EAI Endorsed Transactions
on e-Learning
Research Article EAI.EU

A traditional-learning time predictive approach for e-learning systems in challenging environments

Kenmogne Edith Belise[1,*]

[1]Faculty of Science, Department of Mathematics and Computer Science, LIFA, Po. Box. 67 Dschang, Cameroon

Abstract

The explosion of world-wide-web has offered people a large number of online courses, e-classes and e-schools. Such e-learning applications contain a wide variety of learning materials which can confuse the choices of learner to select. Although the area of recommender systems has made a significant progress over the last several years to address this problem, the issue remained fairly unexplored for challenging environments. This paper proposes an approach to predict traditional-learning times for recommender systems in such environments.

Received on 12 September 2016; accepted on 6 July 2017; published on 29 November 2017

Keywords: challenging environment, context, e-learning, offline learning, online learning, traditional learning, prediction, recommender, content filtering, collaborative filtering

doi:10.4108/eai.29-11-2017.153391

1. Introduction

E-learning [3–5, 7, 11, 23, 26, 43, 46, 54] and distance education is growing worldwide. According to some research reports [21, 29], in Africa the demand for products and services for e-learning (platforms, authoring tools, content development, etc.) is growing rapidly, as well as Asia and Latin America.

African countries are embracing Information and Communications Technology (ICT) to increase access to Internet. The use of mobile devices and smart phones by a growing number of people could facilitate e-learning and m-learning. In Africa, these types of environments are being introduced into both urban and rural settings, especially in areas in which reliable power sources have not previously been readily available. In a similar way, it will also be necessary to expand the coverage of broadband internet and technology infrastructure, which partly depends on public policies. In many both urban and rural areas, the limited bandwidth of the few available telecommunication lines that are joining the Internet cause line congestion and make access exceedingly slow, often beyond the limit of usability. Therefore, the Quality of the Service (QoS) is not stable and depends on the geographical location. This is also true for the electricity network. Thereby, depending on the performances of both the Internet and the electricity network, we consider three types of learning, namely *online learning*, *offline learning* and *traditional learning* (without a computer). Furthermore, others factors such as the standard of living and the effects of computer use on eye health and vision motivate the choice of these three types of learning even in developed countries. It is well known that staring at a computer screen for too long can cause eyestrain, blurry vision, trouble focusing at a distance, dry eyes, headaches, neck, back, and shoulder pain.

E-learning [3, 7, 11, 23, 26, 54] applications contain the plentiful learning materials namely courses, lessons, references and exercises. As a result, learners face the problem in selecting learning materials which are suitable for their learning levels from the potentially overwhelming number of alternatives. Although the area of recommender systems has made a significant progress over the last several years to address this problem, the issue remained fairly unexplored for challenging environments. In such systems, the rating process of learning materials is based on learning times which in turn involve online learning times, offline learning times and traditional learning times.

★Please ensure that you use the most up to date class file, available from EAI at http://doc.eai.eu/publications/transactions/latex/

*Corresponding author. Email: ebkenmogne@gmail.com

Online learning times and offline learning times can be obtained automatically while traditional learning times should be obtained either manually or from a predictive algorithm. This last issue is the concern of this paper. In this work, we propose an approach to predict traditional-learning times for e-learning systems in challenging environments.

The rest of this paper is organized as follows. In section 2, the previous related works on e-learning material recommender systems are presented. Section 3 introduces a traditional-learning time predictive approach and describes the related paradigm, concepts and algorithms step by step. Section 4 provides concluding remarks along with suggestions for future work.

2. Related Works

2.1. What is context ?

Context [1, 18, 48] is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application. Entity is often a user, an item and the rating from a user over an item in terms of recommender systems.

There are three approaches for using the context in content filtering and collaborative filtering: contextual pre-filtering, contextual post-filtering, and contextual modeling. Three ways are used to obtain the parameters of the learning context of a given learner:

1. Explicitly from the learner by asking direct questions or eliciting through other means.
2. Implicitly from the data or the environment such as the change in location automatically detected by devices or the time stamp of a transaction.
3. Inferring the context using statistical or data mining methods.

2.2. Recommender systems

Basically, a Recommender System (RS) is defined as a decision making strategy for users under complex information environments [30, 38, 43]. Also, RS was defined from the