BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **22**, No 1
Sofia • 2022 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2022-0009

Hy-MOM: Hybrid Recommender System Framework Using Memory-Based and Model-Based Collaborative Filtering Framework

Gina George, Anisha M. Lal

School of Computer Science and Engineering, Vellore Institute of Technology, India
E-mails: gina.george2017@vitstudent.ac.in anishamlal@vit.ac.in

***Abstract**: Lack of personalization, rating sparsity, and cold start are commonly seen in e-Learning based recommender systems. The proposed work here suggests a personalized fused recommendation framework for e-Learning. The framework consists of a two-fold approach to generate recommendations. Firstly, it attempts to find the neighbourhood of similar learners based on certain learner characteristics by applying a user-based collaborative filtering approach. Secondly, it generates a matrix of ratings given by the learners. The outcome of the first stage is merged with the second stage to generate recommendations for the learner. Learner characteristics, namely knowledge level, learning style, and learner preference, have been considered to bring in the personalization factor on the recommendations. As the stochastic gradient approach predicts the learner-course rating matrix, it helps overcome the rating sparsity and cold-start issues. The fused model is compared with traditional stand-alone methods and shows performance improvement.*

***Keywords**: Recommender systems, e-Learning, Personalized, Fused model, Stochastic Gradient Descent.*

1. Introduction

In recent times, there has been a surge of educational resources available over the Internet. This makes the resources dynamic and heterogeneous. Many techniques are integrated to provide a better learning experience and use computer technology and the Web [1]. A considerable transition sees students preferring traditional classroom learning along with e-Learning. For a long period, learning has been done using conventional pedagogical methods. However, with the rapid technological changes, the entire paradigm of learning has seen a significant shift. A recent report by Klynveld Peat Marwick Goerdeler (KPMG) and Google highlights that the EdTech market is expected to touch 9.6 million users by 2021 from 1.6 million users in 2016 [2]. With an overload of resources available, it becomes even more difficult for a learner to make an informed choice over the preferred choice of educational resources available online. Recommender systems become a massive help in such scenarios.

Recommender systems filter and suggest relevant resources to a learner. Recommender systems are defined as software tools and techniques that provide suggestions that aid in the decision-making process in different scenarios, such as what item to buy, which music to listen [3], and which e-Learning resource to enrol for. e-Learning-based recommender systems become increasingly useful since they offer a variety of quality material and flexibility to learners, thereby providing recommendations according to learners' interests and goals.

Traditional recommender systems are either content-based, context-based, or collaborative-based [4]. Content-based recommender systems look into a user's previous transactions and recommend similar items. Context-based recommender systems try to incorporate the context while making recommendations. An example is when a user queries "books for children," and at another time, the same user queries "work-related books," the recommendations need to alter based on the context [5]. Collaborative filtering systems are of two main types: memory-based and model-based [6-9]. The memory-based method is based on mining historical data to find similarities between users for user-based collaborative filtering or between items for item-based collaborative filtering [10]. The model-based method learns the patterns from the historical data and builds the prediction model to predict the unknown ratings. The different model-based approaches are based on clustering, matrix factorization, support vector machine, and stochastic gradient descent. Reference [11] have proposed a model-based approach using Probabilistic Latent Semantic Analysis.

Collaborative filtering methods often have a limitation of cold-start and rating sparsity. Cold-start occurs when there is a new user or a new product for which there is no previous rating. Ratings sparsity is yet another issue in recommender system. This occurs when the number of products are more but ratings may not be given for all the products. Hence, these issues make it difficult to generate personalized recommendations. Latent factor methods (i.e., model-based) can effectively address data sparsity issue. The latent factor model based recommender systems use dimensionality reduction techniques and are considered to be state-of-the-art in recommender systems [2]. Memory-based methods are known to be accurate but not very effective in sparse rating scenarios and are not fast. On the other hand, model-based methods work effectively in sparse rating scenarios and are faster. However, model-based methods are not very accurate. This paper makes use of user-based collaborative filtering, a type of memory-based method and stochastic gradient descent, a type of model-based method. User-based neighbourhood model is used to identify similar users to the target user. Stochastic Gradient Descent (SGD) methods are used mainly to learn unknown variables of the objective function. SGD is one of the most widely used methods in recommender systems [12, 13].

With a plethora of options available, modelling and personalizing the recommendations become more crucial for user [14]. To generate personalized recommendations, recommender systems collect information from the user, particularly the users' interests, preferences, and goals. This information can be collected explicitly through ratings, questionnaires, or implicitly by inferring user behaviour on that particular site [10]. In this paper, learner characteristics like the learning style, knowledge level, and user preference have been considered for personalization in the recommendation. Research in recommender systems widely adopts hybridising either different methods or hybridising ratings with content data [15, 16].

The objectives of this paper are as follows:

- To comprehend circumstantial work related to fusing collaborative filtering and latent factor methods;
- To acquire details of learners focussing on learner's learning style, knowledge level, and preference;
- To generate personalized recommendations of resources to learners by integrating user-based collaborative filtering and stochastic gradient descent.

The rest of the paper is organised as follows. In Section 2, the related work where previous research has been done in recommender systems using collaborative filtering, latent factor models, and hybridizing the two methods are discussed. In Section 3, the proposed architecture and its methodology are explained. Section 4 presents the experimental evaluation of the model. Finally, in Section 5, the conclusion and future work are outlined.

2. Related work

In this section, previous work related to how learner's characteristics are identified, the pre-processing undertaken in recommender systems in the e-learning domain are discussed. Following that, the recommender systems approaches are reviewed in three sub-sections. The neighbourhood based methods (i.e., memory-based) are intuitive and simple to implement [11], while latent factor models (i.e., model-based) provide an alternative to the former method by trying to extract hidden features that explain unobserved ratings. Hybridising the two approaches helps to overcome the limitations of the individual methods.

