ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Indian Journal of Science and Technology, *Vol 9(37), DOI: 10.17485/ijst/2016/v9i37/102074, October 2016*

Collaborative Filtering using Euclidean Distance in Recommendation Engine

A. Jeyasekar*, K. Akshay and Karan

Department of Computer Science and Engineering, Kattankulathur, Chennai - 603203, Tamil Nadu, India;
ajeyasekar@yahoo.com, akshaykommadath@gmail.com, karan.2.10.1993@gmail.com

Abstract

Objectives: Recommendation engine is a part of information filtering system that tries to predict the 'preference' or 'rating' of an item in the E-commerce. Recommendation engines have become extremely common in recent days to make an appropriate recommendation rapidly and effectively about any products on which the user is interested. **Methods/ Statistical Analysis**: One of popular information filtering systems in the recommendation engine is collaborative filtering where the predictions are made based on the usage patterns of the users who are similar to another user. The accuracy of a recommendation engine using collaborative filtering depends on the techniques used to measure the similarity between the user's preferences. Therefore, in this paper we use two metrics to measure the similarity between the user's preferences namely KL Divergences and Euclidean distance. The proposed algorithm works by first clustering the users using k means clustering by utilising the similarity metrics and then computing the global Markov matrix for that cluster. Next, the PageRank value for each user is computed and those values are combined with the global Markov matrix to find the recommendations. **Findings:** We consider the problem of collaborative filtering to recommend potential items of interest to a user already engaged in a session, using past session of the user and other users. Our algorithm leads to the personalized PageRank, where context is captured by the personalization vector. The results show that the collaborative filtering using Euclidean distance metrics for similarity measure performs well than the KL divergence. **Application/ Improvements:** The proposed recommendation engine can be used in a wide variety of applications such music, movies, books, news, research articles, social media, search queries, and products in general in order to provide a effective recommendation.

Keywords: Collaborative Filtering, Euclidean Distance, Jaccard Metric, Recommendation Engine, Similarity Metrics

1. Introduction

A recommendation system, or recommender system tries to make predictions on user preferences and make recommendations which should interest customers. Recommendation systems typically appear on many e-commerce sites because of providing better conversion rates. In the Internet, 35% of Amazon's sales are result of its recommendation engine. Additively, there are many other usages of them such as recommending music, books, movies, or even articles. There are basically two approaches to make recommendations: Collaborative filtering (or social filtering) Content-based filtering Collaborative filtering uses the known behaviour of a group of users to make recommendations for the others. For instance, one can predict whether a particular user will like an item-A using the similar preferences/recommendations of other users[1,2]. The second common approach is content-based filtering that makes use of the comparison between items and the past preferences of a particular user. That is, the items which have similar properties that the user liked or checked previously are likely to be recommended[3].

Suggesting/recommending an item to a user from the given set of items based on the other user's interest/preferences is very difficult. To do so, the similarity between the user's interests is estimated using similarity measure metrics like KL divergence, Euclidean distance, Cosine

*Author for correspondence

distance, Pearson metric etc. The accuracy of recommendation to a user depends on the similarity measure.

Therefore, in this paper, we analyze the two similarity measure metrics namely KL Divergence and Euclidean distance and found that Euclidean distance bases collaborative filtering performs well as compared with KL Divergences.

2. Related Work

There are many types of commonly used algorithms that are used in recommendation engines. In content based filtering, items which are similar to those liked in the past are recommended. Attributes of preferences of a user are matched with the item attributes to make recommendations. Content based recommenders can rate items that are yet to be rated by a user and there is no first rate problem which occurs in collaborative filtering techniques. However, this technique suffers from the drawback that without sufficient domain knowledge, suitable suggestions cannot be made. Overspecialization is another drawback where only the items rated highly against user profile can be recommended[4].

Collaborative filtering is an algorithm that utilises the usage patterns of other similar users in the system. This method works by collecting preferences from other users and combining to provide a fairly personalised recommendation to the active user[5]. Collaborative filtering is categorised into two types: Item based and memory based collaborative filtering. Item based collaborative filtering techniques, user item matrix is first analysed to establish a relationship and then recommended items based on the analysed relationships. It is established that item based algorithm provide better quality than user based algorithms and helps the collaborative filtering to produce high quality recommendation and scale to large data sets[6]. The motivation for memory-based CF comes from the observation that people usually trust the recommendations from like-minded users. In such techniques ratings of similar users are used to predict a user's rating by applying a nearest-neighbour-like scheme. A disadvantage of memory based system is slow response time because the whole database has to be searched for making a single recommendation. Using probabilistic techniques, the efficiency and accuracy is increased[5].

In hybrid approach, both content-based filtering and collaborative filtering are combined together to recommend an item to the user which helps to avoid certain limitations of content-based and collaborative filtering. The content-based and collaborative filter are implemented separately and their predictions are combined for recommending an item to user.

