

Collaborative Filtering and Artificial Neural Network Based Recommendation System for Advanced Applications

Bharadwaja Krishnadev Mylavarapu*

John Deere Co., Moline, IL, USA Email: Bharadwajakrishnadev@johndeere.com

How to cite this paper: Mylavarapu, B.K. (2018) Collaborative Filtering and Artificial Neural Network Based Recommendation System for Advanced Applications. *Journal of Computer and Communications*, **6**, 1-14.

https://doi.org/10.4236/jcc.2018.612001

Received: November 12, 2018 Accepted: December 9, 2018 Published: December 12, 2018

Copyright © 2018 by author and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

http://creativecommons.org/licenses/by/4.0/

Open Access

Abstract

To make recommendation on items from the user for historical user rating several intelligent systems are using. The most common method is Recommendation systems. The main areas which play major roles are social networking, digital marketing, online shopping and E-commerce. Recommender system consists of several techniques for recommendations. Here we used the well known approach named as Collaborative filtering (CF). There are two types of problems mainly available with collaborative filtering. They are complete cold start (CCS) problem and incomplete cold start (ICS) problem. The authors proposed three novel methods such as collaborative filtering, and artificial neural networks and at last support vector machine to resolve CCS as well ICS problems. Based on the specific deep neural network SADE we can be able to remove the characteristics of products. By using sequential active of users and product characteristics we have the capability to adapt the cold start product ratings with the applications of the state of the art CF model, time SVD++. The proposed system consists of Netflix rating dataset which is used to perform the baseline techniques for rating prediction of cold start items. The calculation of two proposed recommendation techniques is compared on ICS items, and it is proved that it will be adaptable method. The proposed method can be able to transfer the products since cold start transfers to non-cold start status. Artificial Neural Network (ANN) is employed here to extract the item content features. One of the user preferences such as temporal dynamics is used to obtain the contented characteristics into predictions to overcome those problems. For the process of classification we have used linear support vector machine classifiers to receive the better performance when compared with the earlier methods.

*Business Analytics Lead, Analytics, PLM COE, PDM.

Keywords

Artificial Neural Network, Support Vector Machine, Recommendation Systems, Cold Start Problems

1. Introduction

1.1. Artificial Neural Network

We can regard this network as a computational model which is based on the configuration as well as purpose of biological neural networks. The mechanism of artificial neural network is human brain processes information which contains a huge amount of associated processing units to facilitate effort simultaneously towards progression information; since, they also produce significant consequences and it consists of the following 3 layers which are shown in **Figure 1** as:

Input layer—The function of the input layer is to accept and input the values of the descriptive characteristic used for apiece examination. The general thing in this layer is number of input nodes which is equivalent to the number of descriptive variables. To converse one or more "concealed layers", the prototype to the network was obtainable by the "Input layer".

Hidden layer—Within the network, the specified conversion to the input values has been applied by the Hidden layers. By using the hidden layer, and arriving curves with the intention of departing as additional concealed nodes or else, the input nodes associated to each node. Through the system of weighted "connections", actual processing is done in hidden layer and it might have the possibility of one or more hidden layers.

Output layer—From concealed layers or else input layer, the connections are received by the output layer. It precedes an output value with the intention of communicate towards the prophecy of the retort patchy.

1.2. Linear Support Vector Machine

Based on the idea of decision planes SVM can define the decision boundaries. The separation among set of objects with irrelevant class memberships is referred as decision plane. The example is given in **Figure 2**.





Figure 2. Linear SVM.

A sample of linear classifier is shown above. Moreover, all classification task are not merely, as well as there is a necessity for more difficult structures in sequence to produce an optimal separation. A hyperlane classifier defines that the classification errands based on depiction sorting out shape to discriminate among substance of dissimilar division memberships. To handle the above mentioned tasks, Support Vector Machines are predominantly suited.

The design beneath displays the fundamental concept back of Support Vector Machines. Here kernels are referred as the unique objects in which it is mapped. The mechanism of reorganize the object is identified as mapping.

