Improving Movie Recommendation Systems Filtering by Exploiting User-Based Reviews and Movie Synopses

Konstantina Iliopoulou, Andreas Kanavos, Aristidis Ilias, Christos Makris, Gerasimos Vonitsanos

Improving Movie Recommendation Systems Filtering by Exploiting User-based Reviews and Movie Synopses

Konstantina Iliopoulou, Andreas Kanavos, Aristidis Ilias,
Christos Makris and Gerasimos Vonitsanos

Computer Engineering and Informatics Department
University of Patras, Patras, Greece
k.iliopoulou1@gmail.com, {kanavos, aristeid, makri, mvonitsanos}@ceid.upatras.gr

Abstract. This paper addresses the subject of Movie Recommendation Systems, focusing on two of the most well-known filtering techniques, Collaborative Filtering and Content-based Filtering. The first approach proposes a supervised probabilistic Bayesian model that forms recommendations based on the previous evaluations of other movies the user has watched. The second approach composes an unsupervised learning technique that forms clusters of users, using the K-Means algorithm, based on their preference of different movie genres, as it is expressed through their ratings. Both of the above approaches are compared to each other as well as to a basic method known as Weighted Sum, which makes predictions based on the cosine similarity and the euclidean distance between users and movies. In addition, Content-based Filtering is implemented through K-Means clustering techniques that focus on identifying the resemblance between movie plots. The first approach clusters movies according to the Tf/Idf weighting scheme, applying weights to the terms of movie plots. The latter identifies the likeness between movie plots, utilizing the BM25 algorithm. The efficiency of the above methods is calculated through the Accuracy metric.

Keywords: Recommendation Systems, Movie Recommendation Systems, Collaborative Filtering, Content-based Filtering, Text Analysis

1 Introduction

Rapid Internet growth continually creates an immense amount of data, as well as the need to find more productive ways to handle it. Users daily have to face the process of choosing between overwhelming varieties of different products during their interaction with any digital platform, a pursuit often tiring and disorienting.

As one of the most popular research issues, one can consider the subject of improving the quality of ranking in Information Retrieval results. To this extent, information need is expressed through the form of queries submitted to a search engine or platform with the intention of receiving any available fact regarding

the inquiry [1,11]. Effective retrieval techniques and methodologies have been mostly derived from the class of probabilistic models, and several approaches have been successfully implemented in this direction [1,3].

Recommendation Systems provide a sound solution for the information flood each user has to face daily during their interaction with online platforms. Aiming to provide users with the ability to find products that may be of their interest, make up the target of these systems in an automatic, fast and efficient way. Their recommendations are based on the analysis of previous user behavior, with respect to the evaluation of products as well as the recognition of the similarity between different users and products. This occurs, in order to derive a prediction of which products a user will consider interesting. Due to their wide-ranging scope of application, the research community takes an active interest in the field.

This paper addresses the subject of Movie Recommendation Systems, focusing on two of the most well-known filtering techniques, Collaborative Filtering, and Content-based Filtering. Specifically, Collaborative Filtering is implemented through two different approaches. The first approach proposes a supervised probabilistic Bayesian model that forms recommendations based on the estimation of the probability a user gives a specific rating to some movie, by examining either the rating given to that movie by the rest of the users or the rating the user gave to other movies he/she has watched. The second approach is an unsupervised learning technique that forms clusters of users using the K-Means algorithm, based on their preference of different movie genres, as it is expressed through their ratings. Both of the above approaches are compared to each other and also to a basic method known as Weighted Sum that makes predictions based on the cosine similarity and the euclidean distance between users and movies. Content-based Filtering is implemented through K-Means clustering techniques that focus on identifying the similarity between movie plots. The first approach clusters movies according to the Tf/Idf weighting scheme, applying weights to the terms of movie plots. The second approach identifies the similarity between movie plots utilizing the BM25 algorithm.

