Solving Sparsity Problem in Rating-Based Movie Recommendation System

Nitin Mishra, Saumya Chaturvedi, Vimal Mishra, Rahul Srivastava and Pratibha Bargah

Abstract Recommendation is a very important part of our digital lives. Without recommendation one can get lost in web of data. Movies are also very important form of entertainment. We watch most movies that are recommended by someone or others. Each person likes specific type of movies. So movie recommendation system can increase sales of a movie rent/sales shop. Websites like Netflix are using it. But there is one problem that can cause recommendation system to fail. This problem is sparsity problem. In this paper, we have used a new approach that can solve sparsity problem to a great extent.

Keywords K-mean clustering · Euclidean distance · K-medoid clustering

1 Introduction

Recommendation systems are used for many purposes. They filter the information and give users what they really wanted. In movie recommendation system, we try to recommend movies based on user interests. There are three points to focus for movie recommendation system:

N. Mishra (✉) · S. Chaturvedi · P. Bargah
RCET Bhilai, Bhilai, India
e-mail: drnitinmishra10@gmail.com

S. Chaturvedi
e-mail: saumyanmishra5@gmail.com

V. Mishra · P. Bargah
IERT Allahabad, Allahabd, India
e-mail: vimal.mishra.upte@gmail.com

R. Srivastava · P. Bargah
JLU Bhopal, Bhopal, India
e-mail: rahul04.shri@gmail.com

H.S. Behera and D.P. Mohapatra (eds.), *Computational Intelligence in Data Mining*, Advances in Intelligent Systems and Computing 556,
DOI 10.1007/978-981-10-3874-7_11

Why: Movie recommendation system is required because movie information are overloaded.

Where: Can be used in websites like bookmyshow.com or Netflix.

What: It tells you what you should watch based on your history and other users' history.

2 Basic Preliminaries

Some methods given below are used to predict the items for users. Although there is large list of methods but we are discussing some methods which are of prime importance to movie recommendation.

2.1 Fuzzy C-Mean Clustering

Fuzzy clustering is one of the types of clustering in which every data point can belong to more than one cluster [1, 2]. Clustering or cluster analysis involves transmission data points to clusters (also called buckets, bins, or classes), or homogeneous classes, such that items in the same class or cluster are as similar as possible, while items belonging to different classes are as dissimilar as possible. Clusters are identified via similarity measures.

Advantages

(1) gives most excellent effect for overlap data set.

(2) not like k-means anywhere data point must completely be in the right place to one cluster hub and here data point is assigned association to each cluster middle as a consequence of which facts end may be in the right place to supplementary subsequently single come together center.

Disadvantages

(1) Apriori requirement of the number of clusters.

(2) With lesser value of β we obtain the improved end result but at the expenditure of more numeral of iteration.

2.2 Gath–Geva Clustering

The Gath–Geva algorithm is an addition of Gustafson–Kessel algorithm that take the volume and density of the cluster into report [3].

- The distance function is preferred ultimately comparative to the (unnormalized) a posteriori opportunity, because a small distance results a high probability and a big distance results a low probability of association [4, 5].
- In contrast to FCM algorithm and the Gustafson–Kessel algorithm, the Gath–Geva algorithm is not based on an objective function, but is a fuzzification of statistical estimators [6].

3 Problem Identification

Online shops today contain lot of items and users. In order to relate users to items they need association of user interest on particular items. But due to time or other constraints, it is generally not possible to have enough ratings on particular items by users. This situation gives rise to problem called sparsity. In real world sparsity is very common. We often do not have enough ratings to make our highly efficient recommendation algorithm to work. It is common in e-business supplies that even the majority of active customers had purchased or rate extremely incomplete proportion of products, when compared to the obtainable total. As a result, techniques to reduce the sparsity in user-item matrices should be proposed. The main reason for sparsity problem are as follows:

- The amount of items that contain ratings by the users would be too small. This can make our recommendation algorithms fail.
- Similarly, the number of users who rate one exact item might be too small compared to the total no. of users connected in the system. These situations provide rise to sparse ranking matrix.

4 Methodology

Figure 1 shows the process of our methodology to deal with the sparsity problem.

1. We will collect information from IMDb (Internet Movie Database), All the required information about the movie will be available.

   The sample data is available of the website.

2. Generate the review and rating matrix and apply k-mean clustering in both matrices. It will make cluster of similar object but it has no predefined classes. And classification of reviews is based on good, bad, and average comments of movies we have taken the 29 * 100 matrix for Rating and 29 * 3 are matrix for review.
3. Both data of matrix is converted into relational data using Euclidean distance. After that we can apply k-mediod clustering.

