On Recommendation of Learning Objects using Felder-Silverman Learning Style Model

Shaimaa M. Nafea, François Siewe, Ying He

Abstract—The e-learning recommender system in learning institutions is increasingly becoming the preferred mode of delivery, as it enables learning anytime, anywhere. However, delivering personalised course learning objects based on learner preferences is still a challenge. Current mainstream recommendation algorithms, such as the Collaborative Filtering (CF) and Content-Based Filtering (CBF), deal with only two types of entities, namely users and items with their ratings. However, these methods do not pay attention to student preferences, such as learning styles, which are especially important for the accuracy of course learning objects prediction or recommendation. Moreover, several recommendation techniques experience cold-start and rating sparsity problems. To address the challenge of improving the quality of recommender systems, in this paper a novel recommender algorithm for machine learning is proposed, which combines students actual rating with their learning styles to recommend Top-N course learning objects (LOs). Various recommendation techniques are considered in an experimental study investigating the best technique to use in predicting student ratings for e-learning recommender systems. We use the Felder-Silverman Learning Styles Model (FSLSM) to represent both the student learning styles and the learning object profiles. The predicted rating has been compared with the actual student rating. This approach has been experimented on 80 students for an online course created in the MOODLE Learning Management System, while the evaluation of the experiments has been performed with the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The results of the experiment verify that the proposed approach provides a higher prediction rating and significantly increases the accuracy of the recommendation.

Index Terms—Recommendation system, collaborative filtering, content-based filtering, hybrid filtering, e-learning, rating prediction, felder-silverman learning style model.

1 INTRODUCTION

E-learning Recommender Systems (E-LRS) have become popular in recent years. Compared with Learning Management Systems (LMS), which offer limited adaptivity and personalization, adaptive educational systems use intelligence algorithms to adapt to students learning style, enhance learning performance, accelerate goal achievement, reduce navigational overheads, and to enhance overall student satisfaction [1]. In addition, the adaptability and diversity of recommendations are desirable in e-learning recommender systems, because learners preferences and abilities keep changing. The diverse and adaptive Learning Objects (LOs) recommendations should be presented to motivate the learning potential of learners and to ensure a long-term learning experience [2], [3], [4].

A lot of learners are moving away from utilising e-learning systems, because they do not find them beneficial [26], [27], [28], [29]. On particular, this is owing to the fact that this type of learning environment cannot interact with learners as well as the traditional face to face one. Moreover, learners usually make a set of choices during learning, for instance, ‘how to learn’, ‘with whom to learn’ and ‘which learning pathway to follow’, and so on. To achieve this goal, it is essential to consider the students learning styles and use them in the design and implementation of e-learning environments, with the aim of making them more realistic and thus, attractive [30]. In this paper, we proposed a novel algorithm to recommend the most suitable course LOs taking into consideration student learning styles and LO profile. This paper is an extension of work originally presented in [70].

A recommender system (RS) enables users to cope with information overload by providing the most appropriate items based on their requirements. Figure 1 shows the traditional method of a two-dimensional recommender system, which has three main components: user, item, and rating. Rating, in this case, refers to the feedback that a user gives for a specific item, being implicit or explicit.

- **Explicit ratings** are when the user rates an item to express his/her level of interest. Rating can be in the form of a numeric value on a multi-point scale, e.g. 1 to 5 [83].
- **Implicit ratings** are generated by the RS itself, through inferences from users’ behaviour.

A user-item matrix is shown in Fig.1 where the elements in the matrix are the users ratings. In the matrix, the rows depict the user list, while the columns represent the items list. The numerical values from 1 to 5 in the matrix reflect the level of preference for a particular user for each item. The objective of RS algorithm in this setting is to predict the missing values in the matrix where users have not provided their preferences for certain items. However, this RS always suffers from data sparsity and cold start.

- Shaimaa M. Nafea with the Department of management information system, Arab Academy For Science Technology and Maritime, Egypt.
- François Siewe and Ying He with the Department of Computer Science and Informatics, De Montfort University, Leicester, UK.
- E-mail: P15017421@myemail.dmu.ac.uk, fsiewe@dmu.ac.uk, ying.he@dmu.ac.uk

Manuscript received April 19, 2005; revised August 26, 2015.
Data sparsity refers to the situation where the amount of information (ratings) of a target user is not sufficient enough to generate reliable related users (i.e., the number of commonly rated items among users is very small). While Cold start refers to the situation where an RS encounters new users or items with no ratings [100].

RSs have been researched and deployed extensively over the last decade in various application areas, including e-commerce [85], books [49], and movies [41], [48]. Another area where this support is very much demanded, is in the e-Learning field, where it is desirable for learners to be offered the most appropriate activities and learning objects to achieve their individual learning goals, whilst supporting their needs in the most efficient way. These kinds of RSs are usually focused on alleviating the information overload (of LOs) by filtering the most relevant content (LOs) to match the students’ preferences [2], [3], [4]. In sum, the major task of an RS is to construct a suitable model to calculate students’ interests. RSs are widely classified into three main techniques in the literature, as shown in figure [2].