2.1. Learner's characteristics used for Recommender systems

The proposed system focuses on finding similar learners based on certain learner characteristics. This subsection looks at the previous work done in extracting learner characteristics. Learning style is intended to identify a learner's comfortable way to learn a subject. Learning style can be extracted by applying several questionnaires, including Myers-Briggs Type Indicator, Five-Factor Model, Kolb's learning style model, Felder-Silverman model [17-19]. The Felder-Silverman (FS) model is the most popular learning style model across computer science research [17]. Authors in reference [20] have designed the dimensions of learning style as Competency, Media Preference, Content Preference, Purpose, Attitude, Learning Feeling, Adaptability, tolerance of repeated learning objects and Preference Priority. This paper uses the Index of Learning Style questionnaire [21].

According to FS questionnaire, the different dimensions are Sensory/Intuitive, Visual/Verbal, Active/Reflective, and Sequential/Global [22]. Sensory learners prefer to learn in an orderly manner and are more interested in memorizing facts, while intuitive learners have characteristics like being independent, self-motivated, and innovative. Visual learners prefer to learn through diagrams, graphs, images, while verbal learners prefer to learn through writing down explanations or taking lecture notes [23]. Active learners prefer to learn through doing and working in large groups, while reflective learners prefer to analysing before doing. Finally, sequential learners prefer to solve problems in a sequential, logical manner, while global learners are able to process information in any manner. The work being proposed here, “Hy-MOM: Hybrid Recommender System Framework using Memory-based and Model-based Collaborative Filtering Framework”, draws similarity from previous work, in which the Index of Learning Style questionnaire is used to understand a learner’s learning style characteristics. The proposed method also considers two other characteristics like knowledge level and learning preference which is understood using a registration form.

2.2. Cleaning and pre-processing

Cleaning and Pre-processing is the most needed and important step while preventing anomalies found in data, like missing data and noisy data [24]. Real-life data need to be pre-processed to be used by different machine learning techniques [14]. The data need to be pre-processed into the format required by the collaborative filtering recommendation engine [25]. The different pre-processing in recommender systems include checking for missing values, transformation, calculating similarity measure, and dimensionality reduction [26]. The different similarity measures used for recommender systems include Euclidean measure, Pearson correlation, Jaccard coefficient, and Tanimoto coefficient. Sampling is yet another step in pre-processing, where the full dataset is sampled into training and testing datasets. Recommender systems also need to deal with a dataset that has features that define a high-dimensional space but have very sparse information in that space [27].

2.3. Memory-based collaborative filtering

Some of the e-Learning based recommender systems use traditional methods like memory-based and model-based collaborative filtering. The following subsections give a brief overview of related work in those areas.

Reference [28] considers users and items as heterogeneous individuals and clusters users based on an evolutionary clustering algorithm. Collaborative Filtering (CF) is performed to calculate user similarity within a cluster. Reference [29] combines collaborative filtering and social networking influencing to personalize queries and thereby generates recommendations. A recommender model for e-Learning has been presented in another work by integrating collaborative filtering and association pattern analysis. Weighting learning objects follow cleaning and pre-processing. The similarity between learners is calculated, and then authors have used k-Similar learners for prediction [24]. A Self-Adaptive Learning Technique (SALT) through teaching has been proposed [1]. The framework being proposed suggests lesslets and learning pathways based on social networking and crowdsourcing principles. A recommendation engine for e-Learning has been proposed by hybridising association rules, content filtering, and collaborative filtering [30]. Based on the data sparsity over the user-resource matrix formed, content filtering or rating prediction is performed.

Reference [31] propose a recommender system for suggesting personalized learning objects using learner model and Adaptive Recommendation Module (ARM). The learner model covers personal information, learning style, expertise level, prior knowledge, and performance of the learner. The ARM calculates Euclidean distance to form the neighbourhood and generates rules using apriori algorithm to generate filtered learning objects. Authors in [32] perform item-based collaborative filtering using adjusted cosine similarity and sequential pattern mining to generate recommendations for e-Learning materials. In another paper, the learners are clustered based on learning style by mining learner's sequential patterns and then generating recommendations based on collaborative filtering [33]. There is yet another type of collaborative filtering known as Automated Collaborative Filtering (ACF). ACF systems predict ratings based on models but do not explain how or why a suggestion was made. However, in reference [34] a framework is proposed for including explanations interface to the ACF system. From work reviewed here, a prevalent issue that arises when applying CF is cold-start and rating sparsity.

Hy-MOM uses user-based collaborative filtering under memory-based collaborative filtering to find the neighbourhood of similar users.

2.4. Model-based collaborative filtering

Considering that model-based collaborative filtering overcomes certain limitations of the memory-based approach, this section looks into related work in this area.

Matrix factorization-based collaborative filtering is increasingly considered effective in solving the memory-based collaborative filtering problem of missing data estimation [11, 35]. Deep neural networks are used with dual-regularized matrix factorization to study in detail the textual content to generate more accurate latent factors in recommender systems [36]. To integrate reliability when matrix factorizations are applied, a method using non-negative matrix factorization has been employed, using the known ratings, and it has predicted an estimate of unknown ratings [37]. The cross-validation technique has been implemented to check the error of the proposed model, with the reliability being found as the inverse of the predicted errors.

