ISSN: 0067-2904

Prediction of Explicit Features for Recommendation System Using User Reviews

Sabeena Yasmin Hera[1]*, Mohammad Amjad
Department of Computer Engineering, Jamia Millia Islamia, New Delhi, India

Received: 21/12/2020 Accepted: 27/11/2021 Published: 30/11/2022

Abstract

With the explosive growth of data, it has become very difficult for a person to process the data and find the right information from it. So, to discover the right information from the colossal amount of data that is available online, we need information filtering systems. Recommendation systems (RS) help users find the most interesting information among the options that are available. Ratings given by the users play a vital role in determining the purposes of recommendations. Earlier, researchers used a user's rating history to predict unknown ratings, but recently a user's review has gained a lot of attention as it contains a lot of relevant information about a user's decision. The proposed system makes an attempt to deal with the problem of uncertainty in the rating histories by using textual reviews. Two datasets are used to experimentally analyze the proposed framework. In this approach, clustering techniques are used with natural language processing (NLP) for prediction. It also compares how different algorithms, such as K-mean, spectral, and hierarchical clustering algorithms, produce a varied outcome and concludes which method is appropriate for the given recommendation scenarios. We also validate how the proposed method outperforms the non-clustering-based methods.

Keywords: Hierarchical clustering, K-mean clustering, Recommendation systems, spectral clustering.

1. Introduction

The growth of the World Wide Web during the 1990s brought about an equivalent increase in the amount of data that is accessible online, exceeding the capacity of individual clients to process this data. Early research in recommendation systems began to grow out of research on information filtering and information retrieval [1]. Recommendation systems support users by analyzing and assessing information from other users to identify content, items, and services (such as websites, digital devices, movies, books, music, TV shows, etc.) [2]. Recommender frameworks effectively foresee things the user may be keen on and add them to the data streaming to the user.

There have been many improvements in the field of recommendation systems, which have empowered a huge number of clients to effectively get access to data for any service, including education, travel, health, food, gaming, and electronics, in almost every field. A recommendation system's most important feature is its ability to devise the preferences and interests of a user by analyzing this user's behavior and/or other user's behavior to yield

*Email: hera.yasmin1@gmail.com

custom recommendations [1,3]. Initially, there were two information filtering techniques introduced for recommendation: content-based filtering [4] and collaborative-based filtering [5].

In a content-based recommender, recommendations are made based on the past preferences of the user. The content is represented by the previous interests of users on their profile. It can be the attributes and features of the document that represent the item. In Collaborative Recommender, recommendations are made on the basis of the past preferences of other similarly interested users. Later, a hybrid recommendation model was developed based on the amalgamation of these two recommendation techniques [6]. User ratings and written reviews are two main sources of data used by recommendation systems. There are numerous elements and factors which influence a client's decision, specifically location, quality, quantity, area, purchasing history, and client's nature, to name a few.

The opinions that are expressed for a service, like reviews, posts, etc., are of great value to customers to help them purchase an item or service. In this respect, there are several statistics showing the relevance of this data from a user's point of view. According to a BrightLocal local market survey conducted in 2018, 91 percent of customers between the ages of 18 and 34 trust online reviews as much as personal recommendations. Approximately 57 percent of customers will only use a company that has four or more stars. According to a 2019 survey, 82 percent of consumers read online reviews, with 52 percent of them saying they "always" read reviews.

This paper exploits the capability of customer reviews to predict rating scores for a review-based recommender system. A recommender system often suffers from scalability and sparsity problems. The proposed framework deals with this issue by using clustering techniques as a pre-recommendation step. In this work, we use K-means clustering [7], spectral clustering [8], and hierarchical clustering [9] algorithms to cluster the users in the dataset according to the reviews and descriptions reflecting their preferences that are provided by them.

The paper is divided into five sections. Section 1 introduces the background, context, and significance of the study. The related work is presented in Section 2. Section 3 describes the techniques and models that are used in the proposed work. It also describes the workflow of the steps contained within it. Section 4 provides an overview of the datasets and the evaluation metric. It also discusses the results of the experiments when the proposed method is applied to the aforementioned datasets. The conclusion and future work are presented in the final section 5.

