Movie Recommendation using K – Means Clustering and Collaborative filtering Algorithm

[1]S.R.Navneeth, [2]Mr. Arunachalam
[1]Research Scholar, Department of Computer Science Engineering, Bharath University, Chennai, Tamil Nadu

[2] Sr.Professor, Department of Computer Science Engineering, Bharath University, Chennai, Tamil Nadu

[1] corresponding author

Abstract: The paper presents k - means clustering algorithm used to find out the ranking from given user information available on social network web sites like orkut, facebook, twitter and collaborative filtering algorithm for reducing information overload. It is one of the simplest clustering algorithms. It is called k-means because it iteratively improves our partition of the data into k sets. Then find out the average of each dataset (data mining value).This algorithm is used to reduce the work complexity. When user entered new comments about movie, it will be automatically calculate average and display the ranking.

1. INTRODUCTION

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. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.The company encourages subscribers to "rate" the movies that they watch, expressing an opinion about how much they liked (or disliked) a film. The company's Cinematch recommendation system analyzes the accumulated movie ratings and uses them to make several hundreds of millions of personalized predictions to subscribers per day, each based on their particular tastes. Users select movies on the Netflix website, and their selections are mailed to them. Based on user viewing history, Netflix recommends other movies to the user. The algorithm Netflix uses based on what other people watched and liked, after watching the same movies. If user watch movie A and others who watch movie A also watch (and like!) movie B, Netflix recommend the user to watch movie B.

The social network is grooving size and number every day. The user maintains the personal information in social network. Automated collaborative filtering systems works by collecting information from social network for the users for an item in a given domain and matching together people who shares the same information needs or the same taste. In data mining, ***k*-means clustering** is a method of cluster analysis which aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean. This results in a partitioning of the data space into Voronoi cells.

The problem is computationally difficult (NP-hard), however there are efficient heuristic algorithms that are commonly employed and converge quickly to a local optimum. These are usually similar to the expectation-maximization algorithm for mixtures of Gaussian distributions via an iterative refinement approach employed by both algorithms. Additionally, they both use cluster centers to model the data, however *k*-means clustering tends to find clusters of comparable spatial extent, while the expectation-maximization mechanism allows clusters to have different shapes.

The outline of this paper is as follows. Section 2 looks at previous work in the area of movie recommendations and currently available recommendation services. Section 3 discusses how data is gathered and represented. Section 4 goes into details of our recommendation algorithms. Section 5 discusses our results and observations. Finally, Section 6 concludes this paper by discussing possible future extensions to our work.

2. PREVIOUS WORK

Most of the online movie recommendation engine such as yahoo and some other websites use collaborative _ltering to generate movie recommendations. This well work on given user had given the rating a large data set of the user's movie viewing history and ratings, which often involves _lling out lengthy surveys. The recommender first identify the item of inserted .The recommendation calculate in two ways, first remove unselect, second one calculated movie. In this process admin does manually so it takes more time and admin should take ranking from trust social network i: e. security social network. Existing work done from some basic recommendation algorithm, Users select movies on the Netflix website and their selections are mailed to them. Based on viewing history, Netflix recommends other movies to user. The algorithm Netflix is used based on the other people watched and liked, after watching the same movies.

If user watch movie A, and others who watch movie A also watch (and like!) movie B, Netflix recommend to watch movie B. The problem with this algorithm is that it is not scalable to large sizes: as k gets large, each of the (large number of) data points must be compared to k different possible centers. So we use another, faster, process to partition the data set into reasonable subsets: Canopy clustering.

The recommender system is used to identify the inserted movie, that recommendation system generally calculating the ranking the movie from social network by two ways. First one similarly selected and second one is mostly liked move, we are taking from social network. The recommendation algorithm is used to filter move from data mining. The cup focus from different data set.

3. DATA GATHERING AND REPRESENTATION

3.1. Data Gathering:

The social network collects the information from user (client) and his friends. The gathering means more than information store in one place that place was data mining. This paper include the k-means algorithm, it calculate the rating from user information. The use information means consider the age, sex, and user commands. we consider each movie to be a separate problem requiring its own classifier. For any given movie m, our training data thus consists of all users who have rated m; more specifically, each training instance corresponds to a distinct user. This user’s feature vector contains the ratings given by that user to every other movie in the dataset (0 if not seen). In addition, each movie rating is accompanied by a binary feature that indicates whether the user has seen the movie. Thus, supposing there are M movies in the database, a single feature vector contains (2M-1) features (including the intercept term), since we do not include the rating of the movie we are classifying as a feature. The corresponding label for each training instance is simply the rating given to movie m by this user.