In a novel approach for a recommendation in the medical domain, sentiment analysis has been performed to gain the emotional offset of users. Latent Dirichlet Allocation is applied to the user reviews to extract user preferences and doctor features [38]. To the resulting matrix, hybrid matrix factorization is done to predict ratings for a doctor and thereby generate personalized recommendations for a user. In another work, a latent factor model has been built by combining ratings and reviews [39]. Authors use the aspects that users are interested in and check the items' polarity. The base Stochastic Gradient Descent (SGD) algorithm has been used by authors in reference [40] to develop a Stratified SGD (SSGD), which represents the weighted sum of the stratum losses. Authors then use the SSGD to create a distributed SGD using MapReduce to scale through millions of elements. Reference [41] discusses that a user's explicit rating is based on two factors: baseline estimate (a value, which is derived using the user bias, item bias, and global mean) and interaction component (describes the relation between users and items). Authors calculate the baseline estimate using stochastic gradient descent and the interaction component using Split Bregman Algorithm. In summary, certain SGD-based approaches observe slower convergence rate, more storage capacity requirement and at times generate noisy gradient estimates of the data. However, SGD-based methods are more accurate for large datasets, and computational time is limited.

Observing some of the merits of stochastic gradient descent from the related works, the proposed work uses stochastic gradient descent under the model-based collaborative filtering.

2.5. Combination of memory-based collaborative filtering with model-based collaborative filtering

This subsection provides an overview of those works that use a combination of both memory-based collaborative and model-based collaborative filtering.

Reference [42] has proposed a regularized multi-embedding-based recommendation model using Weighted Matrix Factorization (WMF), co-liked item embedding, co-disliked item embedding, and user embedding. The WMF is a frequently used collaborative filtering method in recommender systems. Co-liked item embedding suggests what two users have liked as common items. Similarly, co-disliked item embedding occurs when two users commonly disliked items. User embedding suggests that users share similar likes, those items that did not occur for either of those users. In another recent work, collaborative filtering and dimensionality reduction are applied over the songs dataset. The dimensions are reduced to apply rules. Based on the rules, prediction over unknown preferences is done [43]. A recommender system over the movies domain has been proposed in [44] based on clustering ratings using the expectation-maximisation algorithm. Dimensionality reduction is performed using Singular Value Decomposition (SVD) on each cluster. Movie ontology dataset and web crawler are used to get the unique URL for each movie. Reference [8] uses the metadata associated with reviews related to business services to build an artificial neural network. The ratings to be predicted is a multi-label classification problem for which the classification loss function is minimized by applying a stochastic gradient descent algorithm. This hybrid approach is then compared with applying only a collaborative filtering algorithm.

Another work done is designing a framework based on regularized matrix factorization to change the learning rate. This is done by integrating the Deterministic Step Size strategy, Incremental Delta Bar Delta, and the Stochastic Meta Descent. The Gradient Cosine Adaptation is applied to the updating rule to factor in the cosine angle between learning directions during two successive learning epochs [35]. The neighbourhood method and the latent factor models are merged in reference [10]. Initially, author predicts a rating for an unobserved item as a weighted average of ratings by the target learner's ratings. The neighbourhood method is used based on optimizing the global cost function. To this latent factor model, SVD^{++} is applied. It is seen that as the factors being applied during the experiment increase, the Root Mean Square Error (RMSE) decreases but the time per iteration increases. Reference

[45] performes collaborative filtering by applying principal component analysis to estimate missing ratings for existing users. Using recursive rectangular clustering and k-Means clustering, the existing users are clustered. The new user is placed into a cluster and based on the average of actual ratings of existing users in the cluster, a prediction for the new user is made. It is observed that combining these methods has been implemented over music-based, movies-based, business-based data. There is no discussion over the e-Learning domain.

Reference [48] presents a unified model by correcting the predictions given initially by the latent factor models based on the neighbourhood information. The different latent factor methods used are the Regularized Matrix Factorization, Biased Regularized Matrix Factorization, Non-negative Matrix Factorization, and Maximum Margin Matrix Factorization. The corrections are applied based on the user's latest rated items and the item's nearest neighbours. Reference [49] suggests a collaborative recommender system based on matrix factorization. Authors interpret the recommendation using matrix factorization. First, the metadata about items are used to predict latent factors of the models. Next, a shadow model is built containing the latent factor predictors that help identify users' outcomes. Human-understandable explanations are also given for the predicted outcomes of users.

Drawing from the benefits of combining neighbourhood method with latent factor method, Hy-MOM uses a hybridization of memory-based, specifically the user-based collaborative filtering, and model-based, specifically the stochastic gradient descent method.

3. Proposed methodology

The proposed method is a fused model that combines neighbourhood-based (i.e., memory-based) and latent factor model-based (i.e., model-based) filtering outcome thereby:

- The model brings in personalization based on learners' characteristics;
- Cold-start, Rating sparsity issue is overcome with the use of SGD;
- Improved accuracy of the prediction is observed.

The proposed model, Hy-MOM: Hybrid Recommender System Framework using Memory-based and Model-based Collaborative Filtering Framework, uses different indicators like learners' preferences, knowledge level, and learning style to incorporate personalization. This work undergoes two phases to generate recommendations.

Phase 1. Generation of similar learners based on learner's characteristics

The learner's characteristics are extracted, and a learner-learner characteristics matrix is generated. This is then sent for pre-processing. The cleaned matrix is given as input for the User-based Collaborative filtering. A set of similar learners based on learner's characteristics is generated.

Phase 2. Generation of predicted learner-course matrix

The learner-course rating matrix is identified and sent into the stochastic gradient descent module. The entire matrix is predicted as the outcome of this phase.

The outcome of Phase 1 and Phase 2 are fused to provide recommendations and achieve personalization.