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 overviews the basic concepts and algorithms used in this paper. Section 4 and 5 detail the implementation and evaluation respectively. Finally, in Section 6, our concluding remarks and future work are presented.

2 Related Work

In [4], authors depict the setup for an opinion-based entity ranking system. The intuition behind their work is that each entity can be represented by all the review texts and that the users of such a system can determine their preferences on several attributes during the evaluation process. Thus, we can expect that a user's query would consist of preferences on multiple attributes. Authors in [10] further improved the setup by developing schemes, which take into account sentiment and clustering information about the opinions expressed in reviews;

also authors propose the naive consumer model as an unsupervised schema that utilizes information from the web to yield a weight of importance to each of the features used for evaluating the entities.

In addition, in [8], the authors propose a novel probabilistic network scheme that employs a topic identification method, in order to modify the ranking of results as the users select documents. Also, a novel framework aiming to provide the necessary tools for the refinement of search results, taking into account feedback provided by the user, is introduced in [7]. More specifically, the corresponding approach examines query-independent document processing and representation resulting in a lexical based inter-document similarity, that allows clusters to be formed.

Bayesian networks are known to perform well in the recommendation problem, as shown in the studies of [6,13,15]. The main difference between what we propose and the research in [13] is that we aim to predict the actual rating given by a user on a scale of 1-5, and not simply predict whether users will like or dislike an item. We can clearly observe that the authors of this work propose a hierarchical, Bayesian, content-based approach that aims to improve the current recommendation systems by incorporating context-related information when building the user profiles. Similar is the work presented in [15], where a collaborative filtering technique based on a simple Bayesian classifier as a solution to the recommendation problem, is introduced. Related to the above contributions is [6], as authors propose a Bayesian model that incorporates user preference information expressed in their reviews; the research uses collaborative techniques and topic modeling to solve the recommendation problem.

Furthermore, K-Means algorithm, being one of the most applicable clustering algorithms, often fits in the recommendation scheme. The research presented in [16] aims to improve the scalability of collaborative filtering recommendation systems by exploiting the bisecting K-Means clustering algorithm. The key idea is to apply the clustering algorithm in the user-item matrix and utilize the formed clusters as user neighborhoods. In order to form predictions for a specific user, the system implements the collaborative filtering algorithm by examining only the users' neighborhood. The approach in [2] exploits K-Means clustering, in order to form user clusters based on their ratings and a softmax regression classifier to predict the cluster each user belongs to. What is aimed here, is for the system to recommend the highest-rated movies from that cluster. In addition, the study in [9] discusses the clustering of similar documents by using the K-Means clustering algorithm and the Tf/Idf representation of the documents.

The research conducted in [12] centers on rating-based collaborative filtering. Its onus is to build user profiles and predict all missing values by developing a generative latent variable model, which extends already existing models such as the multinomial mixture model and LDA and is called the User Rating Profile model (URP). As a previous work on opinion clustering emerging in reviews, one can consider the setup presented in [5]. Authors propose a probabilistic network scheme, e.g., inference network, which consists of four component levels,

in following takes as input the belief of the user for each query (initially, all entities are equivalent) and produces a new ranking for the entities as output.

The work presented in [14] proposes a scheme for user clustering so as to identify similar tastes in movies; also, the clustering techniques are prone to discovering relationships between movie plots and movie genres. Our work however differentiates from this study since we focus on capturing similarities between movie plots regardless of the genre they ultimately belong to.

3 Preliminaries

The representation of a text in a comprehensible form for data mining and text pre-processing is the first important and critical task, considering tokenization, stop-words removal, punctuation and number removal, POS tagging, lower case as well as stemming. Text representation is based on the assumption that any text is described through its constituent words and is essential about replacing words with a numerical value, making the text editable by standard methods of analysis.

3.1 Text Representation

There are two main categories of text representation:

- Tuples, where the text is represented by a plurality of fields. The number of fields is equal to the size of the vocabulary and each field corresponds to a different word.
- Vector, where the text is represented by a vector. Each different word in the vocabulary corresponds to a component of the vector and defines one of the dimensions of the vector space. Representing a text document using vectors, is the most common approach, as it allows for more efficient calculations.

