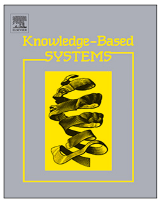
Knowledge-Based Systems 211 (2021) 106545

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



A deep learning based algorithm for multi-criteria recommender systems

Qusai Shambour

Department of Software Engineering, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan

ARTICLE INFO

Article history:
Received 11 March 2020
Received in revised form 15 September 2020
Accepted 16 October 2020
Available online 31 October 2020

Keywords:
Deep learning
Deep autoencoders
Collaborative filtering
Multi-criteria recommender systems

ABSTRACT

Recommender systems have become exceptionally widespread in recent years to deal with the information overload problem by providing personalized recommendations. Multi-criteria recommender systems proved to have more accurate recommendations compared to single-criterion recommender systems as multi-criteria rating reflects the user appreciation of an item in terms of many aspects. On the another hand, deep learning techniques achieve promising performance in many research areas such as image processing, computer vision, pattern recognition and natural language processing. Recently, the application of deep learning in recommender systems have been frequently explored with encouraging results. Accordingly, this paper proposes a deep learning based algorithm for multi-criteria recommender systems in which deep autoencoders are employed to exploit the non-trivial, nonlinear and hidden relations between users with regard to multi-criteria preferences, and generate more accurate recommendations. Experiments on the Yahoo! Movies and TripAdvisor multi-criteria datasets show that the proposed algorithm prove to be very effective in terms of producing more accurate predictions compared with the state-of-the-art recommendation algorithms

1. Introduction

With the rapid development of various resources pertaining to services such as products, books, hotels, movies and restaurants, the explosion of vast flow of information has become increasingly prominent among individuals. Such individuals have to face the problem of information overload and make the right decisions within a vast number of options made available to them. In order to resolve the information overload problem, recommender system emerges to assist the users in making the proper decision by analysing the users' preference information and recommend the items of interest to them according to their preferences [1–4].

Collaborative filtering (CF) is the most commonly used technique in recommender systems. CF generates recommendations based on the users and items. It used ratings to calculate user–user or item–item similarities and produce personalized recommendations based upon them. Regardless of its efficiency, CF techniques suffer from known problems such as cold start, accuracy of predictions and incapability of capturing complex user–item interactions when the rating matrix is very sparse [3,5]. Most of CF techniques utilize the overall rating (single-criterion rating) to produce recommendations, however, this can be considered as a limitation since the single-criterion rating cannot express fine-grained analysis behind the user's preferences. Thus, the user may not get accurate recommendations. Mostly, a user makes a choice of an item by taking into consideration more than one aspect of the item, hence extra aspects of the item may enhance the accuracy of the prediction. For example, in a movie recommender system, some users may like a movie based on its story or acting, while others may like the same movie but for its direction or visuals. To alleviate this problem, multi-criteria recommender systems (MCRSs) have been developed to improve the accuracy of the recommender systems performance by utilizing multi-criteria ratings while producing recommendations. In MCRSs, users can obtain further accurate and personalized recommendations by exploiting their preferences in terms of multiple aspects of items [1,6–8].

Recently, the development of deep learning (DL) technology, as an emerging field of machine learning research, has achieved great success in many emerging fields such as image processing, computer vision, pattern recognition and natural language processing. DL aims to automatically mine multi-level feature representations from data by merging low-level features to form denser high-level semantic abstractions, therefore enable the codification of more versatile abstractions as data representations in the higher layers [9,10]. Lately, DL techniques have been utilized and began to gain huge attention in recommender systems. The reason is that DL techniques can exploit the complex and nonlinear user–item relationships among users and items, which can led to enhancements of the performance in generating accurate recommendations. A number of researchers

E-mail address: Q.shambour@ammanu.edu.jo.

https://doi.org/10.1016/j.knosys.2020.106545
0950-7051/© 2020 Elsevier B.V. All rights reserved.

have proposed the use of DL in building recommender methods but most of these methods make use of only single-criterion ratings in their recommendations. Such recommendations do not take into consideration other multi-criteria ratings. Intuitively, considering such multiple aspects of items may help the deep neural networks to exploit more complex nonlinear user–item relationships, thus generating more accurate recommendations [2,10–13].

In this paper, we propose a deep learning based algorithm for multi-criteria recommender systems. The proposed deep autoencoder-based multi-criteria recommendation algorithm (AEMC) improves the accuracy of recommendations by deeply learning non-trivial, nonlinear and hidden relationships from the multi-criteria data between users and items. The main contributions of this study are as follows:

- A deep autoencoder-based multi-criteria recommendation algorithm (AEMC), which demonstrates the prominence of using multi-criteria ratings and deep learning in recommender systems, is proposed.
- It is shown that considering multiple aspects of items (i.e, multi-criteria ratings) helps the deep autoencoders to exploit more complex, nonlinear and hidden user–item relationships and accurately learn effective user's preferences at fine-grained aspect level, thus improving the accuracy of predictions.
- A set of extensive experiments on two real-world multi-criteria datasets (Yahoo! Movies and TripAdvisor) are conducted in which comparisons are made with other baseline recommendation algorithms to verify the effectiveness of the proposed AEMC algorithm. Experimental results illustrate the superiority of the proposed algorithms with regards to other baseline algorithms.

