Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 10, No. 2, 2022, 185-195.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

Increasing Performance of Recommender Systems by Combining Deep Learning and Extreme Learning Machine

Zahra Nazari, Hamidreza Koohi* and Javad Mousavi

Computer Engineering Department, Shomal University, Amol, Iran.

Article Info

Article History:
Received 28 September 2021
Revised 12 December 2021
Accepted 16 February 2022

DOI:10.22044/jadm.2022.11248.2279

Keywords:
Recommender Systems, Extreme Learning Machine, Restricted Boltzmann Machine, Data Sparsity, Clustering methods.

**Corresponding author: koohi@shomal.ac.ir(H.R. Koohi).*

Abstract

Nowadays, with the expansion of the internet and its associated technologies, the recommender systems have become increasingly common. In this work, the main purpose is to apply new deep learning-based clustering methods in order to overcome the data sparsity problem, and increment the efficiency of the recommender systems based on precision, accuracy, F-measure, and recall. Within the suggested model of this research work, the hidden biases and input weights values of the extreme learning machine algorithm are produced by the restricted Boltzmann machine, and then clustering is performed. Also, this work employs extreme learning machine (ELM) for two approaches, clustering of the training data and determining the clusters of the test data. The results of the proposed method are evaluated in two prediction methods by employing the average and Pearson correlation coefficient in the MovieLens dataset. Considering the outcomes, it can be clearly said that the suggested method can overcome the problem of data sparsity and achieve a higher performance in the recommender systems. The evaluation results of the proposed approach indicate a higher rate of all evaluation metrics, while using the average method results in the rates of precision, accuracy, recall, and F-measure come to 80.49, 83.20, 67.84, and 73.62, respectively.

1. Introduction

Nowadays, the societies have undergone rapid changes in almost every aspect using computers and computer networks. We shop online, collect data through search engines, and spend a considerable portion of our social life online [1]. The exponential growth of information and online users has led to the information overload problem. The efficient extraction of useful data from all accessible online data is challenging since the Internet is growing rapidly every day. The recommender systems are a set of software devices and methods that guide the user in a modified method toward the required items in a vast array of options. The purpose of this personal recommender system is to find a novel item from a large set of data in terms of the previous user preferences [2].

The recommender systems have many applications in various fields including digital libraries, medical applications, e-commerce, etc. These systems are among the most significant sectors in e-commerce. According to the reviews, the existence of recommender systems in this area has increased the revenue and profitability of selling a product. For example, in the field of film, the number of films and viewers has grown dramatically over the past few years. The information is beneficial exclusively for the users tending to watch an indefinite movie. Though, the list of features provided by the web search engine is very large, and it is very time-consuming to evaluate the options of this long list of searches. The recommender systems can use a variety of suggested techniques such as content-based, collaborative filtering-based, hybrid and knowledge-based recommendations [3-10].

The content-based systems recommend items based on the past preferences of a user, while the collaborative filtering systems are simple. The basis of this technique assumes that the users who share the same opinion on several items also agree on the other items, and the clustering methods are generally utilized for RS recommendation. In other words, in this kind of recommendation, the items are determined by evaluating the ratings of the other users on the items. Collaborative filtering is extensively used in RS, and is the most effective recommendation method that has become a favorite topic among the researchers. In the recommender systems, the researchers have several issues and challenges that affect the performance of their algorithms. Challenges in this field include data sparsity, scalability, cold start, and vulnerability to cyber-attacks. Since the number of items and user preferences are very large and unstructured, mostly the overlap between the users is none or very small. High sparsity is a big challenge that effects the quality of predictions and performance of the recommender algorithm since the confidence of predicted ratings is based on a rather small amount of evidence [3, 8-12]. The majority of the present methods for collaborative filtering algorithms are not able to handle very large datasets [13].

Many clustering algorithms have been utilized in collaborative filtering but recently, using deep learning algorithms has become popular, and one of its new methods is the use of Extreme Learning Machine (ELM) methods for clustering. For the first time, in this work, a combination of the ELM and the Restricted Boltzmann Machine (RBM) for clustering in the recommender systems is discussed. The ELM-based clustering algorithm has been utilized to overcome the data sparsity problem and increase the efficiency of the recommender systems based on precision, accuracy, F-measure, and recall. Also RBM is utilized to find the input weights and biases of ELM.

In other words, at first, they train RBM by utilizing the processed data and then give the trained weights and biases to ELM to perform the clustering operation. Indeed, in this research work, clustering is performed by ELM but RBM is utilized for the purpose of improving the ELM performance. Also this work employs the ELM for two approaches: clustering of the training data and determining the clusters of the test data. In the following, each one of these approaches is discussed separately.

The rest of this paper is set out as what follows. In Section 2, a brief review of the former research works on RS, RBM, and ELM is provided. Section 3 presents the new suggested method of this work.

Section 4 clarifies the experimental data and methods. Lastly, conclusions are outlined in Section 5.

2. Literature Review

In 1993, David Goldberg *et al*. [14] began their studies on the recommender systems, focusing explicitly on the rating structures to present the first recommender system. Thus far, numerous methods were recommended to increment the predicted rating accuracy. Among all these approaches, the collaborative filtering techniques have attracted much interest from the researchers due to their simplicity, and regarded as the most popular method within the recommender system [3].