3.2 Tf/Idf

Tf/Idf constitutes one of the most popular algorithms to weigh a keyword in any content, assigning the importance to that keyword based on the combination of term frequency (Tf) and inverted document frequency (Idf) weights. It developed for Information Retrieval, however it is also widely used in Data Mining in combination with classification and clustering algorithms.

Specifically, given a term k_i of the document d_j, the weight w_{ij} defined by the Tf/Idf, is denoted as $Tf/Idf_{i,j}$ so as to apply:

$$Tf/Idf_{i,j} = Tf_{i,j} \times Idf_i \tag{1}$$

3.3 BM25

The word retrieval function BM25 is known due to ranking sets of documents based on query terms appearing in the documents of interest, regardless of their proximity within the document. Specifically, BM is a family of scoring functions with slightly different components and parameters, while various variants of the basic equation of the BM25 are encountered. In our work, a variant of the basic equation of BM25 was used, in which Idf is normalized to avoid negative values of the ranking function.

3.4 Principal Component Analysis

Principal Component Analysis (PCA) is one of the most commonly used methods for dimensionality reduction and feature extraction. It creates uncorrelated linear combinations of the original possibly correlated variables, and by utilizing the eigenvalues and eigenvectors of the variance/covariance matrix, it projects the data on a space of different dimensions transforming them in a way they can be adequately described by fewer dimensions.

3.5 Similarity Metrics

In this paper, vectors represent users or movies, and the similarity is expressed by the distance of these vectors. The two metrics used in the movie recommendation system are Cosine Similarity, which measures the similarities between two vectors, calculating its cosine of the angle between them, and Euclidean Distance, which expresses the similarity of two vectors, estimating the Euclidean distance between them. The similarity between all users or movies in the dataset can be represented by an array whose rows and columns correspond to users or movies. The value of each cell is the similarity between the elements in the corresponding row and column.

3.6 User Based Collaborative Filtering

User-based collaborative filtering makes predictions and suggestions based on the similarities among users. The movies suggested to the user are similar to those users have found interesting, that is, they have rated them highly. For example, assuming two users (U_A and U_B) who have seen the same movies and have both scored them with high ratings, as well as a new movie that U_A is watching and enjoying, the question will be: will this movie be suggested to U_B? A positive response occurs; judging by the past behavior of the two users, it is concluded that they have similar preferences.

3.7 Item Based Collaborative Filtering

Item-based collaborative filtering implements predictions and suggestions based on similarity among movies. The movies suggested to the user are similar to the

ones they have found interesting in the past. The similarity among movies arises by examining the scores received by other users of the system. For example, let us assume three users U_A, U_B and U_C who have seen two movies M_1 and M_2 and have rated them with roughly the same ratings, which makes these movies quite alike. If a fourth user U_D watches the first movie M_1 and scores it high the system will also propose the second movie M_2 to this viewer.

3.8 Weighted Sum

The simplest way of predicting user ratings is known as Weighted Sum and results from the inner product of the user-movie matrix along with the similarity matrix. User preferences can be modeled by a MxN matrix, known as user-item matrix, where M is the total number of users and N is the total number of items available in the system. In this paper, movies are considered items, each cell contains the rating of a specific user for a concrete movie (or is empty in case that they have not provided with an evaluation). Therefore, each user is represented by a vector, whose components are the ratings of the movies they evaluated. Similarly, movies are represented by vectors containing the ratings assigned to them by the users. The aim is to fill the missing values of the matrix with the predicted ratings.

4 Implementation

4.1 Dataset

The dataset used in this study is called ml-latest-small and is available on the MovieLens research site run by GroupLens Research at the University of Minnesota[1]. It consists of 9742 movies and 610 users, and describes the rating given to the movies by the users on a scale of 1 to 5. Each user has rated at least 20 movies, and each movie is rated by at least one user.