From the above, the following are the major challenges in e-Learning recommendation systems:

1) First, the majority of the traditional recommendation algorithms have been developed for e-commerce applications that are unable to cover all the requirements of learning environments. In particular, they do not consider the learning process in their recommendation approach.

2) Second, the recommendation mechanisms that rely exclusively on two dimensions (i.e., users and items) don’t consider the attributes of learners and learning materials [16]. As a result, rich and vital information, such as learners’ learning styles and the properties of learning objects are overlooked.

3) Third, during the continuous learning process, learners do not actively make ratings or give comments, because they aim to achieve their goals within scheduled but limited learning time. As a result, learners’ learning profiles often seem isolated from each other. The extreme data sparsity caused by these factors can render traditional recommendation techniques ineffective.

4) Finally, traditional recommendation methods have a low ability to capture and perceive the changes in learners’ preferences in an adaptive way [72].

In order to avoid these drawbacks and improve the accuracy of Top-N course learning objects recommendation, our key contributions follow below:

1) In comparison with most of existing e-learning recommendation systems such as [39], [86], [89]
### TABLE 1: Summary of existing personalised e-learning recommendation systems

<table>
<thead>
<tr>
<th>Study</th>
<th>Description</th>
<th>Cold-Start</th>
<th>Issue</th>
<th>CF</th>
<th>CBF</th>
<th>HF</th>
</tr>
</thead>
<tbody>
<tr>
<td>[89]</td>
<td>Predict the most suitable learning materials to each learner based on collaboration with other learners.</td>
<td>✓</td>
<td>No past preferences</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>[39]</td>
<td>Recommend the most suitable learning materials for each learner based on the rating similarity with other learners. Finding learning objects that would be suitable for learners preferences (knowledge level and learning style).</td>
<td>✓</td>
<td>No past preferences or No high rated LO in past preferences</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>[37]</td>
<td>Recommend course learning objects based on neighborhood rating.</td>
<td>✓</td>
<td>No high rated LO in past preferences</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>[36]</td>
<td>Constructed a course ontology and retrieved the course according to learners learning styles.</td>
<td>✓</td>
<td>No past preferences</td>
<td>✓</td>
<td>(+ontology)</td>
<td></td>
</tr>
<tr>
<td>[91]</td>
<td>Proposed a hybrid recommender system for learning materials by combining CBF, CF and ontology.</td>
<td>✓</td>
<td>No past preferences or No high rated LO in past preferences</td>
<td>✓</td>
<td>✓</td>
<td>(+ontology)</td>
</tr>
<tr>
<td>[46]</td>
<td>Proposed a hybrid recommender system to recommend learning items in users learning processes.</td>
<td>✓</td>
<td>No past preferences or No high rated LO in past preferences</td>
<td>✓</td>
<td>✓</td>
<td>(+ontology)</td>
</tr>
<tr>
<td>[73]</td>
<td>Proposed a recommender system for storing and sharing research papers and glossary terms among university students and industry practitioners.</td>
<td>✓</td>
<td>No past preferences</td>
<td>✓</td>
<td>(+ontology)</td>
<td></td>
</tr>
<tr>
<td>[92]</td>
<td>Proposed the courseware management architecture with courseware recommendation that combines the user contents filtering and collaborative filtering.</td>
<td>✓</td>
<td>No past preferences</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>[75]</td>
<td>Proposed course recommendation system based on ontology and context aware e-learning.</td>
<td>✓</td>
<td>No high rated LO in past preferences</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>[76]</td>
<td>Recommend learning contents to users based on similarity between user profiles. Proposed framework for recommending learning materials based on the similarity of content items and good learners average rating strategy.</td>
<td>✓</td>
<td>No past preferences</td>
<td>✓</td>
<td>(+ontology)</td>
<td></td>
</tr>
<tr>
<td>[77]</td>
<td>Proposed framework for recommending learning resources based on the learners recent navigation history and by comparing similarities and differences among different learners preferences and instructional content available in the e-learning system.</td>
<td>✓</td>
<td>No past preferences or No high rated LO in past preferences</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>[80]</td>
<td>Recommend learning Materials based on multidimensional attributes</td>
<td>✓</td>
<td>No high rated LO in past preferences</td>
<td>✓</td>
<td>✓</td>
<td>(+association pattern analysis)</td>
</tr>
<tr>
<td>[81]</td>
<td>Proposed approach for selecting and sequencing the most appropriate learning objects.</td>
<td>✓</td>
<td>No high rated LO in past preferences</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

2) A new approach is proposed to overcome the new-learner zero-rated profile recommendation issue by determining the nearest learners with a similar historical rating and similar learning styles profile (Section [3.1.3]).

3) Several recommendations algorithms will be tested in order to find out which is the one that works better for the course learning objects recommendation. The accuracy of the recommendations is measured using traditional evaluation metrics, namely the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE). The results indicate that the hybrid recommender technique has higher accuracy in comparison with collaborative and content based recommendation techniques (Section [4]).