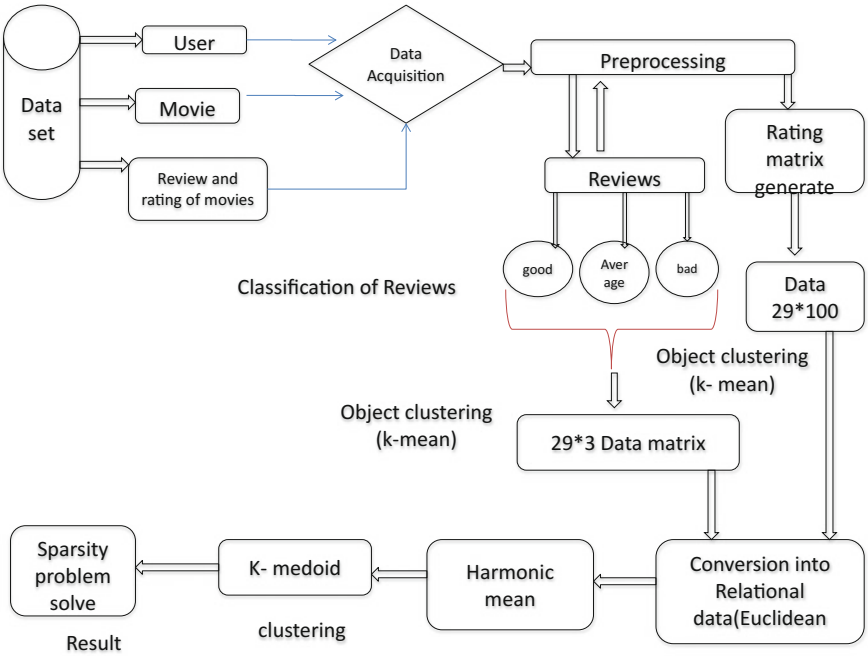

Fig. 1 The process for solving sparsity problem in rating-based movie recommendation system

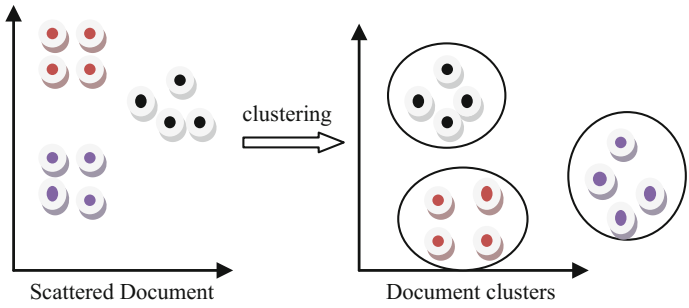

Fig. 2 K-mean clustering

4.1 K-Mean Clustering

The k-means algorithm involves assigning each of the n examples to one of the k clusters, where k is a number that has been defined ahead of time. The goal is to minimize the differences within each cluster and maximize the differences between clusters. These k new centroids, a new binding has to be completed between the similar data set point and the adjacent novel center [7] (Fig. 2).

4.2 Euclidean Distance

In terms of mathematics, the Euclidean distance is the distance between two points in Euclidean space. With this distance, Euclidean space makes a metric space [8]. The associated norm is called the Euclidean norm. In our project first we loaded the rate matrix then rating matrix and applying Euclidean distance in both matrices.

4.3 Harmonic Mean

For two numbers x_1 and x_2, the harmonic mean can be written **as**

$$\frac{H = 2x_1x_2}{X_1 + X_2}$$

k-medoid is based on centroids (or medoids) calculating by minimizing the absolute distance between the points and the selected centroid, rather than minimizing the square distance. As a result, it is more robust to noise and outliers than k-means.

4.4 K-Medoid Clustering

k-medoid is a classical partitioning technique of clustering that clusters the data set of n objects into k clusters known a priori. It is more robust to noise and outliers as compared to k-means, because it minimizes a sum of pairwise dissimilarities instead of a sum of squared Euclidean distances [9].

A medoid can be defined as the object of a cluster whose average dissimilarity to all the objects in the cluster is minimal. It is the most centrally located point in the cluster [9].

5 Experiments and Analysis

Experiment Data:

We have implemented the solution in MATLAB. Some important findings were very positive. The method chosen and discussed above seems to solve the sparsity problem in this domain.

The dataset was collected by the IMDb. The dataset consist of 29 movies and 100 users and each user rating at least 7 movies. The sparsity degree is 99.23%.