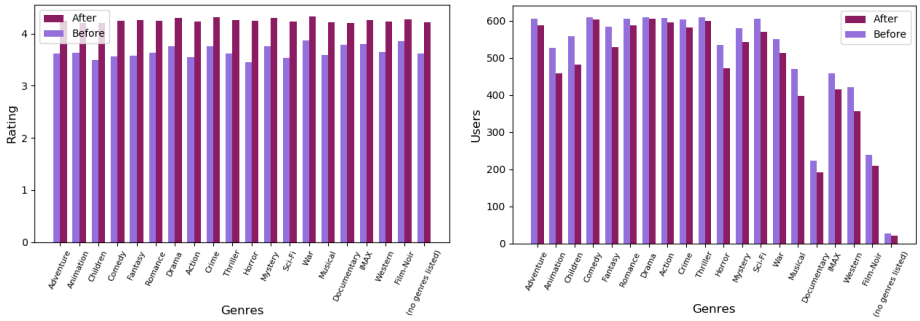
The dataset was extended in order to include also the movie plots by using the identifiers available in the dataset for gaining access to the content of the imdb and tmdb web pages. For each movie, the plots from these two sites were concatenated, and text pre-processing techniques were used with the NLTK Python tool, namely tokenization, lowercase conversion, numbers, and punctuation removal as well as stemming.

The ratings and number of users before and after the removal of low ratings, are presented in Table 1. It is worth noticing that after the removal of ratings with values lower than $3,5$, the mean rating per genre is greater than $4,5$; before the removal, this rating was in the range of $3,5$ to $3,8$. In addition, concerning the number of users per genre, it is shown that the same genres are considered as popular before and after the removal of low ratings.

[1] `https://grouplens.org/datasets/movielens/`

Table 1. Ratings and Number of Users Before and After Removal of Low Ratings

Genres	Number of Movies	Rating Before	Rating After	Number of Users Before	Number of Users After
Adventure	1263	3,61	4,24	606	588
Animation	611	3,63	4,21	527	458
Children	664	3,48	4,2	559	482
Comedy	3756	3,56	4,24	609	603
Fantasy	779	3,57	4,26	583	528
Romance	1596	3,63	4,24	606	588
Drama	4361	3,75	4,3	610	606
Action	1828	3,55	4,23	608	596
Crime	1199	3,75	4,31	603	581
Thriller	1894	3,62	4,25	609	599
Horror	978	3,45	4,24	535	472
Mystery	573	3,75	4,29	580	542
Sci-Fi	980	3,53	4,23	605	570
War	382	3,87	4,33	551	513
Musical	334	3,58	4,21	470	397
Documentary	440	3,77	4,21	223	191
IMAX	158	3,8	4,25	458	415
Western	167	3,64	4,23	420	357
Film-Noir	87	3,84	4,26	239	209
no genres listed	34	3,61	4,22	26	20



(a) Ratings Per Genre (b) Users Per Genre

Fig. 1. Ratings and Number of Users Before and After Removal of Low Ratings

4.2 Input Data for the Clustering Algorithm

In following, the construction of the matrices given as input to the clustering algorithm for the cases of user clustering and movie clustering, was implemented. Concretely, BM25, Tf/Idf and User Preference scores are taken into consideration for our proposed implementations.

4.2.1 Scores Matrices BM25 scores can be utilized in a $1863x1863$ matrix, where its rows and columns correspond to the different movies of the dataset. Each cell holds the BM25 ranking score for the plot of the movie as the column defines the query, and the row defines the plot of the movie.

On the other hand, Tf/Idf scores can be utilized in a $1863x5432$ matrix. Specifically, each row corresponds to a different movie, and the columns correspond to the stemmed terms of all the movie plots. Each cell holds the Tf/Idf weight of the column term for the movie plot in the corresponding row.