2. Review of Literature

Jian Wei, Jianhua He, Kai Chen, Yi Zhou and Zuoyin Tang (2016) introduced a technique as Collaborative Filtering and Deep Learning based Hybrid Recommendation for Cold Start Problem. In several social networks as well as online e-commercial services, Recommender systems (RS) have been used. One among the major advance used for RS is Collaborative Filtering (CF). In this proposed system, to deal with cold start problem, we used hybrid recommendation model which investigate the article contented description well-read since a deep learning as well as be appropriate them towards the time SVD++ CF model. Furthermore, the performance of the proposed model proved excellent prophecy for cold start dilemma when compared to the four active models on behalf of rating of non-cold start items [1].

Mingsheng Fu, Hong Qu, Zhang Yi, Li Lu and Yongsheng Liu (2018) proposed a theory based on A Novel Deep Learning-Based CF Model for Recommendation Systems. We proposed a deep learning method to conquer the setbacks in CF-based technique which emulate an efficient intellectual suggestion by accepting the users as well as items beforehand. To examine efficiency of our proposed system, various experiments are carried out based on 2 standard values (MovieLens 1 M as well as MovieLens 10 M). By the scheme of state-of-the-art, the performance of the system is much better when compared to the preceding techniques on both datasets [2].

Damien Poirier, Francoise Fessant and Isabelle Tellier (2010) developed a system as reducing the Cold-Start problem in Content Recommendation Through Opinion Classification. Here, websites could not eliminate recommender systems because, it has a tool which is used in choosing of products, music, or some other webpage links. Based on the huge quantity of data, the above mentioned type of performance will be chosen. While increasing the User Generated Content, the quantity of data will be increasing automatically due to having the version is Web 2.0. Based on the two factors, a recommender system has been provided with a large textual data. The first step is to construct a user-item-rating matrix, the subjective texts are marked confer to their expressed opinion and the second step is for a collaborative filtering technique, the above mentioned matrix has been used to create recommendation thanks [3].

Reginaldo Aparecido Gotardo, Estevam Rafael and Paulo Roberto Massa Cereda (2013) discussed about Approach to Cold-Start Problem in Recommender Systems in the Context of Web-Based Education. Here, the authors introduced a model to cure cold start problem with intention for Environment Education Web. The two models are Coupled-Learning and Bootstrapping which is used in the proposed method. The data has been increased by concerning algorithms and we can be able to see the different scrutiny on outputs using machine learning which should be collaborate each one erstwhile at the time of starting set of data. By using the above mentioned process, we can have the potential to enhance the performance of recommender system as well as to enhance the starting quantity of information. By using the changes in CBF algorithm, we can easily show aggression at cold start problem. To permit the creation of huge recommendations we need to increase the first level skill, so that here we are using incremental semi-supervised learning to do the process [4].

Ossama H. Embarak (2011) developed a scheme as A Method for Solving the Cold Start Problem in Recommendation Systems. The author introduced the above method by using privacy protected approach. Privacy problem is a big one in recommender systems. Node recommendation and batch recommendation has been proposed for recommendation systems and these systems will be compared with the following models such as Triadic Aspect Method, Naïve Filterbots Method, and MediaScout Stereotype Method. By using the three models as well as two recommendation systems we can easily produce recommendations with the usage of collected works of data's through online. Also, the stages of innovation, accuracy and exposure have been estimated. The performance of all the entire stages which is mentioned above is much better when compared to the three models [5].

Hanafi, Nanna Suryana, Abd Samad Bin Hasan Basari (2005) introduced a scheme as An Understanding and Approach Solution for Cold Start Problem Associated with Recommender System: A Literature Review. Recommender system has been classified into four types based on their scientific model such as Collaborative, Contents Based, Knowledge Based and Demographic filtering. By comparing all these techniques, Collaborative is the major which is often used by the researchers. In this collaborative filtering, a problem will occur which is called as Cold start problem. To overcome the entire dilemma in cold start problem, the authors introduced the proposed method by analyzing the state-of-the-art of the present prototype. Finally, the performance is much better when compared to others [6].