2. Literature Review

Rating prediction in recommendation systems can be described either as review-based score (RBR) prediction or missing score (MS) prediction in a [user, item] matrix. The major difference between them is that RBR is based on the textual feedback given by the consumers, whereas MS prediction in the [user, item] matrix is based on the rating history of the consumers. The rating score prediction using the latter type becomes difficult when the [user, item] matrix is sparse. Therefore, few research studies consider using text-based information for predicting scores. Pang and Lee [10] used Support Vector Machine (SVM) to predict ratings using reviews. They formulated it as a multi-classifier and regression problem and concluded that a regression model of SVM performed better than that of the multi-classifier model.

RS uses clustering algorithms to recognize clusters of consumers with comparable preferences. Many CF-based recommendation algorithms have integrated clustering methods to mitigate the problems of scalability and sparsity. Ghazarian et al. [11] suggested a collaborative approach that produces group recommendations based on the item and the user's similarity. Similarity of items is recognized using SVM (Support Vector Machines) and similarity of users is recognized using similarity measures. In addition, it fills the vacant entries of the user-item matrix by predicting the most suitable values.

While the scheme demonstrates a greater accuracy value and decreased error rate, the issue of scalability continues to be unresolved. Wei et al. [12] evaluated a RS film using a hybrid strategy that customizes both the sentence-level tags posted on the films and the personal scores given by the user. Traditional methods measure [user, item] matrix similarity, whereas this method uses Singular Value Decomposition (SVD) techniques and matrix factorization. [13] proposed a cluster-based collaborative filtering system based on K-means to smooth out the unrated data by cluster for each user. Using a weighted co-clustering algorithm, George and Merugu [14] used a collaborative filtering approach that involves synchronized clustering of users and items. Birtolo et al. [15] utilized fuzzy clustering on item-based CF recommender systems to create a trust aware clustering CF.

In this paper, we perform partitioning based on textual information and study the prediction of the existing reviews.

3. Techniques and Model Used

In this section, the proposed method for rating score prediction will be demonstrated. The steps followed by the suggested method are: dataset acquisition; data preprocessing; feature extraction; clustering the dataset; and then applying the recommendation algorithm.

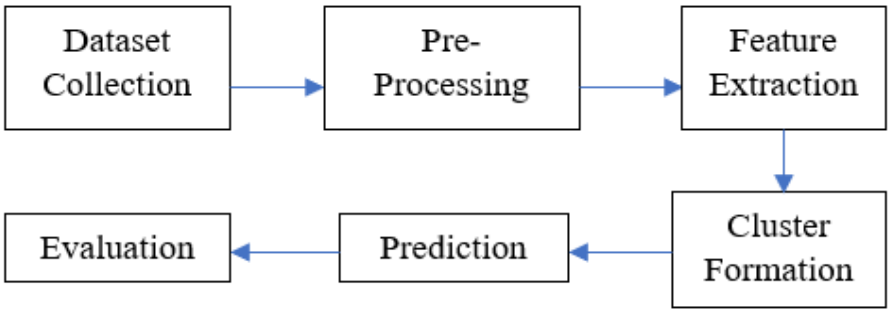


Figure 1: Flowchart of the proposed work

3.1 Pre-processing

Once the data is collected, it needs some preprocessing before analyzing and evaluating the data. Incomplete, inaccurate, and contaminated data analysis can lead to inappropriate and below-quality results. Since the data is in natural language form, a set of Natural Language Processing (NLP) tasks needs to be performed before further processing. It includes tokenization, stopword removal, and stemming.

Tokenization is the first step in preprocessing. It involves dividing longer text strings into smaller parts, or tokens. It is possible to tokenize larger pieces of text into phrases, tokenize phrases into words, etc. Tokenization is also known as text segmentation or lexical analysis. It detaches numbers, words, symbols, and other characters from the string. The next step is stopword removal. Stopwords are words with high prevalence all over the sentences; these words do not contain much valuable information. Such words are generally used to connect components of a sentence instead of displaying topics, items, or purposes. By comparing the

text to a stopwords list, we can remove words like "the" or "and" etc. Then stemming is performed, which reduces words to their root, generally a suffix, by dropping unnecessary characters. Here, the morphological forms of words are removed. Several stem designs are available, including Porter and Snowball. In our work, Porter Stemmer [16] is used.