4.2.2 User Preference Matrix The user preference matrix represents the users' preferences for the different genres of movies. Since our goal is to provide the user with movie recommendations, the first step was to remove all ratings less than $3,5$. If a user has not rated any movies with a rating greater or equal to $3,5$, the user is excluded from the procedure. For every user, the total number of movies being watched, belonging to a particular genre, was counted. In following, this count was divided by the total number of movies the user has watched. Specifically, each matrix row represents a different user, each column corresponds to a different movie genre, and each cell defines the percentage of preference of that user towards that particular genre.

4.2.3 Clustering Our approach regarding the implementation of a recommendation system utilizing the K-Means clustering algorithm is two-fold, namely, user clustering and movie clustering.

Regarding the user clustering approach, the examination of users' ratings, the percentage calculation of their different genres preference, as well as the clustering based on these percentages, are implemented. The movie clustering approach is utilized in two different cases, namely focusing on the plot of each movie and implementing content-based filtering. In the first case, each movie is represented by a vector containing the Tf/Idf weights of the terms of the movie plots, while in the second case, by a vector containing the corresponding BM25 scores. In both implementations, each user is assigned to the cluster containing the highest number of movies being watched by this user. In order for the proposed method to recommend a movie to a user, the movies being watched by other users of the same cluster are initially examined. In following, the mean rating of those movies, by considering only the ratings given by the users of the cluster, is calculated and finally, the movies with the highest ratings are recommended.

The data is given as input in the form of the matrices, as mentioned above. Dimensionality reduction is considered to be an essential step before executing the K-Means clustering, and the technique utilized for this purpose is Principal Component Analysis (PCA).

4.2.4 Classification The aim of recommendation systems is to calculate the probability of the movie rating given to a movie by a user while taking into consideration the ratings given by other users. Thus, a user-based and item-based

collaborative filtering, as well as a combination of these two methods, are implemented. In the user-based case, each movie is represented by a vector containing all users' ratings given to that movie. Similarly, in the item-based case, users are considered as observations, and the movies form the observation features. The combinatorial filtering results from the mean of the predicted ratings, are derived from the user-based as well as the item-based filtering.

A user id, representing the user to whom movies will be recommended, and the user-movie matrix are given as input. Moreover, the missing values of each row of the user-movie matrix were filled by the mean rating of the user corresponding to that row. For the probability calculation, the multinomial Naive Bayes algorithm is utilized where we have set $a = 1$ for Laplace Smoothing. The specific algorithm is chosen because the possible ratings are on a scale of 1 to 5, meaning they are not binary values and their distribution is not known.

Regarding the data splitting, in the case of user-based filtering, 70% was randomly used for training and 30% for testing. On the other hand, when considering the item-based filtering, the users to whom recommendations will be made are defined as test set whereas training set includes the rest of the users.

5 Evaluation

For the evaluation of our proposed methods, we have used Accuracy, which is one of the most commonly used metrics for the evaluation of a system prediction. Since our goal is to compare the different approaches presented in our work, we have converted the ratings to binary values using a unified manner, in a way that high ratings ($>= 3,5$) are replaced by the value 1 and the lower ratings ($< 3,5$) are replaced by the value 0. In the case of Bayesian classifier, high ratings are considered the ones with value equal to or greater than 3.

For our experiments, 10 users were randomly chosen, with the only requirement having watched and rated at least 100 movies. The accuracy was calculated for each user by examining their actual ratings in relation to the predicted ratings for the movies they have watched. In following, the mean of the predictions accuracy for the different methods is calculated.

The first two principal components were the ones carrying the largest amount of information, as depicted in Table 2. In the case of movie clustering, even though the first two components held a small percentage of the total variance, our experiments showed that using more components in the clustering phase did not significantly improve the clustering results.