Additional details on feature definition, including normalization and scaling, are detailed in the section 5. First, however, we discuss the two types of logistic regression used in our work. The Movie survey was conducted using the Consensus tool to guarantee anonymity. While this tool was excellent for providing a simple user interface and robust management, the schema used to store the data was not appropriate for conducting any sort of analysis on what we gathered. The data was in the form of “SurveyID,” “QuestionID,” “Answer”, and the Movies, Actors, and Directors fields were simply long strings of text.

To mine the data, we first needed to transform the data into a format that was more suitable. What we needed was a table for each multiple answer question (such as Hobbits), plus a table for all of the single answer questions with a single row for each respondent. Additionally, we needed to parse out the individual movies, actors, and directors from the text fields.

To accomplish this task, we leveraged the power of Yukon Data Transformation Services. Yukon DTS allowed us to easily split, convert, parse and pivot the data gathered by Consensus into the eight tables we needed to perform our data mining task. Here is an image of the pipeline task (dubbed the “Octopus”) that performed most of this work.

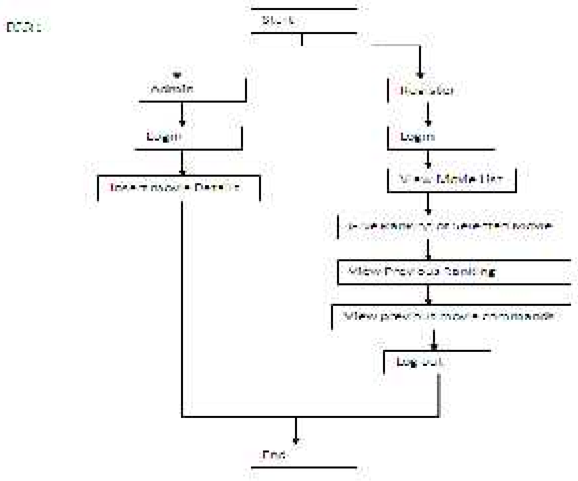

Fig.3.1: Architecture of Movie Ranking

3.2 Quantitative and Qualitative Data collection methods:

The Quantitative data collection methods rely on random sampling and structured data collection instruments that fit diverse experiences into predetermined response categories. They produce results that are easy to summarize, compare, and generalize. Quantitative research is concerned with testing hypotheses derived from theory and/or being able to estimate the size of a phenomenon of interest. Depending on the research question, participants may be randomly assigned to different treatments. If this is not feasible, the researcher may collect data on participant and situational characteristics in order to statistically control for their influence on the dependent, or outcome, variable. If the intent is to generalize from the research participants to a larger population, the researcher will employ probability sampling to select participants.

Typical quantitative data gathering strategies include:

- Experiments/clinical trials.
- Observing and recording well-defined events (e.g., counting the number of patients waiting in emergency at specified times of the day).
- Obtaining relevant data from management information systems.
- Administering surveys with closed-ended questions (e.g., face-to face and telephone interviews, questionnaires etc).

4. RECOMMENDATION ALGORITHMS

4.1 K-Means Clustering:

In data mining, ***k*-means clustering** is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. This result in a partitioning of the data space into cells. The problem is computationally difficult (NP-hard), however there are efficient heuristic algorithms that are commonly employed and converge quickly to a local optimum. These are usually similar to the expectation-maximization algorithm for mixtures of Gaussian distributions via an iterative refinement approach employed by both algorithms. Additionally, they both use cluster centers to model the data, however k-means clustering tends to find clusters of comparable spatial extent, while the expectation-maximization mechanism allows clusters to have different shapes. Given a set of observations ($\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$), where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k sets ($k \leq n$) $\mathbf{S} = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares (WCSS):

This nonhierarchical method initially takes the number of components of the population equal to the final required number of clusters. In this step itself the final required number of clusters is chosen such that the points are mutually farthest apart. Next, it examines each component in the population and assigns it to one of the clusters depending on the minimum distance. The centroid position is recalculated every time a component is added to the cluster and this continues until all the components are grouped into the final required number of clusters.

