithms may be able to provide the same quality as the user-based algorithms with less online computation.

Collaborative-Filtering-enabled Web sites that recommend books, CDs, movies, and so on, have become very popular on the Internet. Such sites recommend items to a user on the basis of the opinions of other users with similar taste. Recommender systems apply data analysis techniques to the problem of helping users find the items they would like to purchase at E-Commerce sites by producing a predicted likeliness score or a list of *top*-N recommended items for a given user. Item recommendations can be made using different methods. Recommendations can be based on demographics of the users, overall top selling items, or past buying habit of users as a predictor of future items. Collaborative Filtering (CF) [Shardanand et al., 1995] is the most successful recommendation technique to date. The basic idea of CF-based algorithms is to provide item recommendations or predictions based on the opinions of other like-minded users. The opinions of users can be obtained *explicitly* from the users or by using some implicit measures. The growth of the Internet has resulted in the availability of a tremendous amount of information and a vast array of choices for consumers. Recommenders systems are designed to help a user cope with this situation by selecting a small number of options to present to the user. They filter and recommend items on the

[ISSN 2250 - 3765]

basis of a user preference model. Among the various types of recommender systems that have been proposed, their filtering techniques fall into two categories: content-based filtering and Collaborative Filtering or social filtering.

Memory-basedCollaborativeFiltering

Algorithms:-

Memory-based algorithms utilize the entire user-item database to generate a prediction. These systems employ statistical techniques to find a set of users, known as neighbors, that have a history of agreeing with the target user (i.e., they either rate different items similarly or they tend to buy similar sets of items).

Proposed Work:-

In this section we briefly present my work related to Collaborative Filtering, recommender system. Other technologies have also been applied to recommender system, including clustering. Clustering is a division of data into groups of similar objects. Representing the data by fewer clusters necessarily loses certain fine details, but achieves simplification. It models data by its clusters. Data modeling puts clustering in a historical perspective rooted in mathematics, statistics, and numerical analysis.

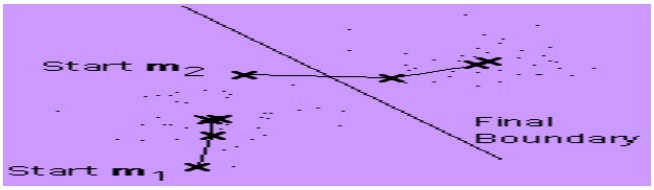
Experimental Procedure:-

Experimental steps: We started our experiment by dividing the data set into training set and test set. First we have to find the sensitivity of data by different algorithm. to find the sensitivity of data we work with only training data then we further divide the training data into two parts for finding sensitivity of data we randomly choosing training and test data each time and talking MAE (Karypis G., 2000) K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid.

The algorithm is composed of the following steps-

Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.

Assign each object to the group that has the closest centroids.When all objects have been assigned, recalculate the positions of the K centroids.Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from.



To predict means m1 and m2 move into the centers of two clusters.

Proposed Algorithm

Start: Clustering al go (Iteration, Counting, Destination, Center, Check_result)

Step1: Initialize: Iteration=0; Counter=0; a by Two dimension Array; Center of the object, new center of the new object; Final Center of the resultant Object;

Step2: X position and Y position is defined;

Step3: Initialize Graphics Mode.

Step4: Now insert Threshold Value and insert no of objects

Step5:loop1: until value not equal to 0;
Draw the Circle according to the Object value: show the cluster;

Step6:loop2 If Circle is <=3 then don't Change the iteration;

Step 7:loop3 An Iteration =3|| iteration=2;Else if circle>=3 Then iteration++;Or Iteration=iteration+1;

Step8: Calculate the Cluster and center points of the object X direction=(a[loop3][0]-enter[loop][0]*a[loop3][0]-center[0];
Direction=(a[loop3][1]-center)*(a[loop][1]-center).Compute the square root of x and Y result;
End loop.

[ISSN 2250 - 3765]

Step9: Rescanning the object and results Loop1: Compare the threshold value with Cluster destination Increment the Cluster Loop2: Set the Cluster position: Now again Calculate X and Y direction Values Go to Step 5.

Step10: Again Compute the square root of x and Y result If Square Root<threshold value the Max

value=new Square root Circle radius =Max value; Print :all Cluster:

Step11: Merge the object with all cluster Loop1 start Initialize value1.Loop2 start Initialize value 2; Calculate X and Y direction Value Go to step5; New Square root of x and y Value New center=new square root;

Step12: If object<previous result Cluster<previous Cluster Then PRINT New result PRINT destination Matrix PRINT evolution Matrix; PRINT new comparing Result and final clusters Again PRINT "Iteration and Final Comparing Values (Errors).