Akash Abhishek and Dr. S. Ganesh Kumar (2018) proposed a scheme on Enhanced Rating Based Recommender System for Resolving Cold Start Problem. To overcome the dilemma in cold start problem, recommender system with an increased rating scheme is introduced. The establishment has been taken for every data in cold-start problem as well as rating process will be finished. Comparison has been taken for proposed method as well as proposed method and the major vital role is to fine a grained scrutiny happening in concealed data's among the two systems. Finally, the performance shows the proposed method is good and efficiency [7].

Jianbo Yuan, Walid Shalaby, Mohammed Korayem, David Lin, Khalifeh Al-Jadda, and Jiebo Luo (2016) discussed Solving Cold-Start Problem in Large-scale Recommendation Engines: A Deep Learning Approach. Based on the reference of CF, recommendation engines have been communicated among users as well as the products. We have to generate it for the new products which are not perfectly suitable for the new users. Here, the authors also introduced a deep learning model to handle the cold start problem. One more advantage in the proposed system is we can able to provide the active CF based recommendation engine lacking of variations in the core of CF. By using Career builder's CF depends on recommendation system we can able to solve the item cold-start problem. Finally, the performance displayed that our proposed system is more efficient and effective method as well as it will enhance the accuracy of the system [8].

Weiwei Zhang, Fangai Liu, Lu Jiang and Daomeng Xu (2017) introduced a system as Recommendation Based on CF by Convolution Deep Learning Model based on Label Weight nearest Neighbor. Here, the proposed system is categorized into three sections as 1) the adjacent neighbor set has the furthermost collision scheduled the objective user is establish through resemblance. 2) By using convolution neural network, the item feature vector has been studied. 3) At last, the prophecy matrix is decayed by the likelihood matrix. The proposed analysis proved that our model is much efficient for accuracy compared with the traditional recommendation algorithm [9].

Libo Zhang, Tiejian Luo, Fei Zhang and Yanjun Wu (2017) proposed a theory as A Recommendation Model Based on DNN. The proposed system which consisting of two parts. The first part is based on quadric polynomial regression model; it uses a feature representative method. It will be more accurate by increasing the traditional matrix factorization algorithm. The second part is to predict the rating scores, the mentioned suppressed characteristics are considered as the input data of the DNN model. The results have been verified that the recommendation performance is much better in this model [10].

Shi, Y., Larson, M. and Hanjalic (2014) A suggested a survey called, Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. For the intention of recommended systems, the U-I matrix is used to set the creature presentation of consumers for products in an assortment. Here, we can determine the recommendation system and we can generate address to them by using CF. In recommender system technology, we can easily detect as well as suggest about the theory based on conservatory of preceding methods and process of integration [11].

G. K. Dziugaite ad D. M. Roy (2015) developed a system based on Neural Network matrix factorization. We use matrix factorization to recuperate the omitted or contaminated entries as well as we can have the potential to written in terms of product of two low-rank matrices. Here, we recuperate the internal item for consumption by using a multi-layer feed-forward neural network. The proposed system will have the merits of graph characteristics. Finally, the performance is good by using the proposed system and it is compared with the existing system [12].

Mohammed Wasid and Vibhor Kant (2015) introduced a technique as A Particle Swarm Approach to Collaborative Filtering Based Recommender Systems through Fuzzy Features. Here, to study consumers weight based on several characteristics, we introduce an algorithm called particle swam optimization algorithm as well as for signifying consumer characteristics we use fuzzy set and it is very efficient in performance. Using the MovieLens dataset, we can calculate the performance of the proposed system [13].

Wei, J., He, J., Chen K., Zhon Y., and Tang Z. (2017) proposed a design called Collaborative filtering and deep learning based recommendation system for cold start items. To remove the contented product characteristics, we use a specific deep neural network SDAE. For the modified content product characteristics into calculation of ratings we use time SVD++ which is used for sequential dynamics of consumer preferences and characteristics of product. By using the two proposed methods, we can estimate as well as compare it with the existing system and analyze that our proposed model performance is much better when compared to others. Finally, we can detect the resolution for cold start problem and it is very efficient and effective [14].

Salah A., Rogovschi N. and Nadif M. (2016) discussed a method as a dynamic collaborative filtering system via a weighted clustering approach. To overcome the existing methods as memory and matrix factorization (MF), we introduced a novel efficient incremental method which is based on weighted clustering system. The proposed method is used to afford a large quality of recommendations which is cost effective as well as based on the quantity of consumers and products, the complexity is low. It is scalable and efficient. Finally, the performance is much good compared to the previous researches. The performance is effective and efficiency [15].