$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \left\| \mathbf{x}_j - \boldsymbol{\mu}_i \right\|^2$$

where μi is the mean of points in S_i.

4.2 STANDARD ALGORITHM

The most common algorithm uses an iterative refinement technique. Due to its ubiquity it is often called the k-means algorithm; it is also referred to as Lloyd's algorithm, particularly in the computer science community. Given an initial set of k means $m_1^{(1)}, \ldots, m_k^{(1)}$ (see below), the algorithm proceeds by alternating between two steps.

Assignment step: Assign each observation to the cluster whose mean is closest to it (i.e. partition the observations according to the Voronoi diagram generated by the means).

$$S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\| \le \|x_p - m_j^{(t)}\| \ \forall\ 1 \le j \le k\}.$$

Where each $x_p$is assigned to exactly one$S^{(t)}$, even if it could be is assigned to two or more of them.

Update step: Calculate the new means to be the centroids of the observations in the new clusters.

$$\mathbf{m}_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

The algorithm has converged when the assignments no longer change. Commonly used initialization methods are Forgy and Random Partition. The Forgy method randomly chooses *k* observations from the data set and uses these as the initial means. The Random Partition method first randomly assigns a cluster to each observation and then proceeds to the update step, thus computing the initial mean to be the centroid of the cluster's randomly assigned points. The Forgy method tends to spread the initial means out, while Random Partition places all of them close to the center of the data set. According to Hamerly et all the Random Partition method is generally preferable for algorithms such as the *k*-harmonic means and fuzzy *k*-means. For expectation maximization and standard *k*-means algorithms, the Forgy method of initialization is preferable.

5. RESULT AND EVALUATION

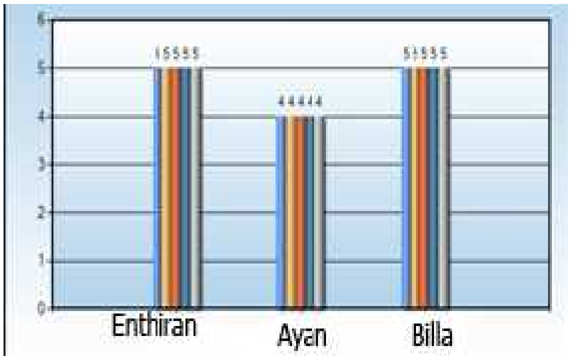

Fig.5.1: Individual Rating Chart

The graph generated based on user commands shown in fig.5.1. Suppose we would like to give commands for movie list first register your details after that login personal network in social network after enter in to the social network user should give commands on movies based on direction, story, music.

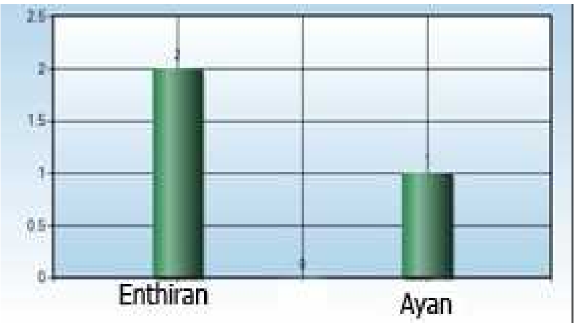

Fig.5.3. Overall Rating Chart

The above graph shown in fig.3.generated based on users over all commands. K-means clustering algorithm is used to display the ranking details based on users different commands.

6. CONCLUSION

Every user requirements are different from others. The project is based on users expectations Here k-means clustering algorithm is used for accurate movie recommendation information in social network with input information. Suppose need information in social network, user should give input information and should get output information from data mining. This output information was generated automatically based on user requirements. The ranking generation based on users commends like sex, age, rating etc.,

7. REFERENCES

[1] Ankit Gupta, Rohan Jain "Movie Recommendations Using Social Networks" Dec 12, 2008.

[2]. Navjot Kaur, Jaspreet Kaur Sahiwal, Navneet Kaur "Efficient k-means clustering algorithm using ranking method in data mining" international Journal of Advanced Research in Computer Engineering & Technology Volume 1, Issue 3, May2012.