The organization of this paper is presented as follows. Section 2 briefly reviews the background concepts of deep autoencoders and MCRSs, and some related works on DL based recommender systems. Section 3 describes our proposed algorithm. Section 4 presents experimental methodology and results. Conclusions and future works are shown in Section 5.

2. Background and related works

2.1. Autoencoder

The deep autoencoder is a special type of the deep neural network, whose input vectors and the output vectors have the same dimensionality. It is a nonlinear feature extraction method used for learning a representation of the original data at hidden layers. As shown in Fig. 1, a simple autoencoder contains input, hidden and output layers. The input layer $x \in \boldsymbol{R}^D$ represents the input feature vectors of the original data, the hidden layer denoted as $h \in \boldsymbol{R}^H$ $(H < D)$, has less code dimension than the input that used to represent the transformed feature, and the output layer, $x' \in \boldsymbol{R}^D$, that reconstruct the input data. A simple autoencoder is called as a deep autoencoder when there are more than one hidden layer. An autoencoder is trained using one of the back-propagation functions, usually the stochastic gradient descent method. An autoencoder consists of two main parts: an encoder part that maps the input into the code, and a decoder part that maps the code to a reconstruction of the original input as shown in Eqs. (1) and (2), respectively [14,15].

$$h = g(Wx + b) \tag{1}$$

$$x' = f(\mathrm{W}_1 h + b_1) \tag{2}$$

where $g(\cdot)$ and $f\ (\cdot)$ are nonlinear activation functions, x is the input data, h is the encoded data, $W \in \boldsymbol{R}^{H \times D}$ is the weight matrix

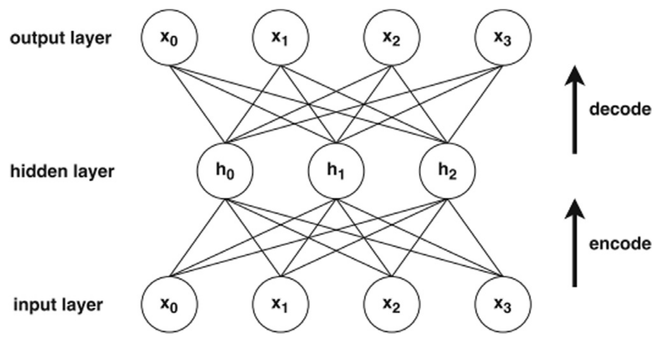


Fig. 1. A simple autoencoder [10].

between the input layer and hidden layer, $W_1 \in \boldsymbol{R}^{D \times H}$ is the weight matrix between the hidden layer and outer layer, b is biased vector for the hidden layer, and b_1 is biased vector for the outer layer.

AutoEncoder is trained to minimize the reconstruction error between x and x'. For this purpose, this study uses the $L1$ loss function (Least absolute deviations), which is formulated as follows:

$$\mathrm{L}\left(x, x'\right) = \left\|x - x'\right\| = \left\|x - f(W_{1g}\left(W_x + b\right) + b_1)\right\| \tag{3}$$

2.2. Multi-criteria recommender systems

The main concept of the recommender system is to build a utility function that can evaluate the fitness of recommending a set of items $i \in I$ to a set of users $u \in U$. The utility function is described as $U \times I \rightarrow R$, where R is a positive integer or real number within a certain range. Accordingly, each user u has to estimate the utility function *R(u, i)* for item i for which *R(u, i)* is not yet known, and then select one or a set of items i that will maximize *R(u, i)* as shown by Eq. (4) [16–18].

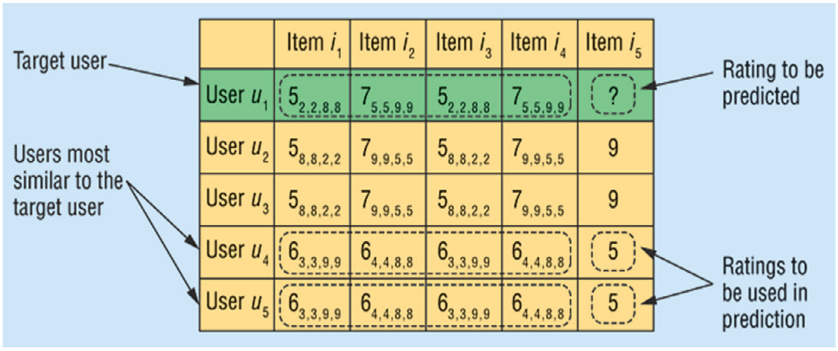
$$\forall u \in U,\ i = \arg\max_{i \in I} R(u, i) \tag{4}$$

The utility function in most current recommender systems considers a single-criteria value that is an overall rating of an item. However, a number of researchers [1,16–21] have acknowledged that the appropriateness of the recommended item for a particular user will certainly rely on more than one utility-related criteria that the user takes into account when making a decision whether an item is appealing and proper for him/her.

This is due to the fact that the additional information delivered by multi-criteria ratings can help to enhance the recommendation performance by accurately modelling users' preferences. Fig. 2 [17] demonstrates the multi-criteria approach in which the overall rating of an item are shown besides other four sub-ratings on different aspects for each item. Notice that even though the overall ratings of the users u_2 and u_3 are almost similar to the overall ratings of the active user u_1, they have rated the different aspects of the item in a completely opposite manner to u_1, thus users u_2 and u_3 actually have opposed preferences to u_1. In fact, users u_4 and u_5 have similar preferences when considering the multi-criteria ratings, and are more appropriate to be used when performing the recommendation [17].