Bishop C. M., developed a technique as Pattern recognition and Machine Learning (2006). In past 10 years, the two factors have experience for extensive improvement such as pattern recognition as well as machine learning. In the proposed method Bayesian method is used to develop from a dedicated recess

towards main stream. Furthermore, we can improve the accuracy of the model as well as the series of exact intrusion algorithms by using Bayesian method. Finally, the performances of the trendy replica are based on kernels which have important conflict on both algorithms as well as applications [16].

Rawat K., Kumar A. and Gautam A. K. (2014) introduced a system as Lower Bound on Naive Bayes Classifier Accuracy in case of noisy data. Here, in for the most part of the professions, the last and one among the major vital stages is classification which includes machine learning, computer vision, etc. For calculating the performance of Naïve Bayes in noise data, we introduce a technique called Lower Bound on Naïve Bayes classifier. Finally, the performance of the system is much better when compared to the existing system [17].

A. Kumar and T. Sairam (2018) discussed a model called Machine Learning Approach for User Accounts Identification with Unwanted Information and data. To detect the false information as well as unnecessary accounts in Facebook in an efficient manner, we proposed the above mentioned model. Our proposed model is based on execution of machine learning as well as it will be work in OSN (online social network). Also, it is used for security purpose while types of text as well as image have been performed through CNN of unendorsed wisdom [18].

3. Proposed System

3.1. System Architecture

Recommender System consists of several classifications. Those are given below.

Content, Collaborative and Hybrid are the major filtering techniques of recommender systems, shown in Figure 3.





From Figure 4 we are using Collaborative filtering (CF) technique. It consists of model and memory based filtering techniques. In one among those techniques we are using Artificial Neural Network. There are several types are Artificial Neural Network algorithms are present. For this process, Multilayer perceptron is used.

3.2. Multilayer Perceptron

The most popular network with feed-forward system

- Back Propagation Algorithm has been applied.
- As a result of having hidden units, Multilayer perceptron can form arbitrary complex decision boundaries.
- Each unit in the first concealed layer imposes a herperplane in the space pattern.
- Each unit in the second concealed layer imposes a hyper region on outputs of the first layer.
- Output layer combines the hyper regions of all units in second layer.

Multi Layer Perceptron is supplementary composite as well as proficient design, since multiple layers of perceptrons are produced significantly as well as consequently by means of the existence of at least one concealed layer which is not associated to the inputs or outputs of the network.

Figure 5 represents the multi layer perceptron which is consists of input layer, hidden layer and output layer. A network of the above mentioned category is classically skilled with supervised learning, bestow to the philosophy draw round in the preceding subsection. Furthermore, a specified learning algorithm intended for MLP networks is the so-called back propagation's algorithm.





3.3. C-SVM Classifier

The minimum error function for the below mentioned category of SVM:

$$1/2 * w^T * w + C \sum_{i=1}^N \varepsilon i$$

Issue to the restraint:

$$Y_i(w^T \varnothing(x_i) + b) \ge 1 - \varepsilon_i$$
 and $\varepsilon_i \ge 0, i = 1, ..., N$

where *C*—Capacity constant, *w*—Vector of coefficients, *b*—Constant, ε_i —Parameters for handling non separable data (inputs).

The *N* guidance cases were labeled by the index *i*. Make a note of $y \in \pm 1$ symbolize the class labels as well as xi symbolize the self-sufficient variables. The kernel ϕ is used to make over information since the input (self-sufficient) towards the characteristic space. The more error is discipline if the C is larger and this must be noted. Hence, to keep away from over fitting, C should be selected in the midst of concern.

3.4. System Architecture

The system architecture for SVM is shown in Figure 6.