Our proposed algorithm has been implemented in C++ using Visual Studio and Windows Presentation Foundation (WPF) to design the Graphical User Interface (GUI). It has been evaluated using a real student dataset from AASTS MOODLE (Arab Academy for Science and Technology and Maritime Transport - Modular Object-Oriented Dynamic Learning Environment).

The remainder of the paper is organised as follows. The next section defines the main concepts used in the proposed approach. Section [5] discusses the proposed recommender algorithm, whilst Section [6] presents the experimental results.
Section 4 concludes the paper and future research directions are proposed.

2 Preliminaries

In the following sub-sections, we present techniques commonly used in recommender systems. These will be analysed later to improve the accuracy of recommendations. An overview of the similarity metrics and K-means clustering algorithm is also given.

2.1 Recommendation techniques

The underlying techniques used in recommender systems can be categorised into two broad classes: (a) content-based recommendation (b) collaborative filtering recommendation. New hybrid recommendation algorithms can be generated by synthesising these two methods [21].

2.1.1 Content-Based Filtering (CBF)

As a traditional recommendation method, the rationale for CBF is simple. The items recommended by this method are similar to the items of users interest [87], with matching information between items and users being the key procedure. In e-learning recommender systems, the items are the learning objects in the e-learning systems and the users are the learners. CBF recommender systems work with profiles of learners that are created at the beginning. A profile has information about a learner and his/her preferences, which are based on how he/she rates the LOs. Generally, when creating a profile, recommender systems make a survey to get initial information about a user in order to avoid the new-user problem. In the recommendation process, the engine compares the LOs that have already been positively rated by learner with LOs he/she has not done so and looks for similarities. Those LOs that are mostly similar to the positively rated ones, will be recommended to the user. In this case, the profiles of other users are not essential and they do not influence the recommendations of the user, for they are based on individual information. Figure 3 presents an illustrative example of CBF. From the figure, we can see the recommendation progress according to three main steps: Item Representation, Profile Learning and Recommendation Generation. As an example for e-learning application, [80] used learners recent navigation histories and similarities and dissimilarities among the contents of the learning materials for online automatic recommendations. Clustering was proposed by [88] to group learning documents based on their topics and similarities. Since in the e-learning environment learning materials are in a variety of multimedia formats, including text, hypertext, image, video, audio and slides, it is difficult to calculate the content similarity of two items [91]. In fact, the existing metrics in CBF only detect similarity between LOs that share the same attributes. This causes overspecialised recommendations that only include LOs very similar to those that the learner already knows.

2.1.2 Collaborative Filtering (CF)

Collaborative filtering became one of the most researched techniques of recommender systems after it was proposed and described by [60]. CF [17] recommends to the target learner learning resources that other similar learners have registered as liking previously. In other words, an important step in CF is to identify those learners most similar to the target learner. The similarity in taste of two learners is calculated based on their rating history. If two learners have the same or almost the same rated LOs in common, then they are deemed to have similar tastes. Such learners and others of the same ilk comprise a group or a so-called neighbourhood. A learner gets recommendations to choose LOs that he/she has not rated before, but have already been positively rated by those in his/her neighbourhood, as shown in Figure 4. To this end, several research efforts have been made to identify similarity measures so as to identify these users with common profiles [84] [85]. CF was used by [89] for prediction of the most suitable materials for the learner as follows. First, the weight between all users and the active learner is calculated by Pearson correlation. Then, the n users that have the highest similarity to the active learner are selected as belonging to the neighborhood. Finally, using the weight combination obtained from the neighbourhood, the rating prediction is calculated. Regardless of its success in many application domains, collaborative filtering has
two serious drawbacks. First, its applicability and quality are limited by the so-called sparsity problem, which occurs when the available data are insufficient for identifying similar users. Second, it requires knowing many user profiles in order to elaborate accurate recommendations for a given user. Given in some e-learning environments the learner population is low, recommendation results with this method in such cases will have poor accuracy.

2.1.3 Hybrid Filtering (HF)

In the last few years, researchers of recommender systems have explored hybridisation of recommendation techniques as an approach for developing effective recommender systems. Table 1 lists some of the techniques that have been used to this end. Hybrid filtering entails combining two or more recommendation techniques to improve performance, as shown in Figure 5. In [92], a combination of content-based and collaborative filtering was implemented to make personalised recommendations for a courseware selection module. The algorithm starts with user u entering some keywords on the portal of the courseware management system. Then, the courseware recommendation module finds within the same user interest group of user u the k courseware with the same or similar keywords that others have chosen. [23] applied association rule mining to identify interesting information through students usage data in the form of IF-THEN recommendation rules and then, used a collaborative recommender system to share and score the recommendation rules obtained by teachers with similar profiles as well as other experts in education.

2.2 Similarity metrics

Similarity metrics are the backbone of CF and CBF helping to predict the ratings of unrated items. Regarding which, in this study, the two most convenient similarity metrics, namely, the Pearsons correlation and cosine similarity [70] are considered.