The collaborative filtering (CF) approaches have two categories: model-based CF and memory-based CF. Recommendations are provided by the memory-based type in terms of the similarities between the items or users, and predict the active user by employing the all user-item database. The main idea of this category is that all the users are likely to buy the items that are the same as the ones previously bought. However, the model-based type uses the user-item database to generate a model off-line, and works on the decreased data, thus helping to overcome the sparsity and scalability problems. The main difference between the memory-based approach and the model-based techniques is that we are not learning any parameters using gradient descent or any other optimization algorithm. The nearest items or users are calculated using Pearson correlation coefficients or cosine similarity, which are only based on the mathematics operations. As with all the existing methods, collaborative filtering has several challenges including data sparsity and scalability. The clustering methods are generally used in the collaborative filtering methods, and can be employed for group users into different clusters to overcome the data sparsity issue [5,8-17].

Bardrul Sarwar *et al.* [11] have developed machine learning techniques to solve the challenges of collaborative filtering for the first-time including the clustering methods, Bayesian network, and rule-based machine learning. This study was the first to find the similarities in the recommender systems using the Pearson correlation coefficient and cosine similarity.

In 2007, using the recommender systems in groups was discussed [18] by Barry Smyth and Anthony Jameson in order to address the confusion of data in group recommendation, attempting to recommend one item to several users rather than recommending one item to only one user.

Koohi and Kiani [12] have improved the data sparsity problem using the fuzzy clustering algorithm and also have used the Pearson correlation coefficients and average methods to find the similarity among the users. Also in [19], these researchers have presented a new technique to discover the neighbor users improving the performance of collaborative filtering.

Meanwhile, many researchers have tried to use heuristic algorithms in order to improve the clustering performance. Katarya [20] has attempted to find the best user recommendation in the recommender systems using the bee colony algorithm, solving the data sparsity problem for the first time. Also Singh and Solanki [21] have presented a study that focus on the film recommender system utilizing the K-means clustering algorithm and the modified cuckoo search algorithm (MCS).

The K-means algorithm is a broadly utilized algorithm for clustering owing to its simple nature, ability to handle numerous data and low implementation time. However, it falls into local optimum due to its randomly generated initial centroids. The algorithm could obtain a global optimal solution in case integrating with the algorithm that was inspired by nature.

Currently, using deep learning methods to enhance the quality of recommendations has also become increasingly common. In [22], Verma *et al.* have presented a study in which collaborative filtering with label consistent Restricted Boltzmann Machine (RBM) has been used.

Behera *et al.* [3] have presented a study in which RBM and fuzzy C-means are used for collaborative filtering. One of the emerging approaches to clustering is the use of extreme learning machines. In [23], for the first time, He *et al.* suggested a clustering method with the extreme learning machine, called unsupervised Elm (US-ELM). [24] proposed a method to extend ELM to cluster via Extreme Learning Machine Auto Encoder (ELM-AE).

2.1. User-based collaborative filtering

The collaborative filtering algorithms are classic personalized recommendation algorithms that are extensively utilized in numerous commercial recommender systems. The collaborative filtering algorithm is based on the precept that since the interests and preferences of the people are stable, so if people have similar interests and preferences; their choices are predictable according to their past preferences. In the user-based CF, the focus is on finding the similarities between the users. This is an automatic prediction method for the user preferences performed by collecting the user's information. This method is in the memory-based category and acts on an n × m user-item matrix. In CF, the first step is to attain the user's history profile indicated as a rating matrix with each entry for an item given by a user's rating of the item. A user-item matrix includes a table each row represents a user of it, a movie is represented by each column, and the number at the intersection of a row and the user's rating value represented by a column. If the user has not yet rated the item, the rating score at this intersection is empty or zero. An instance of a user-item matrix can be seen in Table 1. In the second stage, the association between the users is calculated and the closest neighbors are found. Currently, many noteworthy similarity measurement methods exist in the field. However, the Pearson correlation coefficient is extensively utilized and served as a standard for collaborative filtering since the experimental analyses indicate that the Pearson correlation coefficient performs well for the user-based collaborative filtering in RS, in contrast with other measures of comparing users. Using the following equation, the Pearson relationship between the users a and b is measured [8-10, 12, 19, 25-27].

Table 1. An example of user-item matrix.

	Movie1	Movie2	…	Movie m
User 1	5	0	…	4
User 2	0	2	…	5
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
User n	3	4	…	3

$$sim(a,b)=\frac{\sum_{p\in P}\left(r_{a,p}-\overline{r_a}\right)\left(r_{b,p}-\overline{r_b}\right)}{\sqrt{\sum_{p\in P}\left(r_{a,p}-\overline{r_a}\right)^2}\sqrt{\sum_{p\in P}\left(r_{b,p}-\overline{r_b}\right)^2}} \quad (1)$$

In the above equation, P provides the set of items, $r_{x,s}$ represents the rate of user x on item s, and $\bar{r}_x$ shows the average rating of user x.

The clustering algorithms aim to group the same users into some clusters. Besides the users in the same cluster that the related target user is chosen as the neighboring users. The clustering algorithms have shown that they performs better than similarity measures to find their similar target users. However, the Pearson similarity is generally being used to define the neighboring users; it is also utilized for the rate prediction process. In the final step, the ratings are calculated for each item. The weighted average of the ratings by the neighbors is used in computing the ratings. The following equation is used to predict the rating of the user a for item p [8, 10, 12, 27, 28].

$$pred(a,p) = \overline{r_a} + \frac{\sum_{b \in N} sim(a,b) \times \left(r_{b,p} - \overline{r_b}\right)}{\sum_{b \in N} sim(a,b)} \quad (2)$$

In this equation, $\bar{r}_b$ is the average rating of the user b. Maximizing the average satisfaction is another well-known rate prediction method for calculating the mean of all ratings of item p from the n neighbor users. This equation is as follows [12, 18]:

$$pred(a,p) = \frac{1}{n}\sum_{i=1}^{n} r_{ip} \quad (3)$$

2.2. Restricted Boltzmann machine

The Restricted Boltzmann Machine (RBM) is a generative stochastic neural network introduced by Hinton *et al.* in 1986 and proposing for the binary input data. This network includes two layers, a layer of hidden units (D) with no connections in the same layer and a layer of visible units (V). The visible layer and hidden layer have a symmetric connectivity W, a and c are the bias of each respective layer. RBM has unsupervised learning and over the training phase, it learns the distribution of the likelihood over the input data. Originally, there are two phases for training RBM: (1) forward pass and (2) reconstruction or backward pass. Figure 1 shows the architecture of an RBM [3,24,29].