Therefore, in this paper we propose a recommendation algorithm that makes use of clustering using k-mean algorithm to group the users into separate cluster. The accuracy of k-means algorithm lies on the distance metrics used like Euclidean distance, KL divergence[7]. We use both metrics and find that k-mean algorithm using Euclidean distance metrics outperforms KL Divergence metric.

3. Collaborative Filtering

In this paper, we recommend application sequences to the user by utilising collaborative filtering technique. The application sequences for a user u, containing item *i*, is known as a user session $s=<i_1,i_2,i_3,\ldots,i_n>$. The algorithm employs the session data to arrive at personalised recommendation for a user. Since our problem is to recommend sequences, the user session is represented as a Markov transition probability. The different stages in the algorithm are clustering, finding the global markov matrix of a cluster, page-rank computation and computation of user scores and ranking the recommendation.

3.1 Clustering

The clustering stage is the first stage of the algorithm where the users are grouped into clusters based on how similar the users are to each other. The k- means algorithm is applied to all the users. In the k means algorithm, based on the number of clusters that are required the same number of random centroids are generated. The similarity metric or the distance metric is the most important parameter that is applied that ultimately decide the accuracy of the recommendations[8,9].

We use the Euclidean distance or the KL divergence to find the distance between two users (x,y). The Euclidean distance[10] is calculated as given below

$$||X\text{-}Y||_{EUC} = \sqrt{}\Sigma^{m}_{i=1}\Sigma^{n}_{j=1}|x_{ij} - y_{ij}|^2 \qquad (1)$$

And the KL divergence is computed as given below

$$d_{KL}(x,y) = \Sigma^{p}_{i=1}x_i\log_2(x_i/y_i) \qquad (2)$$

The distance of all the user matrices is compared to all the three randomly generated centroids and each user is assigned to the cluster which is very close to. For getting accurate results from clustering, the clustering stage is iterated until a stable cluster is obtained. Stability here means that the users stop moving from one cluster to another. Once a stable cluster is obtained, the medians are recomputed by using the Voronoi iteration method. This is done by first comparing the all the points one at a time to all other points within a cluster and summing up the distance values. The new centroid will be selected as the one which has the least sum of distances value. This process is repeated for all the clusters and medians are recomputed for each.

3.2 Finding the Global Markov Matrix of a Cluster

Although a global Markov matrix for all the users as a whole does not give personalized recommendations by applying this global Markov matrix to a cluster which contains all similar users helps to achieve a certain level of personalization. The cluster to which the user belongs is given by π(u), and the global matrix is given by $M_{\pi}^{(u)}$ for user u. Therefore after the clustering is done, a global Markov transition matrix is obtained by analysing the sequence of transitions of all users in the cluster and combining the probabilities of transition into the matrix.

3.3 PageRank Computation

The PageRank is an algorithm that ranks the transitions and gives weight age to each transition[4]. This is applied to a single user to obtain the ranking matrix of that user. Let π(u) denote the cluster to which the user u belongs. The normalized indicator vector c_u is first computed for a user for items appearing in the session s of that user. Then the personalized Page Rank matrix, which is used to personalize the PageRank algorithm for individual users is computed as given below

$$Z_u = (\alpha M + (1\text{-}\alpha)1c_u)^t z_u \qquad (3)$$

Here z_u is the Page Rank matrix of a user u. This is initially computed by using both the global matrix for a cluster, $M_{\pi}^{(u)}$. The z matrix is initialized to one divided by the number of transitions in each row which denotes the transitions that is possible from that item since reach row denotes one item transition state. 1 denotes the matrix of all ones. α is the limiting parameter that is set to 0.85. This [11] value of alpha produces the most accurate results and this value is also used in other implementations that use Page Rank.

3.4 Computation of User Scores

The next stage in the algorithm is the computation of final scores of transitions. This scoring function for the user u is obtained by adding the PageRank matrix computed in the previous stage and the matrix, $M_{\pi}^{(u)}$

$$F_u = z_u + M_{\pi}^{(u)} \qquad (4)$$

Here z_u is the PageRank matrix of a user u.

3.5 Ranking and Producing Recommendations

The matrix f that is obtained will be a square matrix will be have as many rows and columns as the number of items. For each row which is an item transition to any other item, there can be more than one value in a row that is greater than zero. The final recommendation of an item for a particular item is the one which has the highest ranking among all other items in that row.

In this algorithm described the clustering and computing the global Page Rank matrix can have computed offline but the other steps have to be executed every time the user updates the recommendations

4. Performance Analysis

In this section we are going to compare the performance of the recommendation algorithm using Kl divergence and Euclidean distance on our application sequence dataset.

4.1 Dataset Description

The dataset that is used in this paper consists of the application usage patterns of more than 100 users. The users were asked to give their preferred sequence from a group of application. This was done to overcome the cold start problem that many recommendation engines face. The cold start problem is when a new entity comes into the system and makes it hard to make recommend items due to lack of information about this new entity.

4.2 Results

The Performance of both the distance metrics is compared and the results suggest that although in most of the cases both the metrics are similar, Euclidean distance metric in collaborative filtering gives better clustering and similarity between the users in other cases.