3.2 Feature Extraction

The texts are transformed into a structured form by converting them into vectors for further processing. This vector space modelling deals with feature extraction from texts. A statement is transformed into a number vector based on the Bag of Words model with a fixed length; each term of the statement is a Tf-idf score. Tf-idf weight is a statistical measure used in a set or corpus to assess how essential a term is to a text and is calculated using Eq. (1).

$$Tf\text{-}idf\ score = TF * IDF \quad (1)$$

$$TF\ (t) = No.\ of\ times\ t\ appears\ in\ a\ document/\ Total\ no.\ of\ terms\ in\ the\ document \quad (2)$$

$$IDF\ (t) = log\ (Total\ number\ of\ documents/No.\ of\ documents\ with\ term\ t\ in\ it) \quad (3)$$

where *TF*= Term Frequency, *IDF*= Inverse Document Frequency and *t*= term in the document.

3.3 Clustering Algorithms

A clustering technique is used for grouping similar users in one cluster and dissimilar users in another. The proposed work uses clustering methods based on K-means, Spectral and Hierarchical methods to cluster users' datasets.

3.3.1 K-means clustering

K-means clustering is a supervised learning algorithm. The algorithm here takes textual information from users as an input and divides them into K numbers of clusters. The clusters here are formed so that the intracluster sum of squares is minimized.

$$minimize\ J = \sum_{j=1}^{k}\sum_{i=1}^{n}\left\| x_i^{(j)} - c^j \right\|^2 \quad (4)$$

where $x_i^{(j)}$= the user's review; c^j= chosen centroid for the cluster generation; $\left\| x_i^{(j)} - c^j \right\|^2$= distance measure used for clustering; k= number of required clusters.

3.3.2 Spectral clustering

In recent times, Spectral Clustering has emerged as a widespread clustering method for grouping texts into clusters. Spectral clustering is very helpful when the composition of the individual clusters is highly non-convex. The Laplacian matrix is the most important step in the spectral clustering technique. It is also called the graph Laplacian [18]. It uses the spectral decomposition of the Laplacian matrix built on the input dataset [17]. The graph for the Laplacian matrix is an undirected weighted graph. The users' text graph can be built using ε-neighborhood graph, K-nearest neighbor graph, or fully, connected graph. The number of clusters existing in the dataset can be deduced by projecting the points into a non-linear embedding and analyzing the Eigen values of the Laplacian matrix [19].

$$L = D - Wt \quad (5)$$

where L= Un-normalized Laplacian matrix; D = Diagonal Matrix/Degree matrix and Wt = weight matrix with wtij=wtji ≥ 0 since, graph is undirected.

3.3.3 Hierarchical clustering

In the hierarchical clustering technique, clusters are formed by dividing or integrating data in a top-down or bottom-up manner, respectively. It is categorized into agglomerative clustering and divisive clustering based on the approach taken for forming clusters [20]. The agglomerative follows the bottom-up strategy, which builds clusters by considering every user belonging to an individual cluster and then merging these nuclear clusters into larger clusters until all users are lastly placed in a single cluster or otherwise until a certain termination criterion is fulfilled. The divisive clustering follows the top-down strategy, which breaks down clusters comprising all users into smaller clusters until each user forms a cluster on their own or until certain termination criteria are met. In agglomerative clustering, two users are chosen to be merged using a linkage function. In this work, minimum variance linkage is used for cluster formation. In this linkage function, the decision to merge two clusters is based on the minimum merging cost. The merging cost of two clusters, say A and B, can be defined by the sum of the squares of the data points to their center [21].

$$cost(A,B) = \sum_{i\in A\cup B}\|\vec{x}_i - \vec{m}_{A\cup B}\|^2 - \sum_{i\in A}\|\vec{x}_i - \vec{m}_A\|^2 - \sum_{i\in B}\|\vec{x}_i - \vec{m}_A\|^2 \quad (6)$$

where $\vec{m}_j$= center of cluster j and $\vec{x}_i$ = each user i.