Fig. 1 shows the Hy-MOM framework where initially, the user fills in the registration form. From that, certain learner characteristics like Learning Style (LS), Knowledge Level (KL), and PreFerence (PF) of the learner are extracted. Finally, the designed architecture is explained in detail in the following subsections.

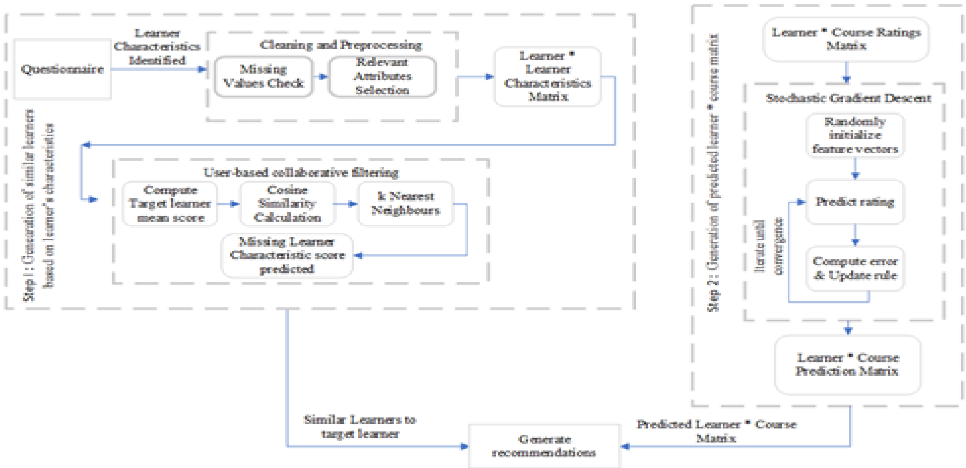

Fig. 1. Hy-MOM framework for recommendation system

3.1. Learner's characteristics identification

Learner's details like profile information and and other learner's characteristics are needed as the basic input to the proposed system. The profile information is obtained through the registration form. Profile information covers basic information, mainly the learner's name, age, gender. The learner's characteristics focussed in this paper are primarily on learning style, knowledge level, and preferences. The Index of Learning style questionnaire method is used to identify the learning style. The knowledge level is to extract the learner's level of knowledge in a course like Beginner, Intermediate, and Advanced. The knowledge level can be found through the registration form. Preferences indicate the learner's preference for a domain of interest and the content type a learner is interested in [46]. This paper covers interests across programming languages, artificial intelligence areas (machine learning, artificial intelligence), and web-related subjects. The learner's preference is understood through the registration form.

3.2. Cleaning and pre-processing

In this paper, the different pre-processing steps include checking for missing values and considering those attributes which are needed for the proposed work.

3.3. User-based collaborative filtering

Through registration, the required learner characteristics are noted. Then, the learner characteristics matrix is sent as input to the user-based collaborative filtering step. The memory-based collaborative filtering can be further divided into user-based collaborative filtering and item-based collaborative filtering. Since this paper focuses on personalizing the recommendations based on learner characteristics, authors have opted for user-based collaborative filtering. The parameters used to find the similarity between the users are based on the learner's learning style, knowledge level, and preference parameters. The k similar users to a given user are found through this step using cosine similarity given by the equation [7]

(1) $$\mathrm{Sim}(U_u, U_v) = \frac{\sum r_{ui} r_{vi}}{\sqrt{\sum r_{ui}^2}\sqrt{\sum r_{vi}^2}},$$

where: $\mathrm{Sim}(U_u, U_v)$ is the degree of similarity between the users u and v; $r_{ui}$is the rating given by user u to item i; r_{vi} is the rating given by user v to item i.

The neighbourhood-based prediction function is defined as seen in the equation [7]

(2) $$r_{uj} = \mu_u + \frac{\sum_{v \in P_u(j)} \mathrm{sim}(u,v) \cdot (r_{vj} - \mu_v)}{\sum_{v \in P_u(j)} |\mathrm{sim}(u,v)|},$$

where: $P_u(j)$ is the set of learners similar to target learner u; sim(u, v) is the degree of similarity between users u and v; r_{vj} is the rating given by user v to item j; μ_u, μ_v is the average rating by user u and user v respectively. Lastly r_{uj} is the predicted rating for user u over item j. The output from this phase is a set of users similar to a target learner based on the learner characteristics.

3.4. Stochastic gradient descent

The ratings given by students for the courses are consolidated to generate the learner-course rating matrix. This goes as input into the stochastic gradient descent algorithm. This method is beneficial for sparse rating matrix, which is mostly the case in the e-Learning domain. It is the SGD algorithm that will then predict the missing rating values. The error between predicting and actual values can be computed by

(3) $$\epsilon_{um} = r_{um} - s(u,m),$$

where: ϵ_{um} is the error rule; r_{um} is the actual rating given by user for an item; $s(u,m)$ is the predicted rating given by user for the item. The algorithm can be repeated through a number of iterations until convergence is reached or until the change in error is very low. The output from the SGD step is the predictions of the entire learner-course ratings. The k similar users got from the learners' characteristics matrix is considered for predicting the rating of the target user.

3.5. Proposed generation of recommendations

The recommendations have been generated using the fused method, in which once similar users are found initially based on learners' characteristics, the learner model is built. It is then passed as input to the fully predicted matrix by applying SGD to the learner-course rating matrix. The weighted average is calculated then to generate the personalized recommendations of courses.

Algorithm 1 shows the procedure of finding similar users using user-based collaborative filtering. Algorithm 2 gives the pseudocode for the model-based stochastic gradient descent. Algorithm 3 shows the procedure for the proposed fused approach, Hy-MOM. The algorithm for finding user similarity is described below.