4. Implementation and Results

4.1. Tensor Flow Training Analysis for Recommendation System

From **Figure 7**, recommendation system is applied here to check the movie status. The mentioned **Figures 8-11** provide the details about the training dataset using artificial neural network.





ubunutu@ubunutu-HP-Notebook: ~/Documents/recommendation/movie-recommendation 😑 🗈 😣
File Edit View Search Terminal Help
<pre>ubunutu@ubunutu-HP-Notebook:~/Documents/recommendation/movie-recommendation-syst em-master\$ python3 train_nn.py Using TensorFlow backend. A.shape = (671, 9066) 0.9264836311340332</pre>
o (None, 45) learn_rate = 0.1 Epoch 1/20
2018-10-07 15:09:04.337418: I tensorflow/core/platform/cpu_feature_guard.cc:141]
Your CPU supports instructions that this TensorFlow binary was not compiled to
use: AVX2 FMA
128/90003 [] - ETA: 2:05:33 - loss: 7.8534 - acc
256/90003 [] - ETA: 1:17:42 - loss: 7.5499 - acc
384/90003 [] - ETA: 1:01:35 - loss: 7.1007 - acc
512/90003 [] - ETA: 53:25 - loss: 6.8051 - acc:
640/90003 [] - ETA: 48:44 - loss: 6.7045 - acc:
768/90003 [] - ETA: 45:36 - loss: 6.7057 - acc:
896/90003 [
1024/90003 [
1152/90003 [] - ETA: 39:57 - Loss: 6.1006 - acc:
1280/90003 [] - ETA: 38:48 - loss: 5.8890 - acc:
1408/90003 [] - ETA: 37:51 - loss: 5.7981 - acc:
1536/90003 [] - ETA: 37:05 - loss: 5.7079 - acc:

Figure 7. TensorFlow training analysis.

ubunutu@ubu	nutu-HP-Notebook: ~/Documents/recommendation/movie-recommendation 😑 💷 😣
File Edit View	v Search Terminal Help
88448/90003	[=====================================
88576/90003	[=====================================
88704/90003	[=====================================
88832/90003	[=====================================
88960/90003	[=====================================
89088/90003	[=====================================
89216/90003	[=====================================
89344/90003	[=====================================
89472/90003	[=====================================
89600/90003	[=====================================
89728/90003	[=====================================
89856/90003	[=====================================
89984/90003	[=====================================
90003/90003	[==================] - 2018s 22ms/step - loss: 4.4966 -
acc: 0.3366	
Epoch 4/20	
128/90003	[] - ETA: 32:03 - loss: 4.6352 - acc:
256/90003	[] - ETA: 30:58 - loss: 4.8259 - acc:
384/90003	[] - ETA: 30:33 - loss: 4.8951 - acc:
512/90003	[] - ETA: 30:16 - loss: 4.8857 - acc:
640/90003	[] - ETA: 30:09 - loss: 4.7548 - acc:
768/90003	[] - ETA: 30:01 - loss: 4.7877 - acc:
896/90003	[] - ETA: 29:55 - loss: 4.8484 - acc:
1024/90003	[] - ETA: 29:49 - loss: 4.8417 - acc:

Figure 8. TensorFlow training analysis.

4.2. Collaborative Filtering with Neural Network

This is the code work of collaborative filtering using artificial neural network. The training, testing and validation processes are carried out in this section. The neural network exponential model is employed here for the process of validation.

4.3. Collaborative Filtering Results

Final results and movie rating is given.

According to our testing, training and validation process the final recommendation is given by using the collaborative filtering method. In the training

Open 🔻	train_nn.py Save Ξ ~/Documents/recommendation/movie-recommendation-system-master Save Ξ	E 🖨 🖲 😣
import import import from nn	numpy as np time util import Collaborative_Filtering_Neural_Net	
def tra	pin():	
process	trains a neural net and saves snapshots every epoch. Runs 20 epochs or until you $_{1}^{\prime},\ldots$	quit the
	train_mat, val_mat, masks = util.k_cross() A = util.load_data_matrix()	
	<pre>start = time.time() net = Collaborative_Filtering_Neural_Net(train_mat[0], val_mat[0], masks[0]) net.learn_rate=.1 net.construct_model(hidden_layer_pattern = 'exponential') # net.load_model_number_'exponential_one_hot_learn_rate1_lr_0.1_04.hdf5') net.train_model(model_number='exponential_one_hot') print('time taken to train in seconds:', time.time() - start)</pre>	
def tes	<pre>st(model_name = '', test_type = 'validation'):</pre>	
	Gets the accuracy and validation error of a model. This function assumes you have been saving your models ''' train_mat, val_mat, masks = util.k_cross() A = util.load data matrix()	
	<pre>net = Collaborative_Filtering_Neural_Net(train_mat[0], val_mat[0], masks[0]) net.learn_rate=.1 net.construct_model(hidden_layer_pattern = 'exponential') net.load_model(model_name)</pre>	
	pred_scores , true_scores= net.predict_values(test_type = test_type) pred_scores = pred_scores.argmax(axis=1) true_scores = true_scores.argmax(axis=1)	
	Python 🔻 Tab Width: 8 👻 🛛 Ln 52, Col 92	▼ INS