3.4 Recommendation and Prediction

After the clustering step and in order to predict ratings for target users, the system computes the similarity of the target user to all its neighborhood users belonging to the same cluster based on their available data. The similarity between users is calculated using cosine similarity [22]. The user with whom the target user is found to have the maximum similarity is used for prediction of ratings. The detailed algorithm for the proposed framework is discussed below:

Algorithm 1: Algorithm for the Recommendation

Input: dataset D containing reviews and ratings
Output: Predicted rating for a given user
begin
D= dataset
K= number of clusters
N=number of reviews in D
y= rating
n= range of test set for each cluster k
m= range of training set for each user k
Step1: Randomly select K reviews from N as the centroid of initial clusters.
Step2: Iteratively divide D into K clusters using aforementioned clustering techniques
Step3: For each cluster
a) Split data into test-set and training set, test set=0.30 and training set= 0.70
b) Calculate cosine similarity between (test[i], training[j]) say, W[i][j]
For i ← 1 to n
For j ← 1 to m
If max_value[i] == W[i][j] // search the matrix W[i][j] for maximum value corresponding to each test[i].
Assign test_y[i] == train_y[j] // Assign the y value of training[j] to test[i]
to get the predicted value of y
End For
End For
End For
End

4. Experimental Evaluation

4.1 Experimental Settings

Two datasets have been used for the proposed work. One of them is consumer reviews of Amazon for beauty and health. It contains 12071 reviews given by users, of which those having a neutral rating were dropped. The other dataset is Amazon Fine Food Reviews dataset, and after the preprocessing, the top 5000 reviews were taken into consideration for experimentation. Although the datasets use a 5-star rating system, we have converted it into the "high" ({4, 5}) and "low" ({1, 2,}) binary groups. In addition, we restructured rating estimation as a classification problem where we estimate the probability that a user would "like" an item or not.

4.2 Evaluation metric

The Mean Absolute Error (MAE) is used to assess the performance measurement of the proposed framework because of its accuracy and simplicity, which suits the experiment's objective. It is one of the most widely used metrics for performance evaluation of recommender systems. MAE estimates in a collection of projections the median magnitude of the errors. The relative scores between the predictions and the resulting observations over the test set are averaged [23]. The MAE is calculated using Eq. (7).

$$Mean\ Absolute\ Error(MAE) = \frac{\sum_{j=1}^{n}\|p_j - a_j\|}{n} \quad (7)$$

where n = number of ratings; p_j= predicted rating of j_{th} user and a_j = actual rating of j_{th} user.

4.3 Result and analysis

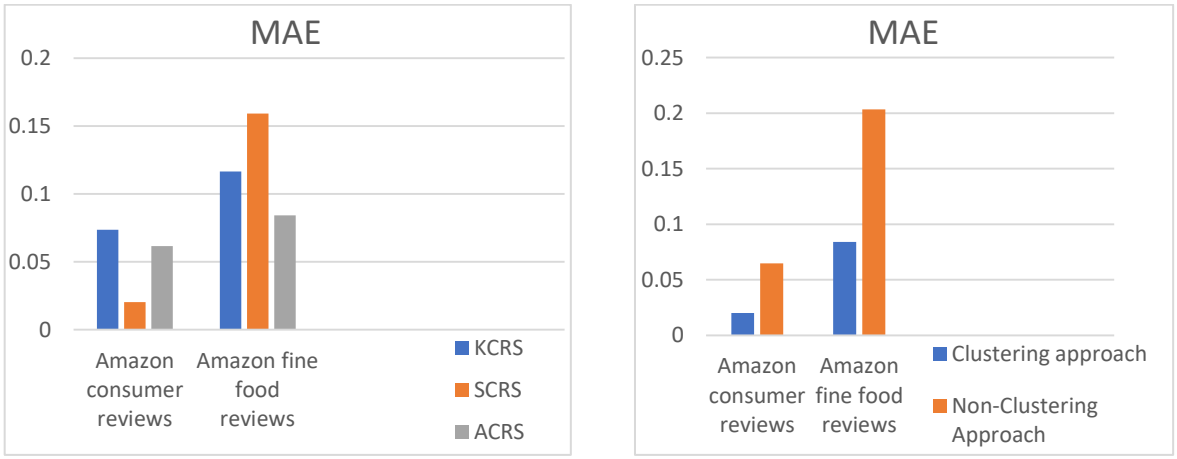
The study was set out to enhance the accuracy of the rating prediction for the recommendation system and to identify the effects of various clustering schemes on it. Hence, this section shows the analysis and results discussion. To conduct the experiment, we used Python for implementation, and the parameters of the host system were Win 10, intel® core™ i5-8265U, X64 and 8GB RAM.