Algorithm 1. User Based Collaborative Filtering
Input: Set of Learners $L = \{l_1, l_2, l_3, \ldots, l_n\}$
Learners Characteristics C = {Learning style, Knowledge level, Preference}
$L * C$ Matrix
Output: List of similar users to target user based on learner characteristics
Method
Step 1. for $l \in L$, $c \in C$, **do**
Step 2. Compute cosine similarity between learners using Equation (1)
Step 3. Filter k nearest neighbours
Step 4. Predict missing learner characteristics score using Equation (2)
end for
Step 5. Output set of similar users

The algorithm for predicting learners course matrix is described below.

Algorithm 2. Stochastic Gradient Descent
Input: Learners * Course Ratings Matrix
Output: Predicted rating of unobserved entries in Learner * Course ratings matrix
Method
Step 1. Repeat until convergence
Step 2. Randomly initialize user-feature, item-feature vectors
Step 3. Predict unobserved rating using equation
Step 4. Compute error and update rule using Equation (3)
end Repeat
Step 5. Output the entire Predicted Learner * Course Ratings Matrix

Algorithm 3 shows the procedure for the proposed fused model for generating recommendations for courses.

Algorithm 3. Hy-MOM: Hybrid Recommender System Framework Using Memory-Based and Model-Based Collaborative Filtering Framework
Input: Set of similar users to target learner based on learner characteristics
Learners * Courses ratings matrix (existing and predicted using Algorithm 2)
Output: Set of personalized course recommendations
Method
Step 1. for similar learners to target learner **do**
Step 2. Extract the courses rated by the similar learners but not rated by target learner
Step 3. **for** course not rated by target learner **do**
Predict rating by calculating average rating based on similar learners'rating on the course
end for
Repeat until end of matrix
if predicted rating is greater than 3 **then**
Step 4. Select the course
Step 5. **end if**
Step 6. **end Repeat**

Step 7. **endfor**

Step 8. Output the personalized recommendations

The recommendations are done based on the learner's similarity and ratings. The advantage of the proposed method is providing personalized recommendations by finding similar learners based on specific characteristics of all learners, mainly knowledge level, learning style, and learner preference. When ratings are not much available at the higher education level, SGD predicts the missing rating values, thereby aiding to overcome the rating sparsity. With learners required to fill in the registration form, some details are obtained for new users too, which aids in avoiding the cold-start issue.

4. Experimental evaluation

The two inputs needed are the learner characteristics and learner-course ratings. The learner characteristics include knowledge level, preference and the learning style. This is obtained by means of the registration form. Learner-course ratings is obtained by asking the students to rate the courses they have taken. The distribution of the course ratings is shown in Fig. 2. The ratings are on a scale of 1 to 10 with 1 given as the lowest rating and 10 as the highest rating. Majority of the courses are given a rating between 2 and 4. The number of learners taken for the study is 100.

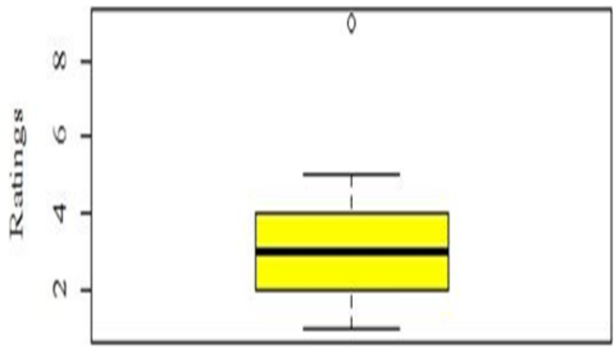

Fig. 2. Distribution of course ratings

The distribution of average ratings per user is shown in Fig. 3. It becomes clear that average ratings are between 2.5 and 4.0. The similarity between users in the user-based collaborative filtering is found by applying cosine similarity. The central and Z-score normalizes the data.

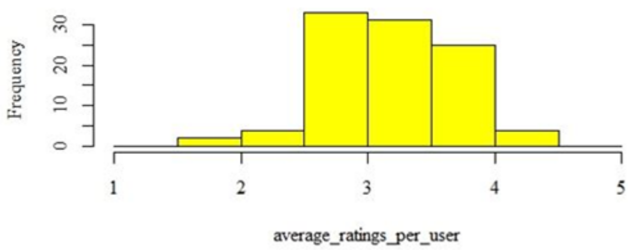

Fig. 3. Average ratings per user

Hy-MOM performance is compared with existing approaches, mainly the User-Based Collaborative Filtering (UBCF) and the Stochastic Gradient Descent (SGD). The existing UBCF is considered with three types of normalization. UBCF_N_C gives an idea of the user based collaborative filtering with null normalization. UBCF_C_C gives an idea of the user based collaborative filtering with central score normalization. UBCF_Z_C highlights the application of user-based collaborative filtering with Z-score.

Table 1. RMSE-MSE-MAE values

No of neighbours = 4		
Performance	RMSE	MSE
UBCF_N_C	2.01377	4.05527
UBCF_C_C	1.991	3.965
UBCF_Z_C	1.911	3.654
SGD	2.007	4.032
Hy-MOM	1.77208	3.14027
No of neighbours = 6		
UBCF_N_C	1.99685	3.9874
UBCF_C_C	1.84968	3.4213
UBCF_Z_C	1.66633	2.7766
Hy-MOM	1.64861	2.7179
No of neighbours = 8		
UBCF_N_C	1.9901	3.9605
UBCF_C_C	1.9442	3.7799
UBCF_Z_C	1.6267	2.6462
Hy-MOM	1.6229	2.6340

Root Mean Squared Error (RMSE) is an evaluation metric that measures the magnitude of the error [36, 47] and is shown in the equation

(4) $$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|^2}\,,$$

where: y_i is the actual rating; $\hat{y}_i$ is the predicted rating; n is the number of users.

