Recommendation System for MovieLens Dataset using Fuzzy C-Mean Clustering Algorithm

Saurabh Saoji¹, Ashish Tyagi², Apurva Handoo³, Jay Prakash⁴

¹Professor, ²,³,⁴Student
Department of Computer Engineering BV(DU)COE, Pune

Abstract: Analysing the current growth in the internet technology there is a high demand for generation and collection of web generated data. Projecting the correct information to the users is the need of the hour for internet sales, online retailing or e-commerce websites. This thus leads to the complexity in choosing the correct algorithm for displaying accurate results. A recommender system can be defined as an intelligent software that would advice the users based upon their previous purchases or search history thus personalizing the website for the intended user according to the services selected.

Introduction
Recommender systems are systems that uses artificial intelligence along with machine learning technology to provide the appropriate recommendation to users. The major cause behind development of these system is to expand the business of e-commerce sites. All the e-commerce sites require huge dataset so as to provide accurate recommendations to their users. This in turn would increase the users interest in searching through a particular site as accurate recommendation would result into less effort by the users.

The items can be recommended based upon theories like top selling item in a particular category, based upon previous items searched by the user, analyzing previously purchased items or items saved for later preview etc. These systems predict for the users as per their requirements and likes. They generate different set of recommendations for each user thus providing customized environment. Each user gets a personalized dataset to choose from. Nowadays most of the e-commerce organizations have enabled recommendation systems at back end offering web and application based recommendations.

Recommender systems have been developed using data mining techniques, heuristics and finding association patterns among the items. Examples of popular recommender systems include Amazon.com for books, CDs and various other products, Movie Lens for movies.

Related work
Various researches have been performed over improving efficiency of recommender systems or better recommendations to the user. Various clustering and optimization techniques are employed for the same. Each research aiming at improving results of the previously operated research. Recommendation systems are designed using various techniques including k-NN, decision tree, clustering, regression, heuristic methods, neural networks and association rule mining. Based on the type of techniques used, recommendation systems can be classified as content based and collaborative based systems. The content based approach has originated from the information retrieval and information filtering domain.

Content based recommender systems generate recommendations based on users past preferences. The rating for any item for any user is calculated based on ratings of similar items given by the user. Many researchers treat it as a classification problem where the goal is to learn a function that predicts which class a document belongs to that is whether liked or disliked. Others view it as a regression problem where the motive is to learn a function that predicts a numeric value (i.e., the rating of the document). Collaborative systems are different from content based systems in a way that they first find similar users for target users and then generate recommendations based on preferences of similar users. In this approach, recommendations are made by finding correlations among the users. The main objective of collaborative filtering is to find rating of the items, not seen by the current user, using the ratings of similar users.

Collaboration based recommender systems can be further classified into two classes, memory based (heuristic based) and model based collaborative systems [1].

Memory based systems calculate the similarity between users based on users ratings. The algorithms related to memory based systems are heuristics that make recommendations based on complete collection of items pre-rated by the users. Model based collaborative recommender systems generate the descriptive model of the system, based on the users preferences, making use of various data mining and machine learning techniques. The techniques that are used include Bayesian models, clustering models, latent semantic models as singular value decomposition, probabilistic latent semantic analysis, multiple multiplicative factor, latent Dirichlet and Markov decision process based models. The predictions for a new user are made based on the constructed model. There are various other probabilistic modeling techniques used for building recommender systems, available in literature. The Markov model happens to be a complex probabilistic model widely used for modelling sequential events. Several hybrid recommendation systems have also been developed using numerous techniques including a sequential pattern analysis. Most of the work related to the design of a recommendation system with sequential information used the Markov model. Sequential and association pattern mining algorithms have been developed to find the sequential patterns in the data. These algorithms try to find associations among the items that exist in data points.
Apriori and PrefixSpan have been the basic approaches to find sequential patterns. Sequential information embedded in the data is an important aspect that may be explored in various applications. In this work, a recommendation system that explores the sequential information present in data for generating recommendations has been developed. Designing a recommendation system that considers sequential information is still an important problem that needs to be addressed. This type of recommendation system will help e-commerce websites develop a decision support system capable of capturing sequential information. Our proposed model considers both the content as well as sequence of a visit during clustering to form groups of users.

**PROPOSED WORK**

Generally, a pattern recognition-based recommender system comprises of two phases; the first phase is clustering followed by classification task. In the first phase, the system is provided with enough learning so that the classification accuracy of the system is quite high or at the desired level. After the system learns, it as a result generates a set of recommendations with appropriate rankings.

Fig. 1 shows the complete flow of our model:

![Flowchart](image)

**4.1 FUZZY C-MEAN CLUSTERING**

In our system, we first formed clusters to acquire knowledge about web users and the classification technique was used later for enhancing the learning capability and to generate recommendations. A web user might have multiple interests for which he needs to be put under multiple clusters.

Detailed information about fuzzy c-mean technique has been provided below:

In **fuzzy clustering** also referred to as **soft clustering**, data elements can belong to more than one cluster, and a membership level in associated with each element. These levels indicate the strength of the association occurred between that data element and a particular cluster. Fuzzy clustering is a process of assigning these membership levels, and then using them to allocate data elements to one or more clusters.