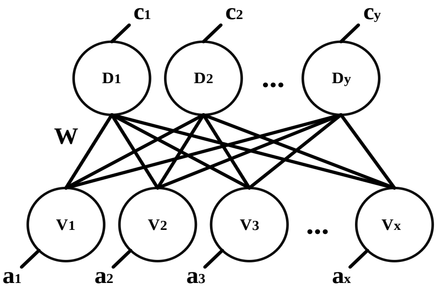

Figure 1. RBM with x visible nodes and y hidden nodes [25].

Suppose that there is M movie, N users and integer rating values between 1 to K. If the same movies are rated by all users, each user can be considered as a single training case for RBM where SoftMax visible units are proportionally connected to a set of binary hidden units. However, for mostly missed ratings, a single RBM should be used for each user. In this case, an RBM only includes visible SoftMax units for the movies rated by that user but every RBM has the same number of hidden units. Thus, the final RBM includes few connections to movies rated by that user. Finally, all weights and biases are tied together. Although a single training case is used for each RBM, when multiple users have rated similar movies, their corresponding RBMs should use the same weights and biases. Figure 2 shows the Restricted Boltzmann Machine with SoftMax visible units and binary concealed units [13].

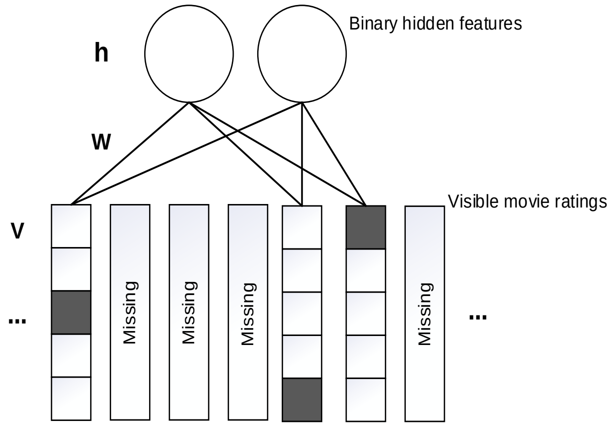

Figure 2. A Restricted Boltzmann machine with binary hidden units and SoftMax visible units [13].

2.3. Extreme Learning Machine

In order to deal with the single-hidden layer feedforward neural network (SLFN) architecture, the extreme learning machine approach was developed. The main feature of the extreme learning machine is that despite the normal comprehending the learning, the hidden layers of SLFNs should not be tuned. The architecture of SLFN is visible in Figure 3, where X and Z, respectively are the input data and output layer array, W represents the input layer's weights matrix, β shows the weights matrix of the hidden layer, and c represents the bias array of the input layer. Normally, within the ELM algorithm, the weight matrix is the uniform distribution in the interval [-1, 1] initializing randomly by sampling all the weight values from a continuous distribution, and during the learning phase, it does not change. If the input weights of ELM are randomly generated, there will inevitably be a set of biases and abnormal (nonoptimal) input weights that affect the behavior of the extreme learning machine. Different algorithms can be used in order to determine the input weights and biases such as the restricted Boltzmann machine, fuzzy C-means, K-means. [29-32]

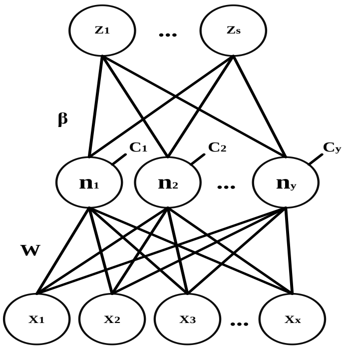

Figure 3. Architecture of a SLFN.

3. Research Methodology

This work aims to offer a combination of extreme learning machines and restricted Boltzmann machines for collaborative filtering in the recommender systems. The research work is overviewed in this section on the proposed method diagram (Figure 4).

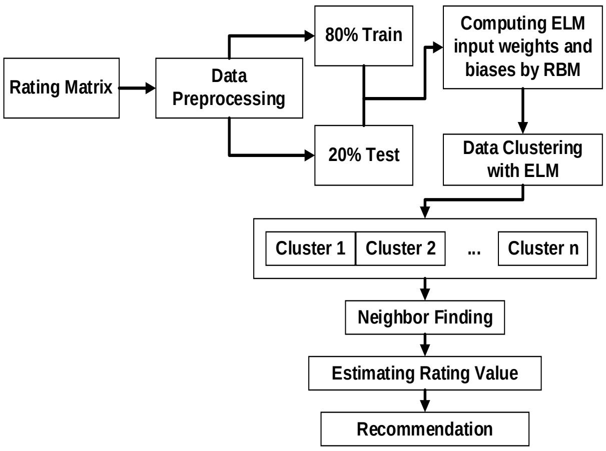

Figure 4. Experimented model.

In the first step, the dataset was divided into sub-sets using the five-fold cross-validation technique, and in each iteration, 80% of the dataset was employed as the training data and the remaining 20% to test the recommendation prediction. Thus there is one testing and four training subsets in each iteration, none of which overlap with each other. Finally, there are five different results based on the five different testing subsets, and we used the average of these results [12,17].