2.3. Deep learning based recommender systems

Considering the latest successes of DL in a variety of domain applications such as natural languages processing, computer vision, image processing, pattern recognition, and machine translation, DL models have been exploited for recommender systems.



	Item i_1	Item i_2	Item i_3	Item i_4	Item i_5
User u_1	$5_{2,2,8,8}$	$7_{5,5,9,9}$	$5_{2,2,8,8}$	$7_{5,5,9,9}$	?
User u_2	$5_{8,8,2,2}$	$7_{9,9,5,5}$	$5_{8,8,2,2}$	$7_{9,9,5,5}$	9
User u_3	$5_{8,8,2,2}$	$7_{9,9,5,5}$	$5_{8,8,2,2}$	$7_{9,9,5,5}$	9
User u_4	$6_{3,3,9,9}$	$6_{4,4,8,8}$	$6_{3,3,9,9}$	$6_{4,4,8,8}$	5
User u_5	$6_{3,3,9,9}$	$6_{4,4,8,8}$	$6_{3,3,9,9}$	$6_{4,4,8,8}$	5

Target user
Users most similar to the target user
Rating to be predicted
Ratings to be used in prediction

Fig. 2. Multi-criteria rating setting [17].

The strength of DL models in discovering hidden features open the doors to produce effective solutions to the limitations of conventional recommender systems such as accuracy, sparsity and scalability. Thus, the adoption of DL models in recommender systems became a favourite and trending topic [2,10–12,22,23].

Accordingly, a number of researchers have employed DL architectures for more enhancing the performance of their recommender systems. For example, Wu et al. [24] developed a real-time recommendation approach by using a deep recurrent neural network (DRNN) to model user's browsing patterns. In the DRNN approach, a user session is represented as a series of web pages, indicating the pathway from the first page to the purchased item. It extracts the most frequent purchase patterns of users and attempts to cut down the path for potential users, so they can quickly get to the pages of their desired products. The DRNN approach has been successfully deployed in the NetEase's e-commerce website. Wu et al. [25] proposed a Collaborative Denoising Auto-Encoder (CDAE) recommendation method. The Denoising Auto-Encoder structure is used to model distributed representations of the users and items through formulating the user–item feedback data. The authors conducted a set of experiments on three real-world datasets and demonstrated that the CDAE method outperforms other state-of-the-art recommendation methods. Paradarami et al. [26] proposed a DL neural network framework where both collaborative and content-based features are used to produce better prediction results than baseline memory-based CF recommendation approaches. The framework have empirically shown outstanding and promising results when compared to baseline memory-based CF approaches. Wei et al. [27] proposed two recommendation models to address the recommendation problems for cold start items. The models integrate a time aware CF model with a DL neural network. The models extract the content features of the items and utilize temporal dynamics of user preferences and item features for the ratings prediction of cold start items. Experiments on a large Netflix dataset showed the effectiveness of the proposed recommendation models in outperforming the traditional recommendation models in solving the problem of cold start items. Zhang et al. [28] proposed a hybrid CF model based on a Semi-AutoEncoder structure. It leverages both the learned semantic rich representations and content information to generate personalized recommendations. Experiments on two real-world datasets demonstrate the effectiveness of the proposed recommendation model in outperforming the state-of-the-art recommendation methods.

In terms of addressing the data sparsity problem of CF approaches, Alfarhood and Cheng [29] propose a DL model as a hybrid CF named DeepHCF. It is composed of Multi-Layer Perceptron and Convolutional Neural Network deep models that were trained together with two sources of data, ratings matrix and item reviews. A prediction layer with factorization machines utilizes the users and items' latent features learned by each model for the final prediction. Experimental results on four different real-world datasets with different levels of sparsity show that DeepHCF achieves better performance when the data sparsity is exceptionally high. Tallapally et al. [30] proposed a deep neural network technique utilize the multi-criteria ratings in recommender systems. The input layer of the network and the loss function are well modified to learn the relationship between multi-criteria ratings and their relevant overall ratings. Experimental results on two real-world datasets demonstrate the enhancement of the proposed model in outperforming state-of-the-art traditional recommendation approaches. Liu et al. [31] proposed a deep hybrid recommender system framework based on auto-encoders (DHA-RS). The DHA-RS integrates neural CF with stacked denoising auto-encoders to effectively learn user and item features from implicit feedback and auxiliary information in order to improve the precision of predictions. Experiments conducted on the real-world dataset shown that the DHA-RS performs better than state-of-the-art recommendation methods.