Table 1. demonstrates the experimental results of the clustering-based framework for recommender systems using textual information. We have evaluated different numbers of clusters and presented the results for numbers that were close to the optimal result. It shows the effect of different clustering schemes, i.e., K-means clustering-based recommender system (KCRS), spectral clustering-based recommender system (SCRS), and agglomerative clustering-based recommender system (ACRS) on the accuracy of the prediction in terms of

Table 1: MAE of KCRS, SCRS, ACRS for different number of clusters

Number of clusters	MAE					
	Amazon consumer reviews			Amazon fine food reviews		
	KCRS	**SCRS**	**ACRS**	**KCRS**	**SCRS**	**ACRS**
C=11	0.0871	0.0233	0.0857	0.1858	0.1718	0.1051
C=12	0.0911	0.0202	0.092	0.1556	0.1592	0.0908
C=13	0.0737	0.0211	0.0717	0.1166	0.1911	0.0842
C=14	0.0747	0.0209	0.0616	0.1968	0.1849	0.0917

MAE. We have also determined the threshold value which gives us the minimal MAE for each of the algorithms. After extensive testing, it is found to be near $\log_2\lfloor n/2 \rfloor \pm 1$. It also appears that the performance of the recommendation system depends on the type of dataset and the number of clusters. The clustering-based framework outperforms the non-clustering one only when the above two conditions are met.



(a) **(b)**

Figure 2: This figure depicts Comparison of Optimal Results Of KCRS, SCRS And ACRS For the Datasets in (a) Frame 1 and Comparison of The Best Performing Clustering-Based Approach to the Non-Clustering Based Approach in (b)

Figure 2 shows the comparison between the optimal values of KCRS, SCRS, and ACRS. For the Amazon consumer reviews dataset, SCRS outperforms the other two approaches. It is because the dataset was concave in nature. Whereas for Amazon fine food Reviews ACRS performed better than the KCRS and SCRS. The proposed approach is then compared to the non-clustering approach for recommendation system. It shows that the proposed clustering approach performs better than the non-clustering one.

5. Conclusion

User reviews are important for recommendation purposes as they describe the intentions of a user in a better way, along with the objectivity of rating a service. In this paper, we present the recommendation framework using different clustering techniques. Clustering helps us to reduce the search space for finding the most similar and relevant item or user to the active user. It also overcomes the scalability problem. Through the experimentation, we therefore conclude that the clustering-based rating score prediction system outperforms the non-clustering-based one when the number of clusters formed is optimal. A best-fit clustering-based prediction system will outperform others based on different clustering methods for any given number of clusters and will perform better than the non-clustering-based RS when the number of clusters is optimally considered.

The first proposal in future developments would be to assess reviews on the grounds of each phrase, as each phrase may have a distinct polarity and outlook based on objectivity. Analyzing reviews with punctuation included and deciphered could be further refinements. For the identification of the emojis written along with textual content, the textual analysis can be further extended. Other techniques of clustering may be employed to group comparable items. The framework can also be used to explore different domains, too.

References

[1] J.A. Konstan and J. Riedl, "Recommender systems: from algorithms to user experience," *User Model User-Adapt Interact*, vol. 22, pp. 101–23, 2012.