In order to determine the optimal number of clusters to be formed, the elbow curve method has been used. The optimal number of clusters k, in the user clustering is set as 5, and in the case of movie is set as 4. Even for the case of movie clustering, the elbow curve method provided ambiguous results, and after testing the algorithm numerous times, it was observed that high values of k lead to uneven distribution of points inside the clusters, as presented in Table 3. Furthermore, by examining the most frequent words of the movie plots in each cluster, it was found that for values bigger than 4, the formed clusters had in

Table 2. Principal Component Variance

PCA Features	User Preference Matrix	Tf/Idf Matrix	BM25 Matrix
0	0,34391	0,00394	0,06221
1	0,24211	0,00370	0,01613
2	0,09896	0,00334	0,01260
3	0,06825	0,00313	0,01207
4	0,05264	0,00303	0,01166
5	0,04318	0,00284	0,01114
6	0,02570	0,00279	0,00972
7	0,02316	0,00262	0,00931
8	0,02169	0,00261	0,00856
9	0,01582	0,00248	0,00769
10	0,01369	0,00245	0,00731
11	0,01183	0,00238	0,00724
12	0,00943	0,00236	0,00693
13	0,00888	0,00232	0,00664
14	0,00794	0,00227	0,00630

common the most frequent words, indicating that there was no clear separation regarding the content of the movie plots.

Furthermore, Tables 4, 5 and 6 introduce the experimental results in terms of Clustering, Classification using Bayesian model and Weighted Sum, respectively. Regarding clustering, the Tf/Idf movie method yields the best accuracy results. It is worth mentioning that the accuracy of the user clustering method is equal to 66.17%, and even though it is lower than the movie clustering method accuracy, it outperforms the basic Weighted Sum approaches.

In addition, regarding the classification experimental evaluation, the best result was derived from the movie based probabilistic Bayesian approach. Specifically, the Bayesian model outperforms the basic Weighted Sum model with a difference approximately 24% for the user-based case and about 20% for the movie-based case. The low accuracy score of the combinational Bayesian approach is due to the fact that the system examined only the first 1000 movies for the movie-based method.

Finally, the high similarity of the Weighted Sum results can be considered because of the low dimensionality of the user vectors utilized in our experiments.

6 Conclusions and Future Work

This paper offers an extensive analysis of different approaches for the implementation of Movie Recommendation Systems, providing an integrated solution to the recommendation problem. On one hand, it proposes the Bayesian Collaborative filtering approach that renders the best results, outweighing the other models discussed in this paper. On the other hand, it proposes a Content-based technique based on movie clustering according to their plots through the Tf/Idf

Table 3. Sum of Squared Errors

Number of Clusters	User Clustering	Tf/Idf Movie Clustering	BM25 Movie Clustering
1	130.5	1840.6	72655490
2	101.3	1835.8	69883117
3	86.9	1831.6	68966348
4	79.1	1827.7	68471660
5	73.8	1825.3	68207625
6	69.9	1822.2	67713490
7	66.5	1820.1	67586018
8	63.3	1816.8	67131015
9	61.4	1817.3	66882121
10	59.4	1811.1	66828590
11	57.6	1810.3	66513930
12	56.3	1809.5	66236101
13	54.7	1808.3	66236317
14	53.7	1805.7	66045114
15	52.3	1804.9	65882937
16	50.6	1802.1	65557487
17	49.6	1801	65666520
18	49.2	1801.2	65355018
19	48.1	1797	65455797

Table 4. Clustering

User Clustering	66,17%
Tf/Idf Movie Clustering	67,96%
BM25 Movie	63,86%

Table 5. Classification (Bayes)

User Based	80,96%
Movie Based	83,64%
User and Movie Based	30,45%

weighting scheme, which yields a solution to the movie recommendation problem when user preference information is not available. Furthermore, the three different implementations are introduced, namely, K-Means for users and movies clustering, Naive Bayes classification and Weighted Sum for user-based, item-based and both in combination collaborative filtering.