Figure 9. Collaborative filtering program.

Open▼ E Gave E ■ -/Documents/recommendation/movie-recommendation-system-master Save E ■ ■	8		
<pre>net.construct_model(hidden_layer_pattern = 'exponential') # net.construct_model(inn_model_exponential_one_hot_learn_rate1_lr_0.1_04.hdf5') net.train_model(model_number='exponential_one_hot') print('time taken to train in seconds:', time.time() - start)</pre>			
<pre>def test(model_name = '', test_type = 'validation'):</pre>			
Gets the accuracy and validation error of a model. This function assumes you have been saving your models			
train_mat, val_mat, masks = util.k_cross() A = util.load_data_matrix()			
<pre>net = Collaborative_Filtering_Neural_Net(train_mat[0], val_mat[0], masks[0]) net.learn_rate=.1 net.construct_model(hidden_layer_pattern = 'exponential') net.load_model(model_name)</pre>			
pred_scores , true_scores= net.predict_values(test_type = test_type) pred_scores = pred_scores.argmax(axis=1) true_scores = true_scores.argmax(axis=1)			
<pre>#get Accuracy num_correct = np.sum(pred_scores == true_scores) accuracy = num_correct/pred_scores.shape[0]*100</pre>			
<pre>#get MSE error = pred_scores-true_scores mse = np.mean(np.power(error, 2))</pre>			
<pre>print('The {} accuracy of the model is {}%'.format(test_type, accuracy)) print('The {} mean squared error of the model is {}'.format(test_type, mse))</pre>			
<pre>ifname == 'main': train()</pre>			
<pre>tradity test('nn_model_exponential_one_hot_round_2_lr_0.1_08.hdf5', test_type='training') test('nn_model_exponential_one_hot_round_2_lr_0.1_08.hdf5', test_type='validation')</pre>			
Python V Tab Width: 8 V Lh 52, Col 92 V II	12		

Figure 10. Collaborative filtering program.

process the fitting items which we are used as an input are 670. All the input items are trained, tested and validated. And the final collaborative filtering results are calculated.

Figure 11. Collaborative filtering results.

5. Conclusions

Recommendations for complete cold start problem and incomplete cold start problems are the major issues in collaborative filtering mechanism. Here complete cold start states that no ratings are received during the process and incomplete cold start states that more than 0 ratings are received but they are very fewer in counts. To overcome those issues we used three methods in the proposed system. They are collaborative filtering approach, ANN as well as SVM.

For the extraction of content features from the items ANN work is used, while the SVM is responsible for the process of unknown data prediction. It concentrates on the calculation of temporal dynamics and item features. Several experiments are handled in order to test and train the data set. The proposed analysis proved that our method is more improved than the earlier baseline approaches for complete and incomplete cold start problem in CF technique. In future, we have to increase the recommendation of cold start problems and it is necessity to improve the security problem.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Wei, J, He., J.H., Chen, K., Zhou, Y. and Tang, Z.Y. (2016) Collaborative Filtering and Deep Learning Based Hybrid Recommendation for Cold Start Problem. 2016 *IEEE* 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress, Auckland, 8-12 August 2016, 874-877. https://doi.org/10.1109/DASC-PICom-DataCom-CyberSciTec.2016.149