The Fuzzy C Means algorithm divides a finite collection of n elements \( X = \{x_1, \ldots, x_n\} \) into a collection of c fuzzy clusters with respect to a given criteria.

When given a finite set of data, a list of C cluster centers

---

**ISSN: 2455-2631**

© July 2020 IJSDR | Volume 5, Issue 7

---

IJSDR2007060 | International Journal of Scientific Development and Research (IJSDR) [www.ijsdr.org] | 438
\[ C = \{ c_1, \ldots, c_c \} \text{ and a partition matrix } \]

\[ W = \omega_{ij} \in [0,1], i = 1, n, j = 1, \ldots, c \]

where each element \( \omega_{ij} \) depicts the degree to which element \( X_i \) belongs to \( C_j \) cluster are returned by the algorithm. Similar to the K-means clustering, the FCM aims to minimize an objective function

\[
\min_{c} \arg \sum_{i=1}^{n} \sum_{j=1}^{c} \omega_{ij}^m || X_i - C_j ||^2, \\
\text{Where:}
\]

\[
\omega_{ij}^m = \frac{1}{\sum_{k=1}^{c} \left( ||X_i - C_k||^2 \right)^{\frac{1}{m-1}}}
\]

This difference from the \( k \)-means is objective function by the addition of the membership values \( \omega_{ij} \) and the fusilier \( m \in R \), with \( m \geq 1 \) . The fusilier \( m \) determines the level of cluster fuzziness. A large \( m \) results in smaller memberships \( \omega_{ij} \) and thus, fuzzier clusters. In the limit \( m = 1 \), the memberships \( \omega_{ij} \) converge to 0 or 1, which thus implies a crisp partitioning. In the absence of experimentation or domain knowledge, \( m \) is usually set to 2.

4.2 SELECTION ON TOP ‘N’ CLUSTERS

After applying fuzzy c-mean clustering, we get cluster centre values for each cluster upon which processing is done instead of entire data of users present in cluster.

In order to get our model work, we are required to find out cluster number in which each user occurs. Now since we are dealing with soft clusters we may get a number of cluster numbers for each user. That’s acceptable.

We are taking top ‘n’ clusters in our proposed work depending on the average value of cluster centre values. It comes from the fact that not all but significant clusters will have remarkable impact on recommendations.

4.3 FINDING USERS SIMILAR TO TARGET USER

We now choose our target user for whom we wish to make recommendations. After choosing target user we now need to find all users similar to target user as they would have effective impact on user’s recommendations.

In order to find similar users we used partition matrix ‘U’ obtained from fuzzy c-mean clustering algorithm used at the earlier phase of proposed work.

After we have got users similar to target user we generated resultant matrix MAT which contains all similar user entries.

4.4 WEIGHT CALCULATION FOR EACH PAGE CATEGORY

Since we have obtained set of similar users we are just left with calculation of weight for each page category. Higher the weight higher is the probability of the page to be recommended to the user.

In our proposed model we are having 17 page categories defined with ranking from 1 to 17. The data set has seventeen categories: “front page”, “news”, “tech”, “local”, “opinion”, “on-air”, “misc”, “weather”, “health”, “living”, “business”, “sports”, “summary”, “bbs” (bulletin board service), “travel”, “msn-news” and “msn-sports”.

For the purpose of weight calculation we maintain three matrices namely ‘X’, ‘Z’ and ‘G’.
- Matrix X contains count of number of times the target user has opened the specific page.
- Matrix Z contains count of number of times all the similar users of target user have opened the specific page.
- Matrix G contains whether a page has been opened by the target user at all or not.

Taking into consideration all these matrix values it can be understood that matrix X and matrix Z favours recommendation of a page to the user.

Corresponding equations are as follows:

\[
(1) \\
\text{if (msndata(user,i)==k)} \quad x(k)=x(k)+1; 
\]
(2)

\[
\text{if}(\text{mat}(i,j)==k) \\
\text{z}(k)=\text{z}(k)+1;
\]

(3)

\[
\text{if}(\text{mat}(i,j)==k) \\
\text{if}(g(k)==0) \\
g(k)=g(k)+1;
\]

(4)

Where K stands for page category from 1 to 17, matrix MAT contains all similar user entries of the target user and msndata refers to dataset taken for analysis.

At last when all the above three matrix values have been calculated the final weight of each page category can be calculated using formulae stated below:

\[
\text{WEIGHT}(i)=\frac{(X[i]+Z[i])}{(\text{size-G}[i])}
\]

(5)

Where i stands for page category and values used can be obtained using eq 2,3 and 4.

4.5 CALCULATION OF ACCURACY

Accuracy has been defined as the ratio of the number of correct recommendations to the number of total recommendations.

\[
\text{ACCURACY} = \frac{(\text{no of correct recommendations})}{(\text{no of total recommendations})}
\]

In this case, we recommend the next possible visit of the user, which is compared with the actual next step of the user (testing). If the predicted next step is the same of the actual next state, then the event is termed as hit, else it is termed as miss. Hence, accuracy is our case will be given as:

\[
\text{Accuracy} = \frac{(\text{no of hits})}{(\text{no of hits} + \text{no of miss})}
\]

(6)

Considering the confusion matrix Accuracy of the recommendation system can be defined as the ratio of relevant retrieved elements to all retrieved, non-retrieved, relevant and non relevant elements.

References