2.2.1 Pearson Correlation

The Pearson correlation coefficient is a measure of the linear dependence between two variables (real-valued vectors). Specifically, that of two variables x and y is formally defined as the covariance of the two variables divided by the product of their standard deviations (which acts as a normalisation factor) [78] and it can be equivalently defined by Eq. 1.

\[ P(x, y) = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}} \]  

where \(\bar{x}\) and \(\bar{y}\) are the mean values of x and y, respectively.

The coefficient \(P(x, y)\) ranges from -1 to 1 and is invariant to linear transformations of either variables. The value -1 represents perfect negative linear dependence, 0 no linear dependence, and 1 perfect positive linear dependence. Used as a similarity metric, negative values indicate dissimilarity, while positive values measure the similarity between the two variables with 1 be the perfect similarity.

2.2.2 Cosine Similarity

The cosine similarity involves measuring the angle between two vectors [25] and is calculated by Eq. 2, as the ratio of the scalar product by the product of the magnitudes.

\[ c(x, y) = \frac{x \cdot y}{||x|| \cdot ||y||} \]  

The values of \(c(x, y)\) range from -1 to 1 in general, and from 0 to 1 if the coordinates of x and y are non-negative values. This paper is interested in the latter where the value 0 represents no similarity and 1 perfect similarity.

2.3 K-means clustering algorithm

Clustering is one of the most common data mining techniques used in recommendation systems in order to develop recommendation rules or build recommendation models from large data sets [79]. It can be defined as the process of organising objects in a database into clusters (or groups), such that objects within the same cluster have a high degree of similarity, while those belonging to different ones have a high degree of dissimilarity. The K-means algorithm [71] is one of the most popular clustering algorithms due to its simplicity and intuitive interpretation. The algorithm has the following steps.

Step 1: Select K random points from the dataset as initial cluster centroids.
Step 2: Create K clusters by associating each data point with its closest cluster centroid, according to the Euclidean distance defined by Eq. 3.

\[ D(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \]  

Step 3: Recalculate the centroid of each cluster as the mean of all the data points in that cluster.
Step 4: Repeat steps 2 and 3 until the centroids no longer change.

In our proposed system, we apply data clustering by employing the K-means algorithm to improve computational efficiency in terms of accuracy and quality of recommendations. To begin with, k initial cluster centres are identified and then the given data set is iteratively refined.
2.4 Description of the Felder-Silverman Learning Style Model

The term 'learning styles' refers to the preferential way in which the student perceives, processes, understands and retains information [50]. Various learning style models have been presented in the past by researchers, such as those of Felder and Silverman [19], Honey and Mumford [51], Kolb citekolb1984englewood, Mayer and Myers [62], Dunn [61] and Pask [53]. In our case, we use the Felder and Silverman model (FSLSM) [19] to represent both the student learning styles and the learning object profiles for the following reasons.

- First, it is the most widely used in educational systems thanks to its ability to quantify students learning styles, as shown in [2].
- Second, it is used very often in technology enhanced learning and some researchers even argue that it is the most appropriate learning style model for the use in adaptive learning systems such as [20], [21], [22], [23], as well as being easy to implement [33], [63].

FSLSM describes learning styles by characterising each learner according to four dimensions, each of which, is defined as below.

The information processing dimension (active/reflective) tells how one prefers to process information. An active learner wants to try things out, working with others in groups, whilst a reflective one chooses to think things through, working alone or with a familiar partner.

The information input dimension (visual/verbal) determines how ones prefers information to be presented. A visual learner likes visual presentations, pictures, diagrams, and flow charts. A verbal learner prefers written and spoken explanations.

The information understanding dimension (sequential/global) determines how ones prefers to organise and progress towards understanding information. A sequential learner prefers linear thinking and learning in small incremental steps. By contrast, a global learner prefers holistic thinking, systems thinking, and learns in large leaps.

The information perception dimension (sensing/intuitive) states how you prefer to perceive or take in information. A sensing learner is attracted to concrete thinking, is practical as well as being concerned about facts and procedures. While an intuitive learner opts for conceptual thinking, being innovative, as well as being interested in theories and meanings.

It should be noted that each of these dimensions is characterised by a pair X/Y of learning style attributes (i.e. active/reflective, sequential/global, visual/verbal, and sensing/intuitive) meaning that the learning style of a learner in a particular dimension ranges from perfect X to perfect Y. For example, in the information processing dimension, the learning style of a student can be 70% active and 30% reflective. Of course, the percentage of X and the percentage of Y must sum up to 100%. Felder and Silverman [19] developed an Index of Learning Styles (ILS) questionnaire, which comprises 44 questions that has been shown to be effective in identifying the learning style of each individual learner. ILS provides a method of calculating the percentage values of learning style attributes from the learners answers to the questionnaire [19], [69].

The next section presents a novel algorithm for recommending learning objects based on student learning style.