From Table 1, the Hy-MOM method performs the best out of all the methods. UBCF_Z_C score performs better than UBCF_C_C in terms of RMSE. Neither the memory-based model nor the model-based method include personalization. The fused method involves using learner characteristics while generating recommendations. Hy-MOM's main advantages include learner's knowledge level, learning style and preferences to get other similar learners to the target learner, overcoming cold-start, and rating sparsity issue.

Fig. 4 shows the performance of the algorithms graphically in terms of RMSE. UBCF_N_C and UBCF_C_C have relatively similar performance while normalizing with z score helps to perform better than without normalization and with using central score normalization. SGD performs similar to user-based collaborative filtering without normalization. Hy-MOM has a comparatively low RMSE. Additionally, it includes the factors for personalization.

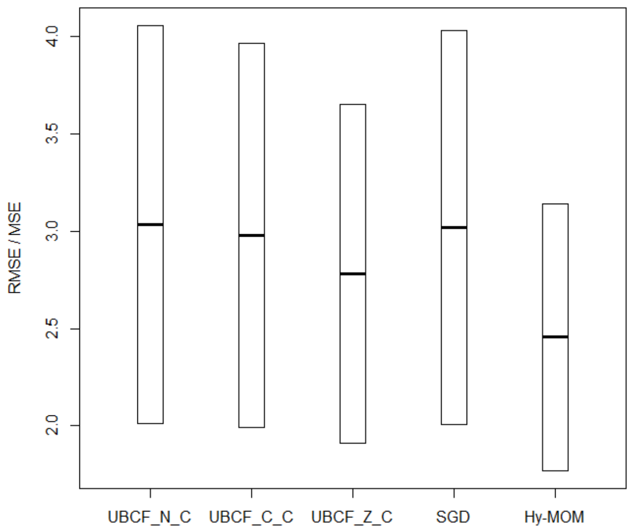

Fig. 4. Comparison of algorithms vs RMSE

4.1. Discussion

The method being proposed is helpful in situations where the availability of ratings is rare, like on courses provided by higher education institutes. The stochastic gradient descent approach helps to predict such missing rating values. The method also helps in systems when similar learners need to be found for a new user who enters the system.

4.2. Limitations

However, certain limitations of the method being proposed do exist. A person is dynamic with the person's characteristics changing and evolving over time. In our work here, only a few learner characteristics have been considered, so the extent of achieving personalization is limited. To overcome this limitation, future work can factor in more characteristics related to a learner. The method of data collection in our work is primarily using the questionnaire method. However, there exists limitations using this method. Students need not give genuine responses concerning their knowledge level, preference, and rating on the different courses. As authors in [50] find, the quality of response from students can be low. Students do not essentially feel the need to give honest answers as they have a perception that answering the questionnaire does not really affect them. It is not necessary that every student be interested on the topic that the questionnaire covers. This may make a student to skip answering [51] few questions or to answer in a fun manner without giving serious thought to each question. A student's concentration may not be fully into answering the complete questionnaire. Experiencing low response rate is also commonly seen with questionnaire method [52]. In the work done here, we too experienced the same. Additionally, employing questionnaire method does not give a chance for clarifying respondents' doubts over any question [53]. All these drawbacks pertaining to the quality of students' responses using the questionnaire method affects performance of the proposed method. These limitations of using questionnaire can be overcome by using other sources of data like applying mining techniques, extracting browsing behaviour, and other artificial intelligence techniques.

5. Conclusion

In conclusion, authors have generated recommendations as per the traditional memory-based method, model-based method, and Hy-MOM method. The stand-alone stochastic gradient descent method performs better than the stand-alone user-based collaborative filtering. The proposed method has relatively low RMSE, with an essential aspect of the personalization factor being included. Based on user similarity and based on stochastic gradient descent, personalized recommendations are generated. The problems of cold-start, rating sparsity is handled in this fused approach since the stochastic gradient approach predicts all the courses' ratings. Certain limitations of this work include the need to get learners' ratings over different courses. Also, the execution time of the SGD algorithm increases with the number of learners increasing. Though the indicators used in this work help to achieve personalization, there is a possibility for the learners not to give genuine responses while using the questionnaire method. Future work includes the need to apply other machine learning algorithms, the need to mine learning management system data to understand learner behaviour and thereby generate the learners' characteristics to be used for personalization. The technique of ontology can be integrated into the Hy-MOM model.

References

1. Karataev, E., V. Zadorozhny. Adaptive Social Learning Based on Crowdsourcing. – IEEE Transactions on Learning Technologies, Vol. **10**, April 2017, No 2, pp. 128-139.
2. KPMG & Google (2017). Online Education in India: 2021. Accessed 21 Janurary 2019. **https://assets.kpmg.com/content/dam/kpmg/in/pdf/2017/05/Online-Education-in-India-2021.pdf/**
3. Ricci, F., L. Rokach, B. Shapira. Recommender Systems: Introduction and Challenges. – Recommender Systems Handbook, Boston, MA, USA: Springer, 2015, pp. 1-34.
4. Wan, S., Z. Niu. A Hybrid e-Learning Recommendation Approach Based on Learners' Influence Propagation. – IEEE Transactions on Knowledge and Data Engineering, January 2019.
5. Adomavicius, G., A. Tuzhilin. Context-Aware Recommender Systems. – Recommender Systems Handbook, Boston, MA, USA: Springer, 2011, pp. 217-253.
6. Ren, L., W. Wang. An SVM-Based Collaborative Filtering Approach for Top-N Web Services Recommendation. – Future Generation Computer Systems, Vol. **78**, January 2018, pp. 531-543.
7. Aggarwal, C. Recommender System the Textbook. Switzerland, Springer International Publishing, 2016.
8. Paradarami, T. K., N. D. Bastian, J. L. Wightman. A Hybrid Recommender System Using Artificial Neural Networks. – Expert Systems with Applications, Vol. **83**, October 2017, pp. 300-313.
9. Fazeli, S., B. Loni, H. Drachsler, P. Sloep. Which Recommender System Can Best Fit Social Learning Platforms? – In: Proc. of European Conference on Technology Enhanced Learning, Springer, Cham, 2014, pp. 84-97.
10. Koren, Y. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. – In: Proc. of 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2008, pp. 426-434.