Furthermore, Fu et al. [32] proposed a CF framework based on deep learning. The framework consists of two stages: (1) learning low-dimensional vectors of users and items; and (2) predicting ratings by employing a feed-forward neural networks to simulate the interactions between users and items. Experimental results on two real-world datasets confirm the effectiveness of the proposed model in outperforming state-of-the-art recommendation methods. Xue et al. [33] introduced an item-based CF method based on deep neural networks named DeepICF. The proposed method use multiple nonlinear layers to effectively model the higher-order item relations amongst items from data. The authors performed a number of experiments on two datasets and demonstrated that the DeepICF method outperforms other state-of-the-art item-based CF approaches. Wang et al. [34] proposed a deep CF model named DCAR to model complex interactions between users and items. The DCAR model employed the autoencoder to obtain the underlying representations of users and items from the implicit feedback matrix as input. Then, an interaction prediction component is designed based on the neural network architecture to predict the corresponding score of user–item pairs. Experiments on several real-world datasets empirically prove the higher performance of DCAR on item recommendation. Hassan and Hamada [21] explored the effectiveness of using artificial neural networks in modelling multi-criteria recommendation problems. The authors designed two different models: a single rating CF, and a multi-criteria CF based on artificial neural networks. They conducted a comparative study to assess the performance of the two models in terms of their accuracy. The empirical results of the study have shown that the proposed multi-criteria CF based on artificial neural networks technique has the highest prediction accuracy than the corresponding single rating CF technique. Kiran et al. [35] proposed a hybrid recommender system using deep learning. The proposed method utilizes embeddings to learn non-linear latent factors for users and items, integrates the DL features with users and items information to produce a hybrid system, and adjusts the learning rates and weight decay. Experimental results on the MovieLens datasets shown outstanding results in terms of predictive accuracy and running time when compared to well known recommendation algorithms.

To summarize, current literature reveals that DL techniques can afford more accurate recommendations than conventional recommendation algorithms. However, there has still been little research work on exploring DL techniques to model MCRSsto better improve their performance. This paper examines the significance of deep autoencoders as a DL technique in improving the prediction accuracy of multi-criteria recommender systems.

3. The proposed algorithm

The proposed deep autoencoder-based multi-criteria recommendation algorithm (AEMC) employs deep feedforward neural networks. As shown in Fig. 3, the AEMC algorithm obtains a raw MC user–item rating matrix $R_{c*x*y,} = \{r_{aui}, a = 1, 2, \ldots c, u = 1, 2, \ldots x$ and $i = 1, 2, \ldots y\}$, c is the number of criteria, x is the number of users and y is the number of items, as inputs, and produces an overall prediction for each item an output.

To avoid overfitting of the training dataset and improve the generalization of the algorithm, we exploited the regularization concept in model training by introducing further information to the cost function, thus, updating the learning algorithm to encourage the network to maintain the weights small [36,37]. Accordingly, the objective loss function of the deep autoencoder model is defined as given by Eq. (5). *L1* regularization is employed to prevent over-fitting on the observed ratings by disregarding the loss of unknown values turned to zero. This means that the contribution of observed ratings is only considered to help improve the performance of the generalization properties of our algorithm in predicting missing values.

$$\operatorname{argmin} \sum_{u=1}^{U} \|r_u - f(r_u)\| + \lambda.\, regularization \tag{5}$$

where r_u is the rating vector for user u consisting of its observed ratings, $f(r_u)$ represents the reconstructed rating vector, and λ is the regularization term.

The AEMC algorithm has four main phases to produce recommendations. In the first phase, the dataset is prepared in order to train and test the deep autoencoder model. Accordingly, the numbers of users and items are extracted from the dataset and arranged in a two dimensional array for each rating criteria, where rows represent the users and columns represent the items. The dataset is split into 80% training data and 20% testing data. To discard the impact of missing values in the AEMC algorithm, they are replaced with zeros.

The second phase is the deep autoencoder construction for each criterion-based user–item matrix with all essential hyperparameters. Architecturally, the form of the autoencoder is a feedforward neural network with an input layer, more than one hidden layer and an output layer. The input and output size is same and is the number of unique items in the dataset. The number of hidden layer is fixed arbitrary to 3, 5 and 7. It is useful that the autoencoder has a smaller hidden layer than the input layer (see Fig. 1). The inner layers use sigmoid as an activation function, as shown by Eq. (6). The outmost layers use ELU as an activation function, as shown by Eq. (7). The autoencoder connexions weights and biases in the network are initialized using normal distributed function with mean of 0.0 and a variance of 0.02, while the biases are set to 0.0 in the beginning. The network parameter optimizer is RMSprop, which is based on the gradient descent optimizer. The optimizer object (RMSprop) is constructed with specific parameters such as the *L1* loss function, the learning rate and weight decay are set to 0.001 and 0.1 respectively. A number of experiments were carried out for diverse values of parameters and the best obtained parameter values are considered to get the optimal results.

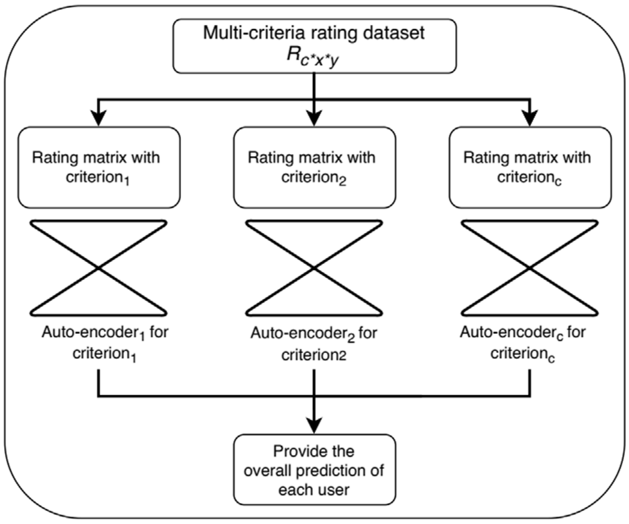

Multi-criteria rating dataset
R_{c*x*y}
Rating matrix with criterion$_1$
Rating matrix with criterion$_2$
Rating matrix with criterion$_c$
Auto-encoder$_1$ for criterion$_1$
Auto-encoder$_2$ for criterion2
Auto-encoder$_c$ for criterion$_c$
Provide the overall prediction of each user