3 Proposed Algorithm for Recommending Personalised Learning Objects

In this work, a new algorithm for rating prediction of the learning objects is proposed. The proposed algorithm predicts the ratings for a given LO, for a given student based on learning styles, and their rating. The FSLSM learning style model described above is adopted to represent both the student learning preferences and the learning object profiles. The rating is given on a scale of 1-5. First, work is carried out to design an effective algorithm for recommending Top-N personalised learning objects in e-Learning systems based on student learning styles, as presented in (Sect. 3.1). Then an experimental study undertaken to find out which algorithm produces the best accuracy for rating prediction. The best performing algorithm is then retained for the recommender system. To present this study clearly, we first provide two definitions.

Definition 1 (Student Profile). It is assumed that the student learning style is represented by a vector of real values ranging from 0 to 1 (or from 0% to 100%) as follows, where the prefixes of learning style attributes are used as placeholder.

\[ LS = (act, ref, vis, ver, seq, glo, sen, int) \]  (4)

Some examples of student learning style vectors are given in Table 3 and these can be calculated using the learners responses to the ILS questionnaire [69] or according to his/her learning behaviour [70].

Definition 2 (Object Profile). The learning content materials are structured into learning objects for each topic. Learning objects are provided in various formats and media in order to meet the learning styles of individual learners. They can be text documents (e.g. pdf), presentations (e.g. powerpoint slides), images, audios, videos, simulations, etc. For example, a visual learner will prefer to watch a video than to read a pdf document, while a verbal one will choose to do opposite. Hence, a learning object profile (OP) can be represented by a FSLSM learning style vector indicating the category of learners that this learning object is suitable for, as in Eq. 5.

\[ OP = (act, ref, vis, ver, seq, glo, sen, int) \]  (5)

Unlike the student learning styles that are calculated through the ILS questionnaire or behaviour, it is assumed that the learning object profile is set by the teacher or an education professional. Some examples of learning object profiles are given in Table 3 for illustration.

In the remaining part of the subsection, we give a description of the proposed predicated rating algorithms in detail and analyse the accuracy of the recommendations. Finally, we present the prediction algorithm proposed in our work.
TABLE 2: Summary of existing personalised e-learning systems considering learning styles

<table>
<thead>
<tr>
<th>System Name</th>
<th>LS Model</th>
<th>Adaptation technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protus 2.0</td>
<td>FSLSM</td>
<td>Learning style identification &amp; personalised content recommendation</td>
</tr>
<tr>
<td>WELSA [54]</td>
<td>Unified LS Model</td>
<td>Presentation recommendation &amp; sequencing</td>
</tr>
<tr>
<td>MAS-PLANG</td>
<td>FSLSM</td>
<td>Content personalisation</td>
</tr>
<tr>
<td>UZWEBSMART</td>
<td>VARK</td>
<td>Content recommendation</td>
</tr>
<tr>
<td>FLORS [56]</td>
<td>FSLSM</td>
<td>Learning objects recommendation</td>
</tr>
<tr>
<td>Tortella and Graf [94]</td>
<td>FSLSM</td>
<td>Video, audio, presentation score calculation in mobile environment</td>
</tr>
<tr>
<td>Christudas [97]</td>
<td>FSLSM</td>
<td>Compatibility and complexity of learning objects</td>
</tr>
<tr>
<td>e-Teacher [98]</td>
<td>FSLSM</td>
<td>Course personalisation</td>
</tr>
<tr>
<td>OSCAR CITS</td>
<td>FSLSM</td>
<td>Course personalisation</td>
</tr>
<tr>
<td>TANGOW [57]</td>
<td>FSLSM</td>
<td>Adapt course structure &amp; sequencing</td>
</tr>
<tr>
<td>iWeaver[18]</td>
<td>Dunn &amp; Dunn Model</td>
<td>Media recommendation (Flash animations or streaming audio)</td>
</tr>
<tr>
<td>learnFit [58]</td>
<td>MBTI</td>
<td>Course personalisation</td>
</tr>
<tr>
<td>CS388 [58]</td>
<td>FSLSM</td>
<td>Course personalisation</td>
</tr>
<tr>
<td>Kurilovas [93]</td>
<td>Honey &amp; Mumford learning style</td>
<td>Teaching/learning strategies</td>
</tr>
</tbody>
</table>

TABLE 3: Examples of student learning style vectors

<table>
<thead>
<tr>
<th></th>
<th>act</th>
<th>ref</th>
<th>vis</th>
<th>ver</th>
<th>seq</th>
<th>glo</th>
<th>sen</th>
<th>int</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fatima</td>
<td>0.7</td>
<td>0.3</td>
<td>0.2</td>
<td>0.8</td>
<td>0.5</td>
<td>0.5</td>
<td>0.6</td>
<td>0.4</td>
</tr>
<tr>
<td>Tom</td>
<td>0.4</td>
<td>0.6</td>
<td>0.1</td>
<td>0.9</td>
<td>0.7</td>
<td>0.3</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>Clara</td>
<td>0.5</td>
<td>0.5</td>
<td>0.6</td>
<td>0.4</td>
<td>0.8</td>
<td>0.2</td>
<td>0.7</td>
<td>0.3</td>
</tr>
</tbody>
</table>