11. H o f m a n n, T. Latent Semantic Models for Collaborative Filtering. – ACM Transactions on Information Systems (TOIS), Vol. **22**, January 2004, No 1, pp. 89-115.
12. K o r e n, Y., R. B e l l, C. V o l i n s k y. Matrix Factorization Techniques for Recommender Systems. – Computer, Vol. **42**, August 2009, No 8, pp. 30-37.
13. Y u, H. F., C. J. H s i e h, S. S i, I. D h i l l o n. Scalable Coordinate Descent Approaches to Parallel Matrix Factorization for Recommender Systems. – In: IEEE 12th International Conference on Data Mining, IEEE, 2012, pp. 765-774.
14. L o p s, P., M. De G e m m i s, G. S e m e r a r o. Content-Based Recommender Systems: State of the Art and Trends. – Recommender Systems Handbook, Boston, MA, USA, Springer, 2011, pp. 73-105.
15. K o n s t a n, J., M. E k s t r a n d. Introduction to Matrix Factorization and Dimensionality Reduction. – Matrix Factorization and Advanced Techniques, 2018. **https://www.coursera.org/lecture/matrix-factorization/introduction-to-matrix-factorization-and-dimensionality-reduction-ncbvP**
16. J a h r e r, M., A. T ö s c h e r, R. L e g e n s t e i n. Combining Predictions for Accurate Recommender Systems. – In: Proc. of 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2010, pp. 693-702.
17. F a t a h i, S., H. M o r a d i, L. K a s h a n i-V a h i d. A Survey of Personality and Learning Styles Models Applied in Virtual Environments with Emphasis on e-Learning Environments. – Artificial Intelligence Review, Vol. **46**, October 2016, No 3, pp. 413-429.
18. R a n i, M., R. N a y a k, O. P. V y a s. An Ontology-Based Adaptive Personalized e-Learning System, Assisted by Software Agents on Cloud Storage. – Knowledge-Based Systems, Vol. **90**, December 2015, pp. 33-48.
19. T r u o n g, H. M. Integrating Learning Styles and Adaptive e-Learning System: Current Developments, Problems and Opportunities. – Computers in Human Behavior, Vol. **55**, February 2016, pp. 1185-1193.
20. W a n, S., Z. N i u. An e-Learning Recommendation Approach Based on the Self-Organization of Learning Resource. – Knowledge-Based Systems, Vol. **160**, November 2018, pp. 71-87.
21. S o l o m a n, B. A., N. C a r o l i n a, R. M. F e l d e r. Index of learning Styles Questionnaire. – Learning, 1996, pp. 1-5.
22. O u f, S., M. A. E l l a t i f, S. E. S a l a m a, Y. H e l m y. A Proposed Paradigm for Smart Learning Environment Based on Semantic Web. – Computers in Human Behavior, Vol. **72**, July 2017, pp. 796-818.
23. P r o v i t e r a, M. J., E. E s e n d a l. Learning and Teaching Styles in Management Education: Identifying, Analyzing, and Facilitating. – Journal of College Teaching & Learning, Vol. **5**, January 2008, No 1, pp. 69-78.
24. B o u r k o u k o u, O., E. El B a c h a r i, M. El A d n a n i. A Recommender Model in e-Learning Environment. – Arabian Journal for Science and Engineering, Vol. **42**, February 2017, No 2, pp. 607-617.
25. T a r u s, J. K., Z. N i u, D. K a l u i. A Hybrid Recommender System for e-Learning Based on Context Awareness and Sequential Pattern Mining. – Soft Computing, Vol. **22**, April 2018, No 8, pp. 2449-2461.
26. G o r a k a l a, S. K., M. U s u e l l i. Data Mining Techniques Used in Recommender Systems. – Building a Recommendation System with R, Birmingham, UK, Packt Publishing, 2015, pp. 9-15.
27. A m a t r i a i n, X., A. J a i m e s, N. O l i v e r, J. M. P u j o l. Data Mining Methods for Recommender Systems. – Recommender Systems Handbook, Boston, MA, USA, Springer, 2011, pp. 39-71.
28. C h e n, J., H. W a n g, Z. Y a n. Evolutionary Heterogeneous Clustering for Rating Prediction Based on User Collaborative Filtering. – Swarm and Evolutionary Computation, Vol. **38**, February 2018, pp. 35-41.
29. M a r g a r i s, D., C. V a s s i l a k i s, P. G e o r g i a d i s. Query Personalization Using Social Network Information and Collaborative Filtering Techniques. – Future Generation Computer Systems, Vol. **78**, January 2018, pp. 440-450.