The main idea of this work is to substitute the bias and ELM input weights by the trained weights and biases using RBM. The approach presented in this research work has two parts. First, the ELM biases and input weights are computed by the RBM training, and then the trained biases and input weights are used for clustering with ELM training. Because when the ELM weights are assigned randomly; there is, unavoidably, a group of nonoptimal hidden biases values and input weights probably affecting the performance [30].

In other words, clustering is performed by ELM and, then RBM is used to improve the ELM performance. Also in the second part of this work, ELM is used to determine the clusters of the test data.

In order to better clarify, the discussions of this work are classified into two parts:

1. Clustering of training data with ELM and determine the distance of test data from cluster centers by Euclidean distance function. (RE-Euclidean).
2. Clustering of training data and determine the clusters of the test data with ELM. (RE-ELM).

The similarity of these methods is the clustering of training data with extreme learning machine and their difference is in how they determine the clusters of the test data.

Also we employed RBM with binary hidden units for generating the biases and input weights for ELM. Thus the RBM input data should be a binary vector.

For a better understanding, suppose that a particular user rated M movies. K is the scale of ratings. For example, in MovieLens dataset K is equal to five since the ratings in this dataset are on a scale of 1 (bad film) to 5 (masterpiece). Let V be a $K \times M$ perceived binary indicator matrix with V_i^k=1 when the movie i is rated by the user as k and 0 otherwise. The architecture of restricted Boltzmann machine with binary hidden units and SoftMax visible units can be seen in Figure 2 [13].

For applying RBMs into movie ratings, the first problem is the missing ratings and how to deal efficiently with them. If all of the users rated the same movie, each user can be considered as a single training case for RBM. Suppose that there are N users and M movie. Thus we had M SoftMax visible units that were symmetrically connected to the set of binary hidden units. However, when most of the ratings are missing, a different RBM should be considered for each user. Thus in this research work, RBMs have the same number of hidden units but their visible units have a different number, which depends on the number of each user ratings. For example, if a particular user rated few movies, its corresponding RBM had few connections [13, 35].

Initially, the weight matrix was generated randomly with the number of rows equal to the number of films and the number of columns

equals to the number of hidden layer neurons. In this word, the number of hidden layer neurons is 500, and the learning rate is 0.01.
A sub-weights matrix was considered for each training user that depends on their ratings. The weights matrix is shared between all the users and updated after training each user (Figure 5). If the user has not watched a certain movie, it does not contribute to the update of the weight matrix [35].

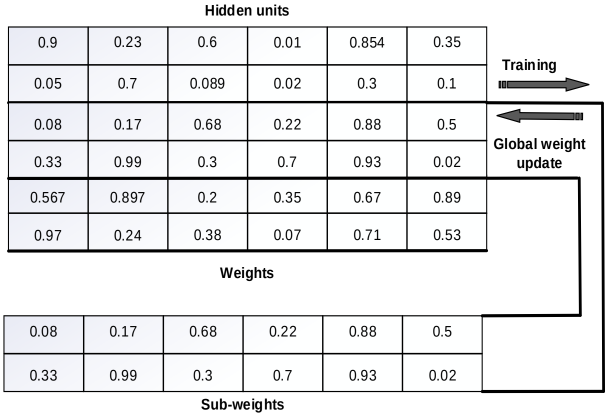

Figure 5. Design of sub-weights for each user.

After finding the clusters, the neighbor users should be found. For determined neighborhood and similarity measures between two users, two well-known approaches have been used; maximizing average satisfaction and Pearson correlation coefficient [12]. Finally, the proposed approach predicts the ratings for similar items and the top-K items are selected for recommendation.

4. Experimental Evaluation

4.1. Dataset

In this work, the MovieLens dataset was used to test the proposed approach. This dataset is a popular dataset considered for evaluating the recommender algorithms in the related articles and [28] compiled by the lens research group at the University of Minnesota. The MovieLens dataset includes 100,000 ratings on a scale of 1(bad film) to 5(masterpiece) of 1,682 movies by 943 users for at least 20 items rated by every user. In this dataset only 6.3% of ratings are accessible. Hence, it is so sparse [12].
In this work, the MovieLense dataset was used in a user-item matrix format. First, the user information and their ratings were merged, and then 80% of it was utilized as the training data and 20% of it was employed as the test data.

4.2. Evaluation metrics

In this work, the recommendation accuracy, precision, recall, and F-measure were utilized to evaluate the results of the experimented methods. In each experiment, the output of the system is a list of items for the particular user with their corresponding predicted ratings. A confusion matrix can measure them. The confusion matrix is observed in Table 2. The Equations 4, 5, 6, and 7 are for using evaluation metrics, where TP represents the true positive, TN is the true negative, FN and FP are the false negative and false positive [3, 12].

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F - measure = \frac{2 \times Recall \times Precision}{\left(Recall + Precision\right)} \tag{7}$$

Table 2. Confusion Matrix.

		Reality	
		Recommended	Not recommended
Test Out-Come	Relevant	TP	FN
	Irrelevant	FP	TN

4.3. Results

In this work, three different clustering methods were used for comparing the performance of the proposed method, which included K-means, fuzzy C-means, and SOM. Also the average and Pearson similarity-based algorithms were used for prediction.
The K-means clustering method is an unsupervised learning algorithm, and aims to partition the unlabeled data into k clusters based on similarity in a way that the similar data is in the same cluster. K defines the number of clusters that require to be created in the process.
Also other data in the different clusters are farther apart. The process flow of K-means is enumerated below:

1. Partition objects into k nonempty subsets
2. Compute seed points as the centroids of the clusters of the current partitioning
3. Assign each object to the cluster with the nearest seed point
4. Go back to Step 2 (stop when the assignment does not change) [40]

Table 3 represents the results of the evaluation of K-means clustering algorithm with the number of various clusters in the MovieLens dataset, in which the best and optimal results are highlighted in bold type.