Fig. 3. The framework of the proposed algorithm.

$$g = \frac{1}{1 + e^{-x}} \tag{6}$$

$$ELU(x) = \left\{ \begin{array}{ll} x & if\ x > 0 \\ \alpha(e^x - 1) & if\ x < 0 \end{array} \right\} \tag{7}$$

In the third phase the deep autoencoder is trained and tested using the prepared dataset in phase 1. During the training phase the hidden relations between users' preferences are mined nonlinearly by minimizing the error, and the criterion-based predictions are generated utilizing those relations during the testing phase. By experimentation, the best value for the epochs is 200.

In the final phase, the overall rating prediction r_0 for each item is estimated by an aggregation function of multi-criteria ratings $r_1, r_2, \ldots, r_c$. The arithmetic mean is used as an aggregation function to compute the overall rating prediction by averaging the results of the criteria-based predictions for each item, as shown by Eq. (8). After this phase, the loss is calculated in addition to regularization loss. The loss function is minimized by the RMSprop optimizer.

$$r_0 = \frac{r_1 + r_2 \ldots. + r_c}{c} \tag{8}$$

General representation of AEMC algorithm for **c**-criteria is given in Fig. 3, and pseudo code is given in Algorithm 1.

4. Experimental methodology

4.1. Experimental settings

4.1.1. Data set and evaluation metrics

To validate the performance of the proposed AEMC recommendation algorithm, the Yahoo! Movies MC dataset [38] from the Yahoo! movies website (http://movies.yahoo.com) and the TripAdvisor MC dataset [39] from the TripAdvisor website (http://TripAdvisor.com) have been used. The Yahoo! Movies MC dataset contains 34,800 multi-criteria ratings on four criteria: story, acting, direction and visuals, in addition to an overall rating. The

Algorithm 1 AutoEncoder-based Multi-Criteria Recommendation Algorithm (AEMC)

```
        Input: DataSet $R_{c*x*y}$ // c: rating criteria, x: unique user, y: unique item,
        Output: $P_{x*y}$ // overall predictions of user-item ratings

First phase: Multi Criteria Dataset preparation
1:      Extracting the number of criteria, users and items from $R_{c*x*y}$
2:      For each x*y criterion-based matrix C
        Do
3:          Fill missing values with 0s
4:          Split matrix into random train and test matrices
5:      end For
Second phase: Deep Autoencoder Construction
6:          Define number of hidden layers
7:          Define number of Neurones in each hidden layers
8:          Define number of input and output layers as number of unique items
9:          Initialize the weight of the Autoencoder network
10:         Define the activation function: sigmoid
11:         Define the loss function: Least absolute deviations (L1)
12:         Define the learning rate = 0.01
13:         Define the number of epochs: 200
14:         Define the weight decay = 0.1
15:         Construct the optimizer object (RMSprop) with the learning rate, weight decay.
Third phase: Network Training and Testing
16:         For each Epoch Do
17:             For each User for all items Do
18:             Encode the user rating input into another vector V that has a lower dimension than
                the input (using Eq.1)
19:             Decode the vector V to an output vector that has the same dimension as the input
                (using Eq.2)
20:             Calculate the reconstruction error $\mathcal{L}$ (using Eq. 5)
21:             Update the autoencoder weights and biases after each observation (Stochastic
                Gradient descent) using the chosen optimizer and loss function
22:             end For
23:         end for
24:         Generate criterion-based predictions
Forth phase: Produce overall predictions
25:         For each User for all items Do
26:         Compute overall rating prediction by taking the average of the underlying criterion-based
            predictions for each item
27:         end for
```

dataset was collected from 1716 users on 965 movies. The TripAdvisor MC dataset contains 28,829 multi-criteria ratings on seven criteria: value for money, quality of rooms, location of the hotel, cleanliness of the hotel, quality of check-in, overall quality of services and particular business services, in addition to an overall rating. The dataset was collected from 1039 users on 693 hotels. The rating scale is from 1 to 5 in both datasets. Additionally, ratings in both datasets are randomly divided into two parts as training and testing datasets in a percentage of 80% and 20%, respectively.

To evaluate the prediction performance of the proposed and baseline algorithms, the prediction accuracy were evaluated using the statistical accuracy metrics including the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) metrics as defined by Eqs. (9) and (10) [40]. Note that the lower values of MAE and RMSE are, the higher is the prediction accuracy.

$$MAE = \frac{1}{N}\sum_{u,i}^{N}\left|P_{u,i} - r_{u,i}\right| \tag{9}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,i}^{N}(P_{u,i} - r_{u,i})^2} \tag{10}$$

where N denotes the total number of ratings, $P_{u,i}$ denotes the predicted rating of user u given on item i, and $r_{u,i}$ denotes the actual rating of user u given on item i.

4.1.2. Implementation details

The proposed AEMC recommendation algorithm is implemented in python using Keras 2.1.5 using the Tensorflow backend. The AEMC is implemented for each criterion. The deep autoencoders are trained with a number of fixed hyper-parameter values such as Epoch = 200, learning rate = 0.01 and batch size = 100. The values of these parameters are fixed after several experiments to determine the most optimal value for each parameter. The connexion weights between each adjacent layers are initialized as uniformly randomly and the biases are initialized as ones.