Improved clustering algorithm for finding better result

START: Clustering_algo Iteration,Counting,Destination,Center,Check_result)

Step 1:

Initialize: Iteration=0;Counter=0;a by Two dimension Array;Center of the object; New center of the new object;Final Center of the resultant Object;

Step 2: X position and Y position is defined;

Step 3: Initialize Graphics Mode;

Step4: Now insert Threshold Value and insert no of objects

Step5: loop1: until value not equal to 0;

Draw the Circle according to the Object

value: show the cluster;

Step 6: loop2 If Circle is <=3 then don't Chan

the iteration;

Step 7: loop3 An Iteration =3|| iteration=2;

Else if circle>=3 Then iteration++; Or

Iteration=iteration+1;

Step8: Calculate the Cluster and center points of the object

X direction=(a[loop3][0]-Center[loop][0]*

a[loop3][0]-center[0];

Ydirection=(a[loop3][1]-center)*(a[loop][1]- center);

Compute the square root of x and Y result;End loop;

Step9: Rescanning the object and results

Loop1: Compare the threshould value with Cluster destination Increment the cluster

Loop2: Set the Cluster position:

Now again Calculate X and Y direction Values Go to Step 5;

Step10: Again Compute the square root of x and

Y result If Square root<threshold value the

Maxvalue=new Squar root Circle radius =Max value;Print :all Cluster:

Step11: Merge the object with all cluster

Loop1 start Initialize value1;

Loop2 start Initialize value 2;

Calculate X and Y direction Value Go to

Step5;

[ISSN 2250 - 3765]

New Square root of x and y Value New center=new square root;

Continue object scanning; And calculate the object and cluster then merge them; Repeat Step Untill is not finished; If merging is finished then print a new result after merging: Now rescanning the result again; Repeat step

Step12: If object<previous result and

Cluster<previous Cluster

Then PRINT New result:

PRINT destination Matrix

PRINT evolution Matrix;

PRINT New comparing Result and final cluster Again PRINT "Iteration and Final Comparing Values (Errors)".

Experimental platfor All our experiment is implemented using C++ technology . we use excess for creating Database.

Experimental Results:-In this section we present our experimental results of applying item-based collaborative filtering techniques for generating predictions. Our results are mainly divided into two parts-quality results and performance results. In assessing the quality of recommendations, we first determined the sensitivity of some parameters before running the main experiment. These parameters include the neighborhood size, the value of the train/test ratio x, and effects of different similarity measures. For determining the Sensitivity of various parameters, we focused only on the train data set and further divided it into a train and a test portion and used then to learn the parameter.

Experimental Setup:

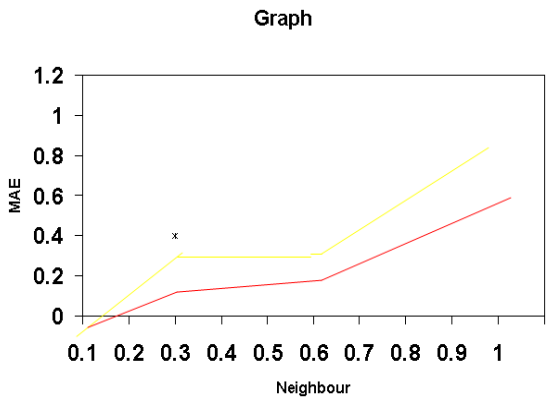
We experimented with the MovieLens1, EachMovie2, and ook-crossing3 data sets. While we report only the Movie Lens results (out of space considerations), the model behaves consistently across the three data sets. The Movie Lens data set contains 100,000 ratings (1-5 scales) from 943 users on 1682 movies (items), where each user has rated at least 20 items. To test on different number of training users, we selected the users in the data set at random into a training user set (100, 200, 300, training users, respectively) and the remaining users into a test user set.

Table 1.1

Neighborhood	MAE
0.3	0.4
0.4	0.4
0.6	0.4
0.8	0.6
01	01

Comparison graph for clustering and method:-

Now its time for comparing. It was found that Comparison of prediction quality of Collaborating filtering and Improved collaborative filtering algorithm. We compare prediction quality at x=0.3,0.6



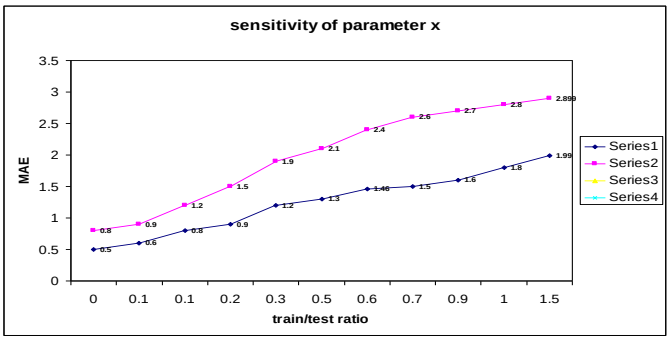
Comparison graph 1.1

Neighborhood	MAE	MAE
0	0.5	0.8
0.1	0.6	0.9

[ISSN 2250 - 3765]

0.1	0.8	1.2
0.2	0.9	1.5
0.3	1.0	1.9
0.5	2.0	2.1
0.6	2.5	2.4
0.8	2.9	2.6

Table 1.2: Calculated Values



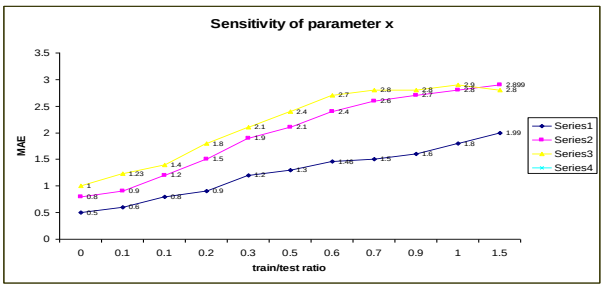
Comparison graph 1.2: Sensitivity of parameter x versus mean absolute

Series 1: shows improved clustering line chart.
Series 2: shows previous clustering line chart

We have drawn two conclusions from the result. First proposed item based algorithm provide better quality then previous one .Second the old clustering based algorithm perform better with very sparse data set, but as we add more data the quality goes down

Neighborhood	MAE	MAE	MAE
0	0.8	1	0.5
0.1	0.9	1.23	0.6
0.1	1.2	1.4	0.8
0.2	1.5	1.8	0.9
0.3	1.9	2.1	1.0
0.5	2.1	2.4	2.0
0.6	2.4	2.7	2.5
0.8	2.6	2.8	2.9

Table.1.3 Proposed Algorithms



Comparison graph 1.3: Sensitivity of mean absolute error versus parameter

Series 1: shows improved clustering line Chart. Series 2: shows previous clustering line Series 3: shows user based line chart

In this section we present our experimental results of applying item-based collaborative filtering techniques for generating predictions. Our results are mainly divided into two parts-quality results and performance results.

Performance Results

Having clearly established the superior quality of item-based algorithms over the user-based ones, we focus on the scalability challenges item-based similarity is more static and allows us to pre compute the nearest items. These methods although save the time require an 0(n2) space for n times.

Conclusion

In this work we have proposed a recommender systems based on "Item Based Collaboration Filtering". This system helps the users to find items they want to buy. For this we developed efficient Clustering algorithms which are able to generate most similar items efficiently and with high accuracy. In development of our new algorithm we have applied iterative approach on k-means algorithm to improve accuracy of the Clusters. In each iteration algorithm generate the clusters and test the accuracy of the Clusters. This improved algorithm shows how much error is reduced, by

[ISSN 2250 - 3765]

previous one. This is benefit of my algorithms which shows Error and calculates it. After calculating again testing Filtering process is started. This result is computed on Few Data which are predefined in the form of array. The result can be vary if Change the data from the Database.

References:

1) AGGARWAL, C. C., WOLF, J. L., WU, K.-L., YU, P. S. 1999. Horting Hatches an Egg: A New Graph-Theoretic Approach to Collaborative Filtering. In Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.
2) AMENTO, B., TERVEEN, L., HILL, W., HIX, D. and SCHULMAN, R. 2003. Experiments in Social Data Mining. To appear in ACM
3) CANNY, J. 2002. Collaborative Filtering with Privacy via Factor Analysis. In Proceedings of the 25th Annual International ACM SIGIR
Conference on Research and Development in Information retrieval, ACM Press, New York,
4) CLAYPOOL, M., BROWN, D., LE, P, and WASEDA, M. 2001. Inferring User Interest. IEEE Internet Computing 5, 32-39.
CLEVERDON, C., KEAN, M. 1968. Factors Determining the Performance of Indexing Systems.
5) COSLEY, D., LAM, S. K., ALBERT, I., KONSTAN, J. A, and RIEDL, J. 2003. Is Seeing Believing? How Recommender Interfaces off 1999. An Empirical Evaluation of User Interfaces
4) BAEZA-YATES, R., RIBIERO-NETO, B. 1999. ACM Press / Addison Wesley, New York.
BAILEY, B. P., GURAK, L. J., KONSTAN, J. A. 2001. An Examination of Trust Production in Computer-Mediated Exchange. In Proceedings
of the 7th Conference on Human Factors and the Web, July 2001.
5) BALABANOVÍC, M., SHOHAM, Y. 1997. Content-Based, Collaborative Recommendation. Communications of the ACM 40, 66-72.
BASU, C., HIRSH, H., COHEN, W. W. 1998. Recommendation as Classification: 6) BILLSUS, D., PAZZANI, M. J. 1998. Learning collaborative information filters. In Proceedings of the Fifteenth National Conference on