TABLE 4: Examples of learning object profile

<table>
<thead>
<tr>
<th></th>
<th>act</th>
<th>ref</th>
<th>vis</th>
<th>ver</th>
<th>seq</th>
<th>glo</th>
<th>sen</th>
<th>int</th>
</tr>
</thead>
<tbody>
<tr>
<td>OP_1</td>
<td>0.2</td>
<td>0.3</td>
<td>0.5</td>
<td>0.5</td>
<td>0.2</td>
<td>0.1</td>
<td>0.9</td>
<td>0.4</td>
</tr>
<tr>
<td>OP_2</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.3</td>
<td>0.7</td>
</tr>
<tr>
<td>OP_3</td>
<td>0.9</td>
<td>0.1</td>
<td>0.7</td>
<td>0.3</td>
<td>0.7</td>
<td>0.7</td>
<td>0.1</td>
<td>0.9</td>
</tr>
</tbody>
</table>

3.1 Rating prediction algorithms

In this step, the algorithm predicts the rating for a given learner for a given set of learning objects using prediction approaches. The following sections will briefly explain these approaches.

3.1.1 Predicting ratings based on Collaborative Filtering

From the description of the previous section, we notice that the traditional CF methods heavily rely on the co-rated items. However, the similarity computation cannot be performed when there are no rated items, which is called cold start problem (see Sect[1]). For improving the accuracy and quality of recommendation, our research CF is implemented as follows:

Let LS be the learning style vector of the active student.

1) Apply K-means to cluster the students profiles.
2) Select cs the nearest SP cluster to LS as in Eq. [1]
3) Foreach LO x
   a) Let I = set of the top-n nearest elements to LS in cs that have rated x as in Eq. [2]
   b) If |I| > 0 then calculate the predicted rating for x as in Eq. [6]
   c) If |I| = 0 then calculate the predicted rating for x as in Eq. [7]
4) Recommend the top-n highly rated LOs.

\[ \hat{r}_1(LS, x) = \frac{\sum_{u \in I} sim(LS, u) \times r(u, x)}{\sum_{u \in I} sim(LS, u)} \]  

The solution of cold-start (case |I| = 0 ) proposed by measuring the similarity between the context of learning objects LO and student profile to predict and fill in missing user ratings and subsequently improve the accuracy of collaborative filtering as Eq. [7].

3.1.2 Predicting ratings based on Content-based filtering

As explained in section 2.1, the general principle of content-based approaches is to identify the common characteristics of learning objects that have received a favorable rating from a learner, and then recommend to him/her new LO that share these characteristics. In this work, we proposed an algorithm of a similarity model to enhance the accuracy or recommendation, the similarity between SP and LO is calculated using the following steps:

Let LS be the learning style vector of the active student.

1) Let O be the set of all learning objects rated by LS.
2) If O ≠ ∅ then
   a) Apply K-means to cluster O
   b) Foreach LO x
      i) Let co = the nearest LO cluster to x as in Eq. [1]
      ii) Let J = set of the top-n nearest elements to x in co as in Eq. [2]
      iii) Calculate the predicted rating for x as in Eq. [6]
   c) Recommend the top-n highly rated LOs.
3) If O = ∅ then
   a) Apply K-means to cluster all the learning objects
   b) Let co = the nearest LO cluster to LS
   c) Foreach x ∈ co
      i) Calculate the predicted rating for x as in Eq. [7]
d) Recommend the top-n highly rated LOs in $co$.

$$\hat{r}_3(LS, x) = \frac{\sum_{u \in J} \sim(x, u) \times r(LS, u)}{\sum_{u \in J} \sim(x, u)}$$  \hspace{1cm} (8)

3.1.3 Predicting ratings based on Hybrid Filtering

For improving the accuracy and quality of recommendation, our research HF is implemented as follows: Let $LS$ be the learning style vector of the active student.

1) Let $\alpha$ be the weight of CF in the hybrid model; $0 \leq \alpha \leq 1$.
2) Apply K-means to cluster the students profiles
3) Select $cs$ the nearest SP cluster to $LS$
4) Let $O$ be the set of all learning objects rated by $LS$.
5) Apply K-means to cluster $O$.
6) Foreach LO $x$
   a) Let $I$ = set of the top-n nearest elements to $LS$ in cs that have rated $x$.
   b) Let $co_x$ be the nearest LO cluster to $x$.
   c) Let $J$ = set of the top-n nearest elements to $x$ in $co_x$.
   d) If $\|I\| > 0$ and $\|J\| > 0$ then calculate the predicted rating for $x$ as in Eq. (9).
   e) If $\|I\| = 0$ and $\|J\| > 0$ then calculate the predicted rating for $x$ as in Eq. (8).
   f) If $\|I\| > 0$ and $\|J\| = 0$ then calculate the predicted rating for $x$ as in Eq. (7).
   g) If $\|I\| = 0$ and $\|J\| = 0$ then calculate the predicted rating for $x$ as in Eq. (8).
7) Recommend the top-n highly rated LOs.