4.1.3. Baseline algorithms

To validate the recommendation accuracy of the proposed AEMC recommendation algorithm, we compare its performance with some well-known baseline recommendation algorithms. Notably, we compare the AEMC recommendation algorithm with a state-of-the-art multi-criteria CF recommendation methods and single rating CF recommendation methods, which are listed as follows:

- The Multi-layer feedforward artificial neural network model, with the multi-criteria ratings as input and the overall rating as an output. The ANN architecture consists of two hidden layers. The first hidden layer consists of ten nodes and the second hidden layer consists of five nodes [41]. The hyper-parameters such as the activation function, learning rate, and number of epoch are the same as those used in the proposed AEMC algorithm.

Table 1
The recommendation accuracy using different number of rating criteria.

# of Rating Criteria	MAE	RMSE
Using 1 criterion	0.6779	0.7473
Using 2 criteria	0.6712	0.7445
Using 4 criteria	**0.6435**	**0.7257**

- The multi-criteria user-based CF recommendation algorithm (MC-UCF) [17].
- The single-criteria user-based CF algorithm (SC-UCF) which employs Constrained Pearson Correlation (CPC) as a similarity measure [42].

4.2. Experimental results

4.2.1. Parameters sensitivity

To obtain the optimal performance of the AEMC algorithm, a number of experiments are applied in order to study the parameters sensitivity and fine-tune them by varying hidden layer numbers, activation functions and optimization algorithms.

In Table 1, we examine the performance of the AEMC model generated using different numbers of the rating criteria (Rating of Acting, Rating of Story, Rating of Visual, Rating of Direction). As shown, the accuracy performance of the proposed algorithm is enriched with increasing number of rating criteria as more precise user preferences are obtained. It can be shown that the proposed algorithm provides the best prediction accuracy when utilizing all of the four rating criteria.

Through reading state-of-the-art research papers, different numbers of encoder layers, different types of activation functions and different types of activation functions may affect the accuracy performance of the proposed algorithm. The role of the deep autoencoder is to mine deeper into the hidden characteristics of the input data, in order to capture the most significant features of the training data. With smaller number of hidden layers, the deep autoencoder is forced to capture the most popular train data features. If too much ability is given to the deep autoencoder, it can learn anything form the data distribution. First of all, the number of encoder layers is set to 1, 3, 5 and 7, respectively, as shown by Table 2. It can be clearly seen in the results that the optimal number of encoder layers is 3, which is the most effective for the performance of the algorithm in terms of the algorithm prediction accuracy.

It can be clearly seen from Table 3 that the performance of the proposed algorithm is affected by five different activation functions, and the variation in experimental results is very clear. In the cases of a linear encoder and decoder, the deep autoencoder learn how copying the input data to the output data without learning useful information. It can be shown that the Sigmoid and ELU pair, used for inner layer and the outermost layers activation functions, presents the best results in terms of both MAE and RMSE when compared with the other pairs.

In order to represent how different optimization algorithms may affect the performance of the proposed algorithm in terms of accuracy, a number of experiments are executed with different optimization algorithms, as shown by Table 4. For each experiment, the optimal encoder layers and activation functions are utilized. It is shown that the RMSprop optimization algorithm provides the best results compared with other algorithms.

4.2.2. Performance comparison

The recommendation accuracy in terms of MAE and RMSE of the proposed AEMC recommendation algorithm and the baseline algorithms on both Yahoo! Movies and TripAdvisor datasets are reported in Tables 5 and 6. It can be noted from the results that the proposed AEMC recommendation algorithm significantly outperformed the baseline algorithms. The average percentages of MAE improvement of the proposed AEMC algorithm in terms of recommendation accuracy on the Yahoo! Movies dataset are 11.73%, 19.97% and 35.51% over the MC-ANN, the MC-UCF and the SC-UCF algorithms respectively. The average percentages of MAE improvement of the proposed AEMC algorithm in terms of recommendation accuracy on the TripAdvisor dataset are 8.36%, 17.32% and 27.32% over the MC-ANN, the MC-UCF and the SC-UCF algorithms respectively. The results are very promising and confirm the improvement of using multi-criteria with DL based algorithms for recommender systems. Considering the results, it is noticeable that extracting non-trivial, nonlinear and hidden relations between users in terms of multi-criteria ratings significantly improves accuracy.

To sum up, the proposed AEMC algorithm improves accuracy of recommendations by (1) considering multiple aspects of items in terms of multi-criteria ratings that provide more complex

Table 2
The sensitivity of AEMC using different number of Encoder/Decoder Layers (Number of Epochs: 200, Activation Function: Sigmoid).

Number of encoder layers	Train	Test	
		MAE	RMSE
1 Encoder/Decoder layer	0.6076	0.7077	0.7774
3 Encoder/Decoder layers	**0.5751**	**0.6435**	**0.7257**
5 Encoder/Decoder layers	0.6505	0.6945	0.7537
7 Encoder/Decoder layers	0.6917	0.7736	0.8179

Table 3
The sensitivity of AEMC using different Activation functions for inners and outer layers.