- [2] Fu, M.S., Qu, H., Yi, Z., Lu, L. and Liu, Y.S. (2018) A Novel Deep Learning-Based Collaborative Filtering Model for Recommendation Systems. *IEEE Transactions on Cybernetics*, 1-13.
- [3] Poirier, D., Fessant, F. and Tellier, I. (2010) Reducing the Cold-Start Problem in Content Recommendation through Opinion Classification. 2010 IEEE/ WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Toronto, 31 August-3 September 2010.
- [4] Gotardo, R.A., Rafael, E. and Cereda, P.R.M. (2013) Approach to Cold-Start Problem in Recommender Systems in the Context of Web-Based Education. 12*th International Conference on Machine Learning and Applications*. Miami, 4-7 December 2013, 645-648. <u>https://doi.org/10.1109/ICMLA.2013.199</u>
- [5] Embarak, O.H. (2011) A Method for Solving the Cold Start Problem in Recommendation Systems. 2010 International Conference on Innovations in Information Technology, Abu Dhabi, 5-27 April 2011, 1556-1561.
- [6] Hanafi, M., Suryana, N., Bin, A.S. and Basari, H. (2018) An Understanding and Approach Solution for Cold Start Problem Associated with Recommender System: A Literature Review. *Journal of Theoretical and Applied Information Technology*, 96, 5233-5237.
- [7] Abhishek, A. and Ganesh Kumar, S. (2018) Enhanced Rating Based Recommender System for Resolving Cold Start Problem. *International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)*, 25, 15-18.
- [8] Yuan, J.B., Shalaby, W., Korayem, M., Lin, D., AlJadda, K. and Luo, J.B. (2016) Solving Cold-Start Problem in Large-scale Recommendation Engines: A Deep Learning Approach. arXiv:1611.05480v1.
- [9] Zhang, W.W., Liu, F.G., Jiang, L. and Xu, D.M. (2017) Recommendation Based on Collaborative Filtering by Convolution Deep Learning Model Based on Label Weight nearest Neighbor. 10th International Symposium on Computational Intelligence and Design, Hangzhou, 9-10 December 2017, 504-507. https://doi.org/10.1109/ISCID.2017.235
- [10] Zhang, L.B., Luo, T.J., Zhang, F. and Wu, Y.J. (2017) A Recommendation Model Based on Deep Neural Network. *IEEE Access*, 6, 9454-9463.
- [11] Shi, Y., Larson, M., and Hanjalic, A. (2014) Collaborative Filtering beyond the User-Item Matrix: A Survey of the State of the Art and Future Challenges. ACM Computing Surveys (CSUR), 47, 204-207.
- [12] Dziugaite, G.K. and Roy, D.M. (2015) Neural Network Matrix Factorization. arXiv:1511.06443.
- [13] Wasid, M. and Kant, V. (2015) A Particle Swarm Approach to Collaborative Filtering based Recommender Systems through Fuzzy Features. *Procedia Computer Science*, 54, 440-448.
- [14] Wei, J., He, J., Chen, K., Zhon, Y. and Tang, Z. (2017) Collaborative Filtering and Deep Learning Based Recommendation System for Cold Start Items. *Expert Systems with Applications*, 69, 29-39. https://doi.org/10.1016/j.eswa.2016.09.040
- [15] Salah, A., Rogovschi, N. and Nadif, M. (2016) A Dynamic Collaborative Filtering System via a Weighted Clustering Approach. *Neurocomputing*, **175**, 206-215. <u>https://doi.org/10.1016/j.neucom.2015.10.050</u>
- [16] Bishop, C.M. (2006) Pattern Recognition and Machine Learning. Springer, New York.

- [17] Rawat, K., Kumar, A. and Gautam, A.K. (2014) Lower Bound on Naïve Bayes Classifier Accuracy in Case of Noisy Data. In: Babu, B., et al., Eds., Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012), 28-30 December 2012, Advances in Intelligent Systems and Computing, Vol. 236, Springer, New Delhi, 639-644. https://doi.org/10.1007/978-81-322-1602-5_68
- [18] Kumar, A. and Sairam, T. (2018) Machine Learning Approach for User Accounts Identification with Unwanted Information and Data. *IJMLNCE*, **2**, 119-127.