$$\hat{r}(LS, x) = \alpha \times \hat{r}_1(LS, x) + (1 - \alpha) \times \hat{r}_3(LS, x)$$  \hspace{1cm} (9)

Note the in Eq. (7), the value of $\alpha$ is between 0 and 1. Here are some examples:

- $\hat{r}(LS, x) = 0.5 \times \hat{r}_1(LS, x) + (1 - 0.5) \times \hat{r}_3(LS, x) = 0.5 \times \hat{r}_1(LS, x) + 0.5 \times \hat{r}_3(LS, x)$
- $\hat{r}(LS, x) = 0.2 \times \hat{r}_1(LS, x) + (1 - 0.2) \times \hat{r}_3(LS, x) = 0.2 \times \hat{r}_1(LS, x) + 0.8 \times \hat{r}_3(LS, x)$
- $\hat{r}(LS, x) = 0.8 \times \hat{r}_1(LS, x) + (1 - 0.8) \times \hat{r}_3(LS, x) = 0.8 \times \hat{r}_1(LS, x) + 0.2 \times \hat{r}_3(LS, x)$
- $\hat{r}(LS, x) = 0.75 \times \hat{r}_1(LS, x) + (1 - 0.75) \times \hat{r}_3(LS, x) = 0.75 \times \hat{r}_1(LS, x) + 0.25 \times \hat{r}_3(LS, x)$

“What is the recommendation algorithm that provides the best prediction of student ratings of learning objects?” An experimental study is carried out aimed at finding the answer to this question.

4 Experimental Study

The proposed recommendation system will suggest the most relevant learning objects to its learners from a large list. An experimental study was carried out to determine the most effective recommendation techniques to be used for the recommendation of LOs in e-learning systems. In this section, we describe the dataset [4.1] performance measurement [4.2] and the results [4.3] of our proposed approach with the existing approach.

4.1 Dataset

The dataset of the MOODLE log-file at AAST is used in this study for the autumn and spring semesters in 2016 and 2017 and 2017 and 2018 in the school of business. MOODLE (Modular Object-Oriented Dynamic Learning Environment) is defined as a course management system (CMS), being a free and open source software package designed using pedagogical principles, to help educators by creating effective online learning communities. The course of interest is on networks and e-commerce and comprises 20 topics, with each topic having multiple learning objects in various presentation styles. There was a minimum of 15 learning objects for each topic. The experimental set up consisted of 80 students whose learning styles were identified using the ILS questionnaire, as explained in [69]. During the course, the students were asked to rate each learning object using a 5-level Likert scale, with 1 being not at all useful and 5 be very useful to their learning. In order to evaluate the different aspects of the proposed method, student dataset was split into different parts, including:

1) Cold-start students: a set of students with lower than 5 ratings;
2) Cold-start learning objects: a set of new LOs;
3) All students.

Cold-start was utilised to assess the ability of the algorithms to predict the ratings for those students a few LOs, so little information was available for these users. The goal was to investigate how additional sources of information, such as learning styles of a student, can be used along with rating information to improve the accuracy of rating prediction.

4.2 Performance measurement

In this analysis, accuracy metrics are considered to assess the performance of the proposed recommender system algorithm. We use statistical accuracy metrics to evaluate the accuracy of the rating prediction algorithm. The frequently used statistical metrics are Mean Absolute Error (MAE) and Root-Mean-Square Error (RMSE). $r_i$ is the actual student rating of the learning object $i$ and $\hat{r}_i$ is the predicted student rating for that learning object, $1 \leq i \leq n$.

In the computation of MAE, the first absolute sum of the difference between the actual and predicted rating is calculated, and then, it is divided by the total number of predicted ratings. Hence, a smaller value of MAE indicates a better accuracy of prediction, as in Eq. (10).

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |r_i - \hat{r}_i|$$  \hspace{1cm} (10)

The Root Mean Square Error (RMSE) is calculated by dividing the sum of squares of the differences of the actual and predicted ratings by the total number of ratings on which
the predictions are made. The RMSE is obtained by taking the square root of the MSE, as in Eq. (11).

\[
RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (r_i - \tilde{r}_i)^2} \quad (11)
\]

4.3 Experiments and Evaluation

In our proposed work, a new approach to system for ratings prediction of the course learning objects is proposed. The prototype was implemented in C++ using Visual Studio and Windows Presentation Foundation (WPF) to design the GUI (graphical user interface), with the SQL server being utilised to allocate a system dataset and learner rating. A set of experiments was conducted on a Windows based PC with an Intel core i5 processor having a speed of 2.40 GHz and 16GB of RAM. The GUI, which allows for selecting various combinations of similarity metrics, is depicted in Fig. 6.

The first experiment was focused on the accuracy of rating prediction, whereas the second one focused on the cold-start. Finally, the last part of the evaluation is about the integration of recommendation algorithms into AAST-MOODLE for testing them on real students. Each experiment will be detailed and discussed in the following subsections.