Future work is bound to include the construction of a hybrid model combining the Bayesian and clustering techniques. The user clustering approach, despite not being the most accurate one, provides an adequate way of detecting similarities between users concerning their preferences towards the different genre of movies. It could serve as a base for the probabilistic approach, i.e., only the users belonging to the same cluster would be considered for the calculation of the probabilities of each prediction. Furthermore, it is deemed appropriate to test different techniques for dimensionality reduction, other than the PCA method. Finally, an interesting approach would be to give, as query, one term at time in the BM25 plot clustering.

Table 6. Weighted Sum for Cosine Similarity and Euclidean Distance

Cosine Similarity			Euclidean Distance		
User Based	**Movie Based**	**User and Movie Based**	**User Based**	**Movie Based**	**User and Movie Based**
56,21%	64,14%	60,45%	56,21%	64,14%	60,21%

Acknowledgement

Andreas Kanavos, Aristidis Ilias and Christos Makris have been co-financed by the European Union and Greek national funds through the Regional Operational Program "Western Greece 2014-2020", under the Call "Regional Research and Innovation Strategies for Smart Specialisation - RIS3 in Information and Communication Technologies" (project: 5038701 entitled "Reviews Manager: Hotel Reviews Intelligent Impact Assessment Platform").

References

1. Baeza-Yates, R.A., Ribeiro-Neto, B.A.: Modern Information Retrieval: The Concepts and Technology Behind Search, Second edition. Pearson Education Ltd., Harlow, England (2011)
2. Byström, H.: Movie Recommendations from User Ratings. Stanford University (2013)
3. Croft, W.B., Metzler, D., Strohman, T.: Search Engines: Information Retrieval in Practice. Pearson Education (2009)
4. Ganesan, K., Zhai, C.: Opinion-based entity ranking. Information Retrieval 15(2), 116–150 (2012)
5. Gourgaris, P., Kanavos, A., Makris, C., Perrakis, G.: Review-based entity-ranking refinement. In: 11th International Conference on Web Information Systems and Technologies (WEBIST). pp. 402–410 (2015)
6. Jiang, M., Song, D., Liao, L., Zhu, F.: A bayesian recommender model for user rating and review profiling. Tsinghua Science and Technology 20(6), 634–643 (2015)
7. Kanavos, A., Kotoula, P., Makris, C., Iliadis, L.: Employing query disambiguation using clustering techniques. Evolving Systems pp. 1–11 (2019)
8. Kanavos, A., Makris, C., Plegas, Y., Theodoridis, E.: Ranking web search results exploiting wikipedia. International Journal on Artificial Intelligence Tools (IJAIT) 25(3), 1–26 (2016)
9. Lydia, E.L., Govindaswamy, P., Lakshmanaprabu, S., Ramya, D.: Document clustering based on text mining k-means algorithm using euclidean distance similarity. Journal of Advanced Research in Dynamical and Control Systems (JARDCS) 10(2), 208–214 (2018)
10. Makris, C., Panagopoulos, P.: Improving opinion-based entity ranking. In: 10th International Conference on Web Information Systems and Technologies (WEBIST). pp. 223–230 (2014)
11. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)
12. Marlin, B.: Modeling user rating profiles for collaborative filtering. In: 16th International Conference on Neural Information Processing Systems. pp. 627–634 (2004)

13. Miyahara, K., Pazzani, M.J.: Improvement of collaborative filtering with the simple bayesian classifier. Information Processing Society of Japan 43(11) (2002)
14. Phorasim, P., Yu, L.: Movies recommendation system using collaborative filtering and k-means. International Journal of Advanced Computer Research (IJACR) 7(29), 52 (2017)
15. Pomerantz, D., Dudek, G.: Context dependent movie recommendations using a hierarchical bayesian model. In: 22nd Canadian Conference on Artificial Intelligence. pp. 98–109 (2009)
16. Sarwar, B.M., Karypis, G., Konstan, J., Riedl, J.: Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In: 5th International Conference on Computer and Information Technology (ICCIT). pp. 291–324 (2002)