[3] Bennett, James, and Stan Lanning. "The Netflix Prize." Proceedings of KDD Cup and Workshop (2007).

[4] Sa, Brian, and Patrick Shih. "K-Means for Netflix User Clustering." CS229 (2006).

[5] Sadovsky, Adam, and Xing Chen. "Evaluating the Effectiveness of Regularized Logistic Regression for the Netflix Movie Rating Prediction Task." CS229 (2006).

[6] Jennifer Golbeck. ”Generating Predictive Movie Recommendations from Trust in Social Net-works.” Proceedings of the Fourth International Conference on Trust Management (2006).

[7] T. Joachims. ”Optimizing Search Engines Us-ing Clickthrough Data.” Proceedings of the ACMConference on Knowledge Discovery and Data Mining (KDD), ACM (2002).

[8] Risi Imre Kondor, John Lafferty. ”Diffusion Ker-nels on Graphs and Other Discrete Input Spaces.”Proceedings of the Nineteenth International Con-ference on Machine Learning (2002).

[9]. XindongWu · Vipin Kumar · J. Ross Quinlan · Joydeep Ghosh “Top 10 algorithms in data mining”(2007).

[10].Jennifer Golbeck “Generating Predictive Movie Recommendations from Trust in Social Networks”(2007)

[11].http://www.stanford.edu/class/cs229/proj/RaksheKumar-L1RegularizedLogisticRegression.pdf

[12].Herlocker, J, Konstan, J., Terveen, L., and Riedl, J. Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems22 (2004), ACM Press, 5-53.

[13] Resnick, P. Varian, H.R., Recommender Systems, Communications of the ACM40 (1997), 56-58.

[14]. Abdul-Rahman, A. and Hailes, S. 2000. Supporting trust in virtual communities. In Proceedings of the 33rd Hawaii International Conference on System Sciences. Maui, HW, USA.

[15].Ziegler, Cai-Nicolas, Georg Lausen (2004) Analyzing Correlation Between Trust and User Similarity in Online Communities" Proceedings of Second International Conference on Trust Management, 2004.

[16]. Sinha, R., and Swearingen, K. (2001) "Comparing recommendationsmade by online systems and friends." In Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries Dublin, Ireland.

[17].Swearingen, K. and R. Sinha. (2001) "Beyond algorithms: An HCI perspective on recommender systems," Proceedings of the ACM SIGIR 2001 Workshop on Recommender Systems, New Orleans, Louisiana.

[18].Herlocker , Jonathan L., Joseph A. Konstan , John Riedl, Explaining collaborative filtering recommendations, Proceedings of the 2000 ACM conference on Computer supported cooperative work, p.241-250,December 2000, Philadelphia, Pennsylvania, United States.

[19].Garden, Matthew, and Gregory Dudek (2005) Semantic feedback for hybrid recommendations in Recommendz. Proceedings of the IEEE International Conference on e-Technology, e-Commerce, and e-Service (EEE05), Hong Kong, China, March 2005.

[21]. Perny, P. and J. D. Zucker. Preference-based Search and Machine Learning for Collaborative Filtering: the ``Film-Conseil" recommender system. Information, Interaction , Intelligence, 1(1):9-48, 2001.

[22]. Herlocker , Jonathan L., Joseph A. Konstan , Loren G. Terveen , John T.Riedl, (2004) Evaluating collaborative filtering recommender systems,ACM Transactions on Information Systems (TOIS), v.22 n.1, p.5-53,January 2004.

[23]. Massa P., P. Avesani. 2004. Trust-aware Collaborative Filtering for Recommender Systems. In Proceedings of the International Conference on Cooperative Information Systems (CoopIS) 2004.

[24]. Massa, P., B. Bhattacharjee. 2004. Using Trust in Recommender Systems: an Experimental Analysis. In Proceedings of iTrust2004 International Conference.

[25].Golbeck, Jennifer. 2005. Computing and Applying Trust in Web-Based Social Networks, Ph.D. Dissertation, University of Maryland, College Park.

[26].Golbeck, Jennifer. 2005. Personalizing Applications through Integration of Inferred Trust Values in Semantic Web-Based Social Networks.Proceedings of Semantic Network Analysis Workshop. Galway, Ireland.