4.3.1 Evaluation on Rating Prediction

After performing the preprocess on student dataset, 15 students were chosen randomly and the calculated predicted rating for these LO sets is shown in Table 3. The experimental results in the table show that the HF-0.5 algorithm has the best accuracy. From figure 7, it can be inferred that HF-0.5 has the least value of MAE and thus, provides better predictions. The MAE value of HF-0.5 is 0.9, whilst that of CBF is 1.52, which is greatest among all the three approaches. Hence, the latter method will produce the least accuracy in prediction. Theoretically and experimentally, it has already been proven that the root-mean-square error is always greater than the Mean Absolute Error.

Fig. 7 shows that the proposed HF-0.5 algorithm again delivers a smaller RMSE than other, which indicates that it is more accurate.

4.3.2 Evaluation of Cold-start

From another point of view, the experiments were repeated to evaluate the proposed approaches to handle cold-start problem recommendations for new students and new LOs.

- **New students**: The three different algorithms (CF, CBF and HF-0.5) can deal with new students by incorporating their personalised learning styles with their rating. Fig. 8a and Fig. 8b compare the accuracy of the different recommendation algorithms. The results achieved by the hybrid filtering approach are impressive. Given the above results, analysis, and discussion, it is concluded that the proposed algorithm HF-0.5 performs better than CF and CBF.

- **New learning objects**: Three algorithms can make recommendation for new LOs by measuring the similarity between learning object profile and student learning styles. From 9a and 9b we observe that HF-0.5 consistently outperforms in all the experiments, which indicates that our model handles new items better than CF and CBF.

4.3.3 Real Student Evaluation

The last part of the evaluation was to validate our method on real circumstances by integrating it in AAST MOODLE. The system was modified to be able to read student profiles and, subsequently, recommend the course LOs. To evaluate the student satisfaction with recommendations, a closed-ended questionnaire was administered to the 80 students who participated in the experiment. Previous studies on recommender systems have identified user satisfaction as one of the important evaluation measures [102], [103]. First, they were asked to fill in the FSLSM questionnaire [69] to create their profile, as shown in Fig. 10. After that, recommended five lessons for them over five weeks, with each including a set of LOs. We presented our empirical study to the Business Information Systems department at AAST. The questionnaire sought to find out whether the learner was satisfied or not satisfied with the LO recommendations. Fig. 11 illustrates the responses of the learners on satisfaction with these from each of the three recommendation algorithms.

From the figure it is evident that the majority (95%) of the students were satisfied with the LO recommendations from the HF-0.5 algorithm. On the other hand, just (60%) and (56%) of the students were satisfied with the recommendations from the CF and CBF recommendation algorithms, respectively.

5 Conclusion and Future Directions

Whilst recommender systems have been studied in the past decade, the study of rating prediction for recommender systems is a more recent phenomenon. In this paper, we have concentrated on improving the accuracy and quality of recommendation in case of cold start and sparsity data. To this end, an improved rating predication algorithm has been proposed. To test the performed to compare the hybrid filtering, collaborative filtering, and content based filtering algorithms. The results of the implementation suggest that the HF-0.5 algorithm predicts better ratings to LOs as the value of the Mean Absolute Error is less than other algorithms. In addition to addressing both new students and new LOs issues in this context, the proposed recommender algorithm in the present research seems to improve the accuracy of recommended items to new students. However, this work
has some limitations, which could be addressed in future work. First, the dataset in the current work was quite small and a larger one would add more weight to the findings. A second future direction is working on some other challenges of recommendation systems, such as scalability.

REFERENCES


![Fig. 6: ULearn interface](image)

**TABLE 5: Showing predicted rating using the proposed algorithm**

<table>
<thead>
<tr>
<th>S.ID</th>
<th>LO.ID</th>
<th>Act. R</th>
<th>CBF</th>
<th>HF-0.1</th>
<th>HF-0.2</th>
<th>HF-0.3</th>
<th>HF-0.4</th>
<th>HF-0.5</th>
<th>HF-0.6</th>
<th>HF-0.7</th>
<th>HF-0.8</th>
<th>HF-0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>25</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>13</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>15</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>22</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>8</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>7</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>16</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

Act. R = Actual Rating  
CBF = Content Based Filtering  
CF = Collaborative Filtering  
HF = Hybrid filtering  
S.ID = Student ID  
LO.ID = Learning Object ID
Fig. 7: Accuracy of the recommender algorithm using MAE and RMSE for m=50

(a) Performance comparison for cold students using MAE

(b) Performance comparison for cold students using RMSE

(a) Performance comparison for cold LOs using MAE

(b) Performance comparison for cold LOs using RMSE
Fig. 10: Questionnaire interface

Fig. 11: Student satisfaction with LO recommendations

rating. In 2017 IEEE Conference on e-Learning, e-Management and e-Services (IC3e) (pp. 61-66). IEEE.