In this comparison, we take a sample set of 100 users, 6 application sequences and a fixed alpha value of 0.85 and we obtain similarity values plotted on a scale of 0 to 6 from the least similar to the most similar sequence when compared with the original sequence.

In Figure 1, the most of the points are closely packed towards the bottom of the scatter plot which indicates that the recommended sequences have a similarity value in the bottom range. The mean value obtained while using KL divergence is 1.64.

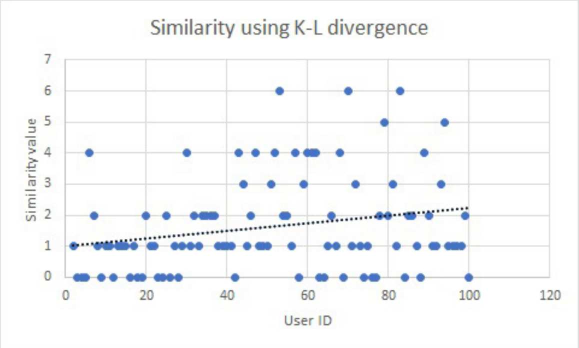

Figure 1. Scatter plot of using KL-Divergence as the similarity metric for all users.

In comparison to the scatter plot shown Figure 1, it is observed that the points shown in Figure 2 are more distributed across the whole plot. The linear plot starts from the value 2 and has a mean similarity value of 2.56.

Therefore, Euclidean distance has a better performance than KL divergence. Hence in this paper, collaborative filtering using Euclidean distance is used in the recommendation engine.

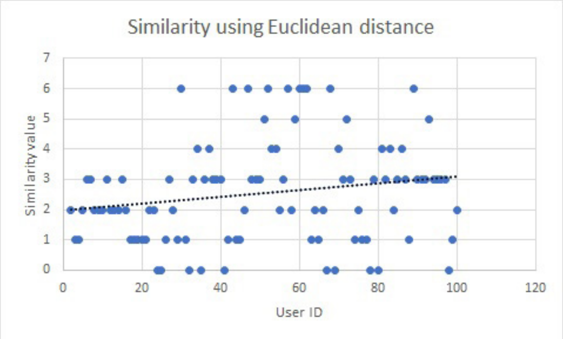

Figure 2. Scatter plot of using Euclidean Distance as the similarity metric for all users.

5. Conclusion

In this project we consider the problem of collaborative filtering to recommend potential items of interest to a user already engaged in a session, using past session of the user and other users. Our algorithm leads to the personalized PageRank, where context is captured by the personalization vector. The results on real-life datasets demonstrate that the proposed recommendation engine achieves a good recommendation performance illustrating its ability to capture the context of a given session.

As part of future work, different types of techniques such as spearman correlation, cosine similarity, Pearson correlation, etc can be used to find the similarity between the matrices, so that, the clustering can be improved and better recommendation could be achieved. Also, we are exploring other collaborative and context filtering methods to incorporate interactional context so that better recommendations could be achieved.

6. References

1. Reddy CA, Subramaniyaswamy V. An enhanced travel package recommendation system based on location dependent social data. Indian Journal of Science and Technology. 2015 Jul; 8(16):1–7.
2. Sarwar B, Karypis G, Konstan J, Riedl J. Item-based collaborative filtering recommendation algorithms. Proceedings of 10th International Conference on World Wide Web, New York; 2001. p. 285–95.
3. Konstan JA. Introduction to recommender systems. ACM Transactions on Information Systems. 2004; 22(1):5–53.
4. Lops P, Gemmis MD, Semeraro G. Content-based recommender systems: State of the art and trends. Recommender System Handbook, Springer Publisher; 2011. p. 73–105.
5. Ekstrand MD, Riedl JT, Konstan JA. Collaborative filtering recommender systems. Foundations and Trends in Human-Computer Interaction. 2011; 4(2):81–173.
6. Yu K, Schwaighofer A, Tresp V, Xu X, Kriegel HP. Probabilistic memory-based collaborative filtering. IEEE Transactions of Knowledge and Data Engineering. 2004; 16(1):56–69.
7. Natarajan N, Shin D, Dhillon IS. Which app will you use next? Collaborative Filtering with Interactional Context. Proceedings of 7th ACM Conference on Recommender Systems, New York; 2013. p. 201–8.
8. Parimala M, Lopez D, Kaspar S. K-neighbourhood structural similarity approach for spatial clustering. Indian Journal of Science and Technology. 2015; Sep 8(23):1–11.

9. Devi DMRP, Thambidurai T. Similarity measurement in recent biased time series databases using different clustering methods. Indian Journal of Science and Technology. 2014; 7(2):189–98.
10. Singh A, Yadav A, Rana A. K-means with three different distance metrics. International Journal of Computer Applications. 2013; 63(11):1–5.
11. Brin S, Page L. The anatomy of a large-scale hypertextual web search engine. Proceedings of 7th International Conference on World Wide Web, Computer Networks and ISDN Systems. 1998; 30(7):107–17.