According to Table 3, it is observed that by increasing the number of clusters, rates of accuracy, recall and F-measure decrease in both the average and Pearson similarity-based methods. The results obtained show that the evaluation metrics have higher rates when the number of clusters is three. The highest accuracy was obtained using the Pearson similarity-based and number of three clusters with a value of 78.27.

Also the highest rates of F-measure and recall belong to the same number of clusters but using the average method in which it equals to 66.72 and 64.75, respectively. However, the highest rate of precision belongs to the average method when the number of clusters is five, in which it equals 69.29.

Table 3. Results of K-means clustering algorithm.

Cluster number	Evaluation metrics	Prediction type	
		Average	Pearson similarity-based
3	Precision	68.82	27.26
	Accuracy	64.41	78.27
	Recall	64.75	1.45
	F-measure	66.72	2.79
5	Precision	69.29	27.24
	Accuracy	63.78	77.62
	Recall	61.56	1.38
	F-measure	65.19	2.67
7	Precision	69.27	26.98
	Accuracy	63.25	77.26
	Recall	59.89	1.34
	F-measure	62.24	2.59

In K-means clustering algorithm, although the Pearson similarity-based technique has a higher accuracy rate than the average method, the precision, recall and F-measure in the average method are higher than the Pearson similarity-based method.

Fuzzy c-means is a technique of clustering that allows any piece of the dataset to belong to more than one cluster. The FCM method assigns membership to each data corresponding to each cluster center based on distance between the cluster center and the data. When the data is near the cluster center, its membership towards the particular cluster center is more. The fuzzy C-means algorithm is implemented in four steps:

1. Suppose that the number of clusters is k.
2. Randomly initialize the clusters.
3. Compute the probability that each data is a member of a particular cluster k.
4. Re-calculate the centroid of the cluster as the weighted centroid given the probabilities of the membership of all data.
5. Iterations continue until convergence or until a user-specified number of iterations has been reached [3,12, 29-32].

Table 4 shows the results of the evaluation of the fuzzy C-mean clustering algorithm with the number of different clusters in the MovieLens dataset, in which the best and optimal results are highlighted in bold.

According to Table 4, it can be seen that with an increasing number of clusters, the rates of accuracy and precision in both average the and Pearson similarity-based methods are decreasing, and the rates of recall and F-measure increase. The results obtained indicate that the highest rate of precision obtained using the average method, when number of clusters is three, equals 69.12.

Also the highest accuracy rate belongs to the same number of clusters but by using Pearson similarity-based method equals 80.27.

The highest rates of recall and F-measure belong to the number of clusters equal to seven using the average method that respectively, equals 66.92 and 67.81.

Table 4. Results of fuzzy C-mean clustering algorithm.

Cluster number	Evaluation metrics	Prediction type	
		Average	Pearson similarity-based
3	Precision	69.12	33.19
	Accuracy	64.82	80.27
	Recall	65.36	4.07
	F-measure	67.20	7.28
5	Precision	68.96	32.70
	Accuracy	64.67	79.89
	Recall	66.07	4.03
	F-measure	67.48	7.21
7	Precision	68.73	32.30
	Accuracy	64.60	79.30
	Recall	66.92	4.30
	F-measure	67.81	7.62

In the fuzzy C-mean clustering algorithm, although Pearson similarity-based method has a higher accuracy rate than the average method, the rates of precision, recall, and F-measure in average method are higher than the Pearson similarity-based method.

Table 5 represents the results of the evaluation of an SOM neural network with a number of different clusters in the MovieLens dataset, in which the best and optimal results are highlighted in bold type.

According to Table 5, with increasing number of clusters, rates of accuracy and precision in both the average and Pearson similarity-based methods decreases, and the rates of recall and F-measure increase.
The results obtained indicate that the highest rate of accuracy obtained using the Pearson similarity-based method and when the number of clusters is 3 * 2 equals 71.08. The highest precision rate is obtained using the average method when the number of clusters is 4 * 2.
Also the highest rates of recall and F-measure belong to the number of clusters equal to 4 * 4 using the average method, which equals 53.21 and 58.10, respectively.

Table 5. Results of SOM neural network.

Cluster number	Evaluation metrics	Prediction type	
		Average	Pearson similarity-based
3*2	Precision	67.08	33.09
	Accuracy	63.07	71.08
	Recall	51.27	2.07
	F-measure	58.12	3.94
4*2	Precision	67.14	32.91
	Accuracy	63.04	70.50
	Recall	51.12	1.87
	F-measure	58.05	3.60
3*3	Precision	64.91	30.58
	Accuracy	62.71	69.70
	Recall	52.14	2.09
	F-measure	57.83	3.95
3*4	Precision	64.27	30.25
	Accuracy	62.08	69.20
	Recall	52.83	2.60
	F-measure	58.00	4.82
4*4	Precision	63.97	29.89
	Accuracy	61.93	69.02
	Recall	53.21	2.76
	F-measure	58.10	5.09

In a SOM neural network, although the Pearson similarity-based method has a higher accuracy rate than the average method, the rates of precision, F-measure and recall in the average method are higher than with the Pearson similarity-based method.
As mentioned earlier, this work employs ELM for two proposed approaches. Table 6 indicates the results of the clustering of training data with ELM, and determines the distance of test data from cluster centers by Euclidean distance function and RE-Euclidean, in which the best and optimal results are highlighted in bold. The results of evaluation of the proposed approach with the RE-Euclidean method show that the highest precision rate is obtained using the Pearson similarity-based with a value of 66.95.