Activation function for inner layers	Activation function for outermost layer	MAE	RMSE
Sigmoid	ELU	**0.6435**	**0.7257**
Sigmoid	Linear	0.6646	0.7505
Sigmoid	ReLU	0.6560	0.7397
Tanh	ELU	0.6780	0.7534
Softmax	ELU	0.6686	0.7483

Table 4
The sensitivity of AEMC using different optimization algorithms.

Optimization algorithm	MAE	RMSE
RMSprop	**0.6435**	**0.7257**
Adam	0.8185	0.8520
adagrad	3.5184	1.8107
SGD	2.3664	1.5005

Table 5
Recommendation accuracy comparison between the AEMC with baseline algorithms on the Yahoo! Movies dataset.

Algorithm	MAE	RMSE	% of improvement
Proposed AEMC	**0.6435**	**0.7257**	–
MC-ANN	0.7190	0.7834	11.73%
MC-UCF	0.7722	0.8617	19.97%
SC-UCF	0.8720	0.9843	35.51%

Table 6
Recommendation accuracy comparison between the AEMC with baseline algorithms on the TripAdvisor dataset.

Algorithm	MAE	RMSE	% of improvement
Proposed AEMC	**0.8712**	**0.8956**	–
MC-ANN	0.9507	1.0260	8.36%
MC-UCF	1.0537	1.4499	17.32%
SC-UCF	1.1986	1.6515	27.32%

nonlinear user–item relationships to be exploited and learned; (2) its superior capacity in learning the reconstruction of multi-criteria rating matrix, which makes it able to effectively extract and learn the complex, nonlinear and hidden relations between users and items that accurately model users' preferences at fine-grained aspect level, and (3) properly fine-tuning its parameters, which significantly affect its correctness and robustness and have a huge impact on its prediction accuracy [43].

5. Conclusions and future work

This paper proposed a deep learning based algorithm for multi-criteria recommender systems, which captures the non-linear and non-trivial user–item relationships to better learn the user preferences and then improves the accuracy of recommendations. Specifically, a deep autoencoder-based multi-criteria recommendation algorithm is developed to enable the codification of more composite abstractions as data representations in the higher layers, in addition to the multi-criteria preferences of users, to better model and learn the fine-grained user–item interactions. Experimental results have revealed the effectiveness of the proposed AEMC recommendation algorithm in terms of recommendation accuracy with significant enhancement with respect to the state-of-the-art baseline algorithms. Future directions will be to extend the proposed algorithm by adopting other DL-based techniques and incorporating additional information source, and study their effects on the quality of recommendations.

CRediT authorship contribution statement

Qusai Shambour: Conceptualization, Methodology, Software, Investigation, Data curation, Writing - Original draft, Writing - review & editing.