[2] E. Frias-Martinez, G. Magoulas, S. Chena and R. Macredie, "Automated User Modeling for Personalized Digital Libraries," *International Journal of Information Management*., vol. 26, no. 3, pp. 234–248, 2006.

[3] F.O. Isinkaye, Y.O. Folajimi and B.A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian Informatics Journal*, Vol. 16, no. 3, pp. 261-273, 2015.

[4] J.B. Schafer, D. Frankowski, J. Herlocker and S. Sen, "Collaborative Filtering Recommender Systems", Edited P. Brusilovsky, A. Kobsa, W. Nejdl, The Adaptive Web, Lecture Notes in Computer Science, Heidelberg, vol. 4321, pp. 291-324, 2007.

[5] M. J. Pazzani and D. Billsus, "Content-Based Recommendation Systems", Edited P. Brusilovsky, A. Kobsa, W. Nejdl, The Adaptive Web, Lecture Notes in Computer Science, Heidelberg, vol. 4321, pp. 325-341, 2007.

[6] Z. Ji, H. Pi, W. Wei, B. Xiong, M. Woźniak and R. Damasevicius, "Recommendation Based on Review Texts and Social Communities: A Hybrid Model," *IEEE Access*, vol. 7, pp. 40416-40427, 2019.

[7] R. Xu and D. Wunsch, "Survey of Clustering Algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645-678, 2005.

[8] U. V. Luxburg, "A Tutorial on Spectral Clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[9] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2017.

[10] B. Pang and L. Lee, "Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales," *in Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics on Association for Computational Linguistics*, Ann Arbor, USA, 2005.

[11] S. Ghazarian and M. A. Nematbakhsh, "Enhancing memory-based collaborative filtering for group recommender systems," *Expert Systems with Applications*, vol. 42, no. 7, pp. 3801–3812, 2015.

[12] S. Wei, X. Zheng, D. Chen and C. Chen, "A hybrid approach for movie recommendation via tags and ratings," *Electronic Commerce Research and Applications*, vol. 18, no. c, pp. 83–94, 2016.

[13] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu and Z. Chen, "Scalable collaborative filtering using *cluster*-based smoothing," *in Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, USA, 2005.

[14] T. George and S. Merugu, "A scalable collaborative filtering framework based on co-clustering," *in Proceedings of the Fifth IEEE International Conference on Data Mining*, Washington, USA, 2005.

[15] C. Birtolo and D. Ronca, "Advances in Clustering Collaborative Filtering by means of Fuzzy C-means and trust," *Expert Systems with Applications*, vol. 40, no. 17, pp. 6997–7009, 2013.

[16] M. Bounabi, K. E. Moutaouakil and K. Satori, "A comparison of text classification methods using different stemming techniques," *International Journal of Computer Applications in Technology*, Vol. 60, No. 4, pp. 298 – 306, 2019.

[17] Z. Zhang and S. R. Kulkarni, "Detection of shilling attacks in recommender systems via spectral clustering," *in 17th International Conference on Information Fusion (FUSION)*, Salamanca, 2014.

[18] C. H. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, "A min-max cut algorithm for graph partitioning and data clustering," *in IEEE international conference on data mining*, San Jose, CA, 2001.

[19] H. D. Menéndez and D. Camacho, "GANY: A genetic spectral-based clustering algorithm for Large Data Analysis," *in IEEE Congress on Evolutionary Computation (CEC)*, Sendai, Japan, 2015.

[20] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *WIREs Data Mining Knowl Discov*, Vol. 2, No. 1, pp. 86-97, 2011.

[21] Zhao, Y., Karypis, G., & Fayyad, U, "Hierarchical Clustering Algorithms for Document Datasets," *Data Mining and Knowledge Discovery*, vol. 10, no. 2, pp. 141–168, 2005.

[22] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, Vol. 22, no. 1, pp. 5–53, 2004.

[23] H. N. Kim, A. T. Ji, I. Ha and J. S. Jo, "Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation," *Electronic Commerce Research and Applications*, vol. 9, no. 1, pp. 73–83, 2010.