Also the highest rates of accuracy and F-measure belong to the Pearson similarity-based method, which equals 80.04 and 63.06, respectively. The highest recall rate of 62.47 belongs to the average method. In general, although the average method has a higher recall rate compared to the Pearson Similarity-based, rates of accuracy, precision, and F-measure in Pearson similarity-based method are higher than in the average method.

Table 6. Results of RE-Euclidean method.

Evaluation metrics	Prediction type	
	Average	Pearson similarity-based
Precision	35.45	66.95
Accuracy	76.46	80.04
Recall	62.47	59.53
F-measure	45.23	63.06

Table 7 shows the results of clustering of the training data, and determine the clusters of test data with ELM, RE-ELM, in which the best and optimal results are highlighted in bold.
The results of evaluation of the proposed approach with the RE-ELM method indicate a higher rate of all evaluation metrics while using the average method results in the rates of precision, accuracy, recall, and F-measure coming to 80.49, 83.20, 67.84, and 73.62, respectively.

Table 7. Results of RE-ELM method.

Evaluation metrics	Prediction type	
	Average	Pearson similarity-based
Precision	80.49	72.16
Accuracy	83.20	80.12
Recall	67.84	59.95
F-measure	73.62	65.49

By comparing the evaluation results, the RE-ELM with average method not only performs better than the Pearson similarity-based method but also has better performance and the results than the RE- Euclidean method. In the following, the evaluation results of methods used in this work are illustrated in the form of charts.

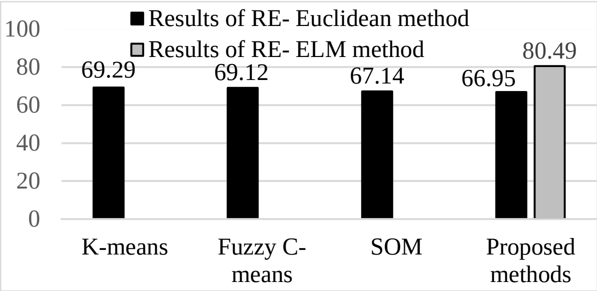

Figure 5. Comparison of precision obtained in each algorithm.

Based on Figure 5, it can be said that the highest precision rate is obtained using the RE-ELM method with a value of 80.49. Also the lowest precision rate belongs to RE-Euclidean method with a value of 66.95.

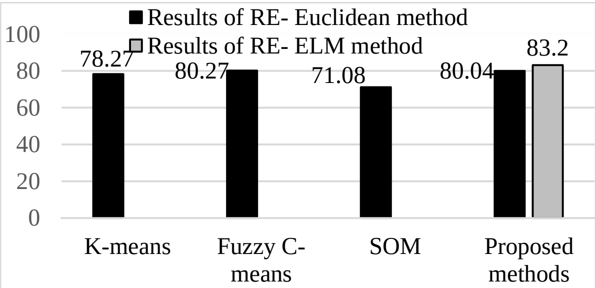

Figure 6. Comparison of accuracy obtained in each algorithm.

According to Figure 6, the highest rate of accuracy is obtained using the RE-ELM method with a value of 83.20. The lowest accuracy rate belongs to the SOM method with a value of 71.08. Also the ELM-Euclidean method has a lower accuracy rate compared to the fuzzy c-means method.

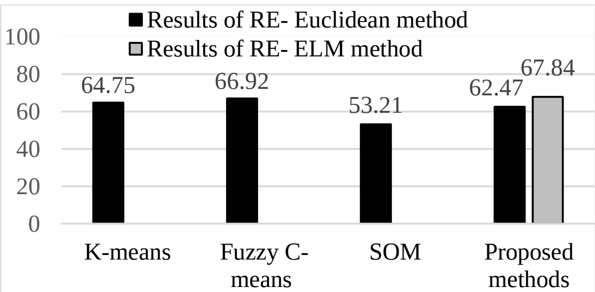

Figure 7. Comparison of recall obtained in each algorithm.

According to Figure 7, the highest rate of recall is obtained using the RE-ELM method with a value of 67.84. The lowest recall rate belongs to SOM method with a value of 53.21.

Also the ELM-Euclidean method has a lower recall rate compared to the K-means and fuzzy c-means methods.

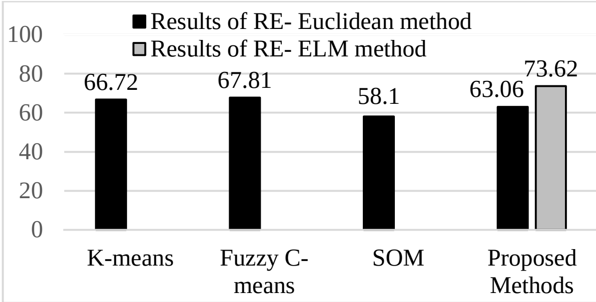

Figure 8. Comparison of F-measure obtained in each algorithm.

According to Figure 8, the highest rate of F-measure is obtained using the RE-ELM method with a value of 73.62. The lowest F-measure rate belongs to the SOM method with a value of 58.1. Also the ELM-Euclidean method has a lower F-measure rate compared to the K-means and fuzzy c-means methods. Based on the results and comparisons, it can be clearly said that the proposed approach with the RE-ELM method performs better than the other methods used in this work and can not only overcome the problem of data sparsity but it can also achieve a higher-performance in the recommender systems.