Declaration of competing interest

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S.M. Al-Ghuribi, S.A.M. Noah, Multi-criteria review-based recommender system–the state of the art, IEEE Access 7 (2019) 169446–169468.
[2] R. Mu, A survey of recommender systems based on deep learning, IEEE Access 6 (2018) 69009–69022.
[3] J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang, Recommender system application developments: A survey, Decis. Support Syst. 74 (2015) 12–32.
[4] Q. Shambour, J. Lu, A trust-semantic fusion-based recommendation approach for e-business applications, Decis. Support Syst. 54 (2012) 768–780.
[5] C.C. Aggarwal, Neighborhood-Based Collaborative Filtering, in: Recommender Systems: The Textbook, Springer International Publishing, Switzerland, 2016, pp. 29–70.
[6] Q. Shambour, M. Hourani, S. Fraihat, An item-based multi-criteria collaborative filtering algorithm for personalized recommender systems, Int. J. Adv. Comput. Sci. Appl. 7 (2016) 274–279.
[7] A. Kouadria, O. Nouali, M.Y.H. Al-Shamri, A multi-criteria collaborative filtering recommender system using learning-to-rank and rank aggregation, Arab. J. Sci. Eng. (2019).
[8] Q. Shambour, A user-based multi-criteria recommendation approach for personalized recommendations, Int. J. Comput. Sci. Inf. Secur. 14 (2016) 657–663.
[9] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.
[10] Z. Batmaz, A. Yurekli, A. Bilge, C. Kaleli, A review on deep learning for recommender systems: challenges and remedies, Artif. Intell. Rev. 52 (2019) 1–37.
[11] S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: A survey and new perspectives, ACM Comput. Surv. 52 (2019) 5.
[12] A. Da'u, N. Salim, Recommendation system based on deep learning methods: a systematic review and new directions, Artif. Intell. Rev. (2019).
[13] L. Zheng, A Survey and Critique of Deep Learning on Recommender Systems, University of Illinois at Chicago, 2016.
[14] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, Cambridge, 2016.
[15] L. Deng, D. Yu, Deep Learning: Methods and Applications, Vol. 7, Foundations and Trends® in Signal Processing, 2014, pp. 197–387.
[16] G. Adomavicius, N. Manouselis, Y. Kwon, Multi-criteria recommender systems, in: Recommender Systems Handbook, Springer, US, 2010, pp. 769–803.
[17] G. Adomavicius, Y.O. Kwon, New recommendation techniques for multicriteria rating systems, IEEE Intell. Syst. 22 (2007) 48–55.
[18] K. Lakiotaki, N.F. Matsatsinis, A. Tsoukias, Multicriteria user modeling in recommender systems, IEEE Intell. Syst. 26 (2011) 64–76.
[19] M. Hassan, M. Hamada, A neural networks approach for improving the accuracy of multi-criteria recommender systems, Appl. Sci. 7 (2017) 868.
[20] H. Mohammed, H. Mohamed, Genetic algorithm approaches for improving prediction accuracy of multi-criteria recommender systems, Int. J. Comput. Intell. Syst. 11 (2018) 146–162.
[21] M. Hassan, M. Hamada, Evaluating the performance of a neural network-based multi-criteria recommender system, Int. J. Spatio-Temp. Data Sci. 1 (2019) 54–69.
[22] B.T. Betru, C.A. Onana, B. Batchakui, Deep learning methods on recommender system: a survey of state-of-the-art, Int. J. Comput. Appl. 162 (2017) 17–22.
[23] N. Nassar, A. Jafar, Y. Rahhal, A novel deep multi-criteria collaborative filtering model for recommendation system, Knowl.-Based Syst. 187 (2020) 104811.
[24] S. Wu, W. Ren, C. Yu, G. Chen, D. Zhang, J. Zhu, Personal recommendation using deep recurrent neural networks in NetEase, in: 2016 IEEE 32nd International Conference on Data Engineering, ICDE, Helsinki, Finland, 2016, pp. 1218–1229.
[25] Y. Wu, C. DuBois, A.X. Zheng, M. Ester, Collaborative denoising auto-encoders for top-n recommender systems, in: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, ACM, San Francisco, CA, USA, 2016, pp. 153–162.
[26] T.K. Paradarami, N.D. Bastian, J.L. Wightman, A hybrid recommender system using artificial neural networks, Expert Syst. Appl. 83 (2017) 300–313.
[27] J. Wei, J. He, K. Chen, Y. Zhou, Z. Tang, Collaborative filtering and deep learning based recommendation system for cold start items, Expert Syst. Appl. 69 (2017) 29–39.
[28] S. Zhang, L. Yao, X. Xu, S. Wang, L. Zhu, Hybrid Collaborative Recommendation Via Semi-AutoEncoder, Springer International Publishing, Cham, 2017, pp. 185–193.
[29] M. Alfarhood, J. Cheng, DeepHCF: A deep learning based hybrid collaborative filtering approach for recommendation systems, in: 2018 17th IEEE International Conference on Machine Learning and Applications, ICMLA, Orlando, FL, USA 2018, pp. 89–96.
[30] D. Tallapally, R.S. Sreepada, B.K. Patra, K.S. Babu, User preference learning in multi-criteria recommendations using stacked auto encoders, in: Proceedings of the 12th ACM Conference on Recommender Systems, ACM, Vancouver, BC, Canada, 2018, pp. 475–479.
[31] Y. Liu, S. Wang, M.S. Khan, J. He, A novel deep hybrid recommender system based on auto-encoder with neural collaborative filtering, Big Data Min. Anal. 1 (2018) 211–221.
[32] M. Fu, H. Qu, Z. Yi, L. Lu, Y. Liu, A novel deep learning-based collaborative filtering model for recommendation system, IEEE Trans. Cybern. 49 (2019) 1084–1096.
[33] F. Xue, X. He, X. Wang, J. Xu, K. Liu, R. Hong, Deep item-based collaborative filtering for top-N recommendation, ACM Trans. Inf. Syst. 37 (2019) 33.
[34] J. Wang, N. Gao, J. Peng, J. Mo, DCAR: Deep Collaborative Autoencoder for Recommendation with Implicit Feedback, Springer International Publishing, Cham, 2019, pp. 172–184.
[35] R. Kiran, K. Pradeep, B. Bharat, DNNRec: A novel deep learning based hybrid recommender system, Expert Syst. Appl. 144 (2020) 113054.
[36] J. Brownlee, Better Deep Learning: Train Faster, Reduce Overfitting, and Make Better Predictions, first ed., Machine Learning Mastery, 2020.
[37] J. Lund, Y.-K. Ng, Movie recommendations using the deep learning approach, in: 2018 IEEE International Conference on Information Reuse and Integration, IRI, IEEE, Salt Lake City, UT, USA, 2018, pp. 47–54.
[38] K. Alodhaibi, Decision-guided recommenders with composite alternatives, in: Information Technology, George Mason University, Virginia, 2011.
[39] D. Jannach, M. Zanker, M. Fuchs, Leveraging multi-criteria customer feedback for satisfaction analysis and improved recommendations, Inf. Technol. Tourism 14 (2014) 119–149.
[40] F.O. Isinkaye, Y.O. Folajimi, B.A. Ojokoh, Recommendation systems: Principles, methods and evaluation, Egypt. Inf. J. 16 (2015) 261–273.
[41] S.C. Yücebaş, MovieANN: A Hybrid Approach To Movie Recommender Systems using Multi Layer Artificial Neural Networks, Vol. 5, Çanakkale Onsekiz Mart Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 2019, pp. 214–232.

[42] U. Shardanand, P. Maes, Social information filtering: algorithms for automating "word of mouth", in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM Press, Denver, Colorado, USA, 1995, pp. 210–217.

[43] D.H. Tran, Z. Hussain, W.E. Zhang, N.L.D. Khoa, N.H. Tran, Q.Z. Sheng, Deep autoencoder for recommender systems: parameter influence analysis, in: 29th Australasian Conference on Information Systems, Sydney, Australia, 2018, pp. 1–11.