30. X i a o, J., M. W a n g, B. J i a n g, J. L i. A Personalized Recommendation System with Combinational Algorithm for Online Learning. – Journal of Ambient Intelligence and Humanized Computing, Vol. **9**, Jun 2018, No 3, pp. 667-677.
31. I m r a n, H., M. B e l g h i s-Z a d e h, T. W. C h a n g, S. G r a f. PLORS: A Personalized Learning Object Recommender System. – Vietnam Journal of Computer Science, Vol. **3**, February 2016, No 1, pp. 3-13.
32. C h e n, W., Z. N i u, X. Z h a o, Y. L i. A Hybrid Recommendation Algorithm Adapted in e-Learning Environments. – World Wide Web, Vol. **17**, March 2014, No 2, pp. 271-284.
33. K l a š n j a-M i l i ć e v i ć, A., B. V e s i n, M. I v a n o v i ć, Z. B u d i m a c. E-Learning Personalization Based on Hybrid Recommendation Strategy and Learning Style Identification. – Computers & Education, Vol. **56**, April 2011, No 3, pp. 885-899.
34. H e r l o c k e r, J. L., J. A. K o n s t a n, J. R i e d l. Explaining Collaborative Filtering Recommendations. – In: Proc. of 2000 ACM Conference on Computer Supported Cooperative Work, ACM, 2000, pp. 241-250.
35. L u o, X., Y. X i a, Q. Z h u. Applying the Learning Rate Adaptation to the Matrix Factorization Based Collaborative Filtering. – Knowledge-Based Systems, Vol. **37**, January 2013, pp. 154-164.
36. W u, H., Z. Z h a n g, K. Y u e, B. Z h a n g, J. H e, L. S u n. Dual-Regularized Matrix Factorization with Deep Neural Networks for Recommender Systems. – Knowledge-Based Systems, Vol. **145**, April 2018, pp. 46-58.
37. Z h u, B., F. O r t e g a, J. B o b a d i l l a, A. G u t i é r r e z. Assigning Reliability Values to Recommendations Using Matrix Factorization. – Journal of Computational Science, Vol. **26**, May 2018, pp. 165-177.
38. Z h a n g, Y., M. C h e n, D. H u a n g, D. W u, Y. L i. iDoctor: Personalized and Professionalized Medical Recommendations Based on Hybrid Matrix Factorization. – Future Generation Computer Systems, Vol. **66**, January 2017, pp. 30-35.
39. Q i u, L., S. G a o, W. C h e n g, J. G u o. Aspect-Based Latent Factor Model by Integrating Ratings and Reviews for Recommender System. – Knowledge-Based Systems, Vol. **110**, October 2016, pp. 233-243.
40. G e m u l l a, R., E. N i j k a m p, P. J. H a a s, Y. S i s m a n i s. Large-Scale Matrix Factorization with Distributed Stochastic Gradient Descent. – In: Proc. of 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2011, pp. 69-77.
41. G o g n a, A., A. M a j u m d a r. Balancing Accuracy and Diversity in Recommendations Using Matrix Completion Framework. – Knowledge-Based Systems, Vol. **125**, Jun 2017, pp. 83-95.
42. T r a n, T., K. L e e, Y. L i a o, D. L e e. Regularizing Matrix Factorization with User and Item Embeddings for Recommendation. – In: Proc. of 27th ACM International Conference on Information and Knowledge Management, ACM, 2018, pp. 687-696.
43. N a j a f a b a d i, M. K., M. N. R. M a h r i n, S. C h u p r a t, H. M. S a r k a n. Improving the Accuracy of Collaborative Filtering Recommendations Using Clustering and Association Rules Mining on Implicit Data. – Computers in Human Behavior, Vol. **67**, February 2017, pp. 113-128.
44. N i l a s h i, M., O. I b r a h i, K. B a g h e r i f a r d. A Recommender System Based on Collaborative Filtering Using Ontology and Dimensionality Reduction Techniques. – Expert Systems with Applications, Vol. **92**, February 2018, pp. 507-520.
45. K i m, D., B. J. Y u m. Collaborative Filtering Based on Iterative Principal Component Analysis. – Expert Systems with Applications, Vol. **28**, May 2005, No 4, pp. 823-830.
46. B e n h a m d i, S., A. B a b o u r i, R. C h i k y. Personalized Recommender System for e-Learning Environment. – Education and Information Technologies, Vol. **22**, July 2017, No 4, pp. 1455-1477.
47. J J. MAE and RMSE – Which Metric is Better? Accessed 7 February, 2019. **http://medium.com/**
48. Y u, P., L. L i n, R. W a n g, J. W a n g, F. W a n g. A Unified Latent Factor Correction Scheme for Collaborative Filtering. – In: 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'14), IEEE, August 2014, pp. 581-586.
49. D a t t a, A., S. K o v a l e v a, P. M a r d z i e l, S. S e n. Latent Factor Interpretations for Collaborative Filtering. – arXiv preprint arXiv:1711.10816, 2017.

50. R o g e r s, M., W. Y a o, A. L u x t o n-R e i l l y, J. L e i n o n e n, D. L o t t r i d g e, P. D e n n y. Exploring Personalization of Gamification in an Introductory Programming Course. – In: Proc. of 52nd ACM Technical Symposium on Computer Science Education, March 2021, pp. 1121-1127.
51. D e b o i s, S. 10 Advantages and Disadvantages of Questionnaires. 8 March 2019. **https://surveyanyplace.com/blog/questionnaire-pros-and-cons/**
52. L e f e v e r, S., M. D a l, Á. M a t t h í a s d ó t t i r. Online Data Collection in Academic Research: Advantages and Limitations. – British Journal of Educational Technology, Vol. **38**, 2007, No 4, pp. 574-582.
53. B e i s k e, B. Research Methods: Uses and Limitations of Questionnaires, Interviews, and Case Studies. – GRIN Verlag, 2007, pp. 1-11.

Received: 15.03.2021; Second Version: 11.06.2021; Third Version: 12.08.2021; Accepted: 22.10.2021