6. Conclusion

The recommender systems are one of the newest topics in the field of artificial intelligence, with data sparsity being the major challenge of these systems. The clustering methods are used as one of the best available high-performance methods to solve the data sparsity challenge. Recently, most e-commerce companies have used deep learning to improve the quality of their recommendations. Also the extreme learning machine algorithms are one of the newest methods in clustering that are studied with regard to machine learning. For the first time, in this work, a combination of the extreme learning machine and the restricted Boltzmann machine for clustering in the recommender systems was discussed.

The main idea of this work was to substitute the bias and ELM input weights by the trained weights and biases using RBM. First, the ELM biases and input weights were computed by the RBM training. Then the trained biases and input weights were used for clustering with ELM training. Also ELM was used to determine the clusters of the test data. The results of the proposed method evaluated in two prediction methods by employing the average and Pearson correlation coefficient in the MovieLens dataset. Also three different clustering methods were used to compare the performance of the proposed method, which included K-means, fuzzy C-means, and SOM. According to the results obtained, it can be clearly said that the proposed approach with the RE-ELM method performs better than the other methods used in this work, and can not only overcome the problem of data sparsity but it can also achieve higher-performance in the recommender systems.

References

[1] A. Vespignani, “Predicting the behavior of techno-social systems,” Science, Vol. 325, pp. 425-428, 2009.

[2] D. K. Behera, M. Das, and S. Swetanisha, “Predicting users’ preferences for movie recommender system using restricted Boltzmann machine,” in *Advances in Intelligent Systems and Computing*, vol. 711, 2019.

[3] D. K. Behera, M. Das, S. Swetanisha, and B. Naik, “Collaborative filtering using restricted boltzmann machine and fuzzy C-means,” in *Advances in Intelligent Systems and Computing*, vol. 710, 2018.

[4] A. Salah, N. Rogovschi, and M. Nadif, “A dynamic collaborative filtering system via a weighted clustering approach,” Neurocomputing, vol. 175, pp. 206-215, 2015.

[5] L. Lü, M. Medo, C. H. Yeung, Y. C. Zhang, Z. K. Zhang, and T. Zhou, “Recommender systems,” Physics Reports, vol. 519, pp. 1-49, 2012.

[6] M. Salehi, I. Nakhai Kamalabadi and M. B. Ghaznavi Ghoushchi, "An effective recommendation framework for personal learning environments using a learner preference tree and a GA," in *IEEE Transactions on Learning Technologies*, vol. 6, no. 4, pp. 350-363, 2013.

[7] E. Q. Da Silva, C. G. Camilo-Junior, L. M. L. Pascoal, and T. C. Rosa, “An evolutionary approach for combining results of recommender systems techniques based on collaborative filtering,” Expert Systems with Applications, vol. 53, pp. 204-218, 2016.

[8] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, Recommender systems: An introduction, Cambridge University Press, Cambridge, 2010.

[9] 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, pp. 83-94, 2016.

[10] D. Almazro, G. Shahatah, L. Albdulkarim, M. Kherees, R. Martinez, and W. Nzoukou, “A Survey Paper on Recommender Systems”, vol. 11, 2010.

[11] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in *Proceedings of the 10th International Conference on World Wide Web*, WWW 2001, pp. 285-295, 2001.

[12] H. Koohi and K. Kiani, “User based Collaborative Filtering using fuzzy C-means,” Measurement: Journal of the International Measurement Confederation, vol. 91, pp. 134-139, 2016.

[13] R. Salakhutdinov, A. Mnih, and G. Hinton, “Restricted Boltzmann machines for collaborative filtering,” in *ACM International Conference Proceeding Series*, pp. 791–798, 2007.

[14] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, “Using collaborative filtering to Weave an Information tapestry,” Communications of the ACM, vol. 35, pp. 61–70, 1992.

[15] K. Ericson and S. Pallickara, “On the performance of high dimensional data clustering and classification algorithms,” Future Generation Computer Systems, vol. 29, pp. 1024-1034, 2013.

[16] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender systems survey,” Knowledge-Based Systems, vol. 46, pp. 109-132, 2013.

[17] C. F. Tsai and C. Hung, “Cluster ensembles in collaborative filtering recommendation,” Applied Soft Computing Journal, vol. 12, pp. 1417-1425, 2012.

[18] A. Jameson and B. Smyth, “Recommendation to groups,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4321, pp. 596-627, 2007.

[19] H. Koohi and K. Kiani, “A new method to find neighbor users that improves the performance of Collaborative Filtering,” Expert Systems with Applications, vol. 83, pp. 30-39, 2017.

[20] R. Katarya, “Movie recommender system with metaheuristic artificial bee,” Neural Computing and Applications, 2018.

[21] S. P. Singh and S. Solanki, “A Movie Recommender System Using Modified Cuckoo Search,” in *Lecture Notes in Electrical Engineering*, vol. 545, pp. 471-482, 2019.

[22] S. Verma, P. Patel, and A. Majumdar, “Collaborative Filtering with Label Consistent Restricted Boltzmann Machine,” in *2017 9th International Conference on Advances in Pattern Recognition*, ICAPR 2017, pp. 1-6, 2018.

[23] Q. He, X. Jin, C. Du, F. Zhuang, and Z. Shi, “Clustering in extreme learning machine feature space,” Neurocomputing, vol. 128, pp. 88-95, 2014.

[24] G. E. Hinton, “A Practical Guide to Training Restricted Boltzmann Machines,”, Lecture Notes in Computer Science, vol. 7700, pp. 599-619, 2012.

[25] J. Kleinberg and M. Sandler, “Convergent algorithms for collaborative filtering,” in *Proceedings of the ACM Conference on Electronic Commerce*, pp. 1–10, 2003.