The experiments are performed as follows:

- From the data, get original rating of movie results.
- Input original rating of movie result to get the accurate rating of movie.
- Change ratings according to our algorithm.
- Compare the Rating of movie results with our algorithm.

Data Set

Original Movie Data Set

In this project we have divided data into three clustering of rating the range of 0–33%, 34–65%, and 66–100%. In this group, we can easily classify that how no. of users rating the same data in the actual data after observation the data gave clustering.

Object Data Set

Object data is classified by the k-mean clustering; k-means is used for solving clustering problem. The process follows a straightforward and simple method to categorize a certain data set from side to side a definite amount of clusters (assume k clusters) is predetermined a priori. The main idea is to classify k centers, one for each cluster. In our case taken $k = 3$.

Relational Data Set

Relational data are given by k-medoid clustering. *K-means* and *k*-medoids algorithms are splits into some parts (breaking the data set up into groups) and both challenge to decrease the distance between points labeled to the center of the cluster (Fig. 3).

In the above figure, graph shows the result of sparsity problem solved, in this graph x label shows the three no. of clustering and y label shows the cluster size. In the first three bars show the actual data, in second cluster shows the object data and third cluster shows the Relational data, the cluster of actual data is and relational data are. The comparison between both clustering first cluster are same, second cluster has two different and third cluster has two different. Total different is 4.

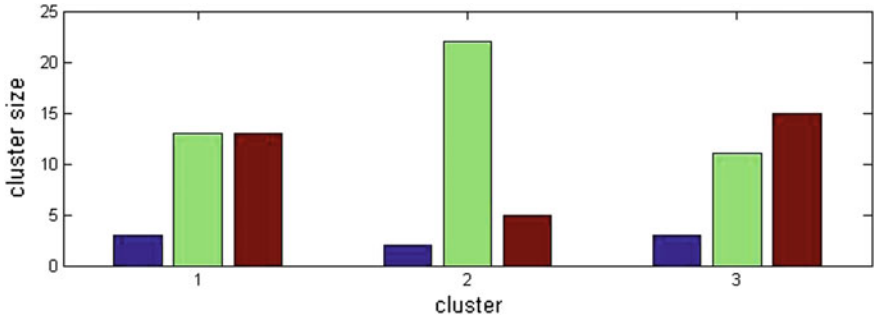

Fig. 3 Graph of result of sparsity problem solved (Generated in MATLAB)

Our dataset contains 29 movies. The actual data and relational data are adjacent value. Hence the k-medoid clustering is the better than the k-mean clustering and 70.22% sparsity problem is solving.

6 Conclusion

In this paper, we have solved sparsity problem in specific dataset of movies. Although we have not tested but this method can be applied on other similar domains like song and video recommendation systems. Our method is very helpful for people developing applications in movie recommendation domain. It will help their recommendation algorithm perform even when there is sparsity of data.

References

1. Mohammed mahmuda rahumen rahumen lecture, et al contextual recommendation system using multidimensional approch. International journal of intelligent information system august 20, 2013.
2. Badrul sarwar, joseph konstan john riedl Using filtering agent to improve prediction quality in the grouplen research collaborative filtering Department of computer science and engineering, University of minnesota in year 2008.
3. Zuping liu sichuon et al. Recommendation algorithm based on user interest, advanced science and technology letters vol. 53, 2014.
4. Beau piccart, jan struf Alleviating the sparsity problem in collaborative filtering by using an adapted distance and a graph based method. IEEE computer technology Year: 2007.
5. Badrul sarwar, george karypis, joseph konstan Item based collaborative filtering recommendation algorithm. Department of computer science and engineering, University of Minnesota Year: 2006.
6. Manos papagelis, dimitris plexousakis Alleviating the sparsity problems of collaborative filtering using trust inferences Institutes of computer science, foundation for research and technology- hellas Years: 2004.
7. Sanghack Lee and Jihoon Yang and Sung-Yong Park, Discovery of Hidden Similarity on Collaborative Filtering to Overcome Sparsity Problem, Discovery Science, 2007.
8. Zan Huang, Hsinchun Chen, et al. Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering. ACM Transactions on Information Systems, Vol. 22, No. 1, January 2004, 116–142. http://dx.doi.org/10.1145/963770.963775.
9. Cheng-kang-Hsieh, et al. Immersive Recommendation: News and Event Recommendations Using Personal Digital Traces (2016) published in Proceedings of the 25th International Conference on World Wide Web on April 11–15, 2016 at Montréal, Québec, Canada.