[26] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of

the state-of-the-art and possible extensions," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734-749, 2005.

[27] Z. D. Zhao and M. S. Shang, "User-based collaborative-filtering recommendation algorithms on hadoop," in *3rd International Conference on Knowledge Discovery and Data Mining*, WKDD 2010, pp. 478-481, 2010.

[28] M. Ramezani, P. Moradi, and F. Akhlaghian, "A pattern mining approach to enhance the accuracy of collaborative filtering in sparse data domains," Physica A: Statistical Mechanics and its Applications, vol. 408, pp. 72-84, 2014.

[29] A. G. C. Pacheco, R. A. Krohling, and C. A. S. da Silva, "Restricted Boltzmann machine to determine the input weights for extreme learning machines," Expert Systems with Applications, vol. 96, pp. 77-85, 2018.

[30] F. Han, H. F. Yao, and Q. H. Ling, "An improved evolutionary extreme learning machine based on particle swarm optimization," Neurocomputing, vol. 116, pp. 87-93, 2013.

[31] G. Bin Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: A survey," International Journal of Machine Learning and Cybernetics, pp. 107–122, 2011.

[32] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," ACM Transactions on Interactive Intelligent Systems, vol. 5, no. 19, pp. 1-19, 2015.

[33] H. Koohi and K. Kiani, "Two new collaborative filtering approaches to solve the sparsity problem," Cluster Computing, pp. 753–765, 2020.

[34] R. Barzegar Nozari and H. Koohi, "A novel group recommender system based on members' influence and leader impact," Knowledge-Based Systems, vol. 205, 2020.

[35] H. Ben Yedder, U. Zakia, A. Ahmed, and L. Trajković, "Modeling prediction in recommender systems using restricted boltzmann machine," in *2017 IEEE International Conference on Systems, Man, and Cybernetics*, SMC 2017, pp. 2063-2068, 2017.

[36] B. Alhijawi, G. Al-Naymat, N. Obeid, and A. Awajan, "Novel predictive model to improve the accuracy of collaborative filtering recommender systems," Information Systems, vol. 96, 2021.

[37] M. Gupta, A. Thakkar, Aashish, V. Gupta, and D. P. S. Rathore, "Movie Recommender System Using Collaborative Filtering," in *Proceedings of the International Conference on Electronics and Sustainable Communication Systems*, ICESC 2020, pp. 415-420, 2020.

[38] R. Barathy and P. Chitra, "Applying Matrix Factorization in Collaborative Filtering Recommender Systems," in *2020 6th International Conference on Advanced Computing and Communication Systems*, ICACCS 2020, pp. 635-639, 2020.

[39] M. R. I. Sarker and A. Matin, "A Hybrid Collaborative Recommendation System Based on Matrix Factorization and Deep Neural Network," in *2021 International Conference on Information and Communication Technology for Sustainable Development*, ICICT4SD 2021 - Proceedings, pp. 371-374, 2021.

[40] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques. In *Data Mining: Concepts and Techniques*, 2012.

افزایش کارآیی سیستم‌های توصیه‌گر با استفاده از تلفیق یادگیری عمیق و یادگیری افراطی

زهرا نظری، حمیدرضا کوهی* و جواد موسوی

دانشکده فنی و مهندسی، دانشگاه شمال،آمل، ایران.

ارسال ۲۰۲۱/۰۹/۲۸؛ بازنگری ۲۰۲۱/۱۲/۱۲؛ پذیرش ۲۰۲۲/۰۱/۱۶

چکیده:

در دنیای حاضر، با گسترش اینترنت و تکنولوژی‌های وابسته به آن، استفاده از سیستم‌های توصیه‌گر بسیار متداول شده‌است. در این پژوهش هدف اصلی بکارگیری روش‌های نوین خوشه‌بندی مبتنی‌بر یادگیری عمیق جهت حل چالش خلوتی داده‌ها و افزایش کارایی سیستم‌های توصیه‌گر براساس معیارهای صحت، دقت، بازخوانی و مقیاس-اف است. در مدل پیشنهادی این پژوهش نقاط آغازین الگوریتم ماشین یادگیری افراطی توسط ماشین بولتزمن محدود تولید و سپس عمل خوشه‌بندی انجام می‌شود. همچنین، از ماشین یادگیری افراطی برای دو هدف انجام فرآیند خوشه‌بندی داده‌های آموزش و تعیین خوشه‌های داده‌های آزمون استفاده شده‌است. نتایج روش پیشنهادی نیز در دو حالت پیش‌بینی با استفاده از روش‌های میانگین و ضریب همبستگی پیرسون در مجموعه‌داده مووی‌لنز مورد بررسی قرار گرفت. با توجه به نتایج بدست آمده، می‌توان صراحتا بیان کرد که الگوی پیشنهادی این پژوهش، نه‌تنها موجب غلبه‌بر چالش خلوتی داده‌ها شده، بلکه می‌تواند به کارایی بالاتری در سیستم‌های توصیه‌گر دست یابد. نتایج ارزیابی نشان می‌دهند که رویکرد پیشنهادی این پژوهش دارای نرخ بالاتری در معیارهای ارزیابی دقت، صحت، بازخوانی و معیار-اف است که این مقادیر به ترتیب برابر ۸۰/۴۹، ۸۳/۲۰، ۶۷/۸۴ و ۷۳/۶۲ می‌باشند.

کلمات کلیدی: سیستم‌های توصیه‌گر، ماشین یادگیری افراطی، ماشین بولتزمن محدود، خلوتی داده، روش‌های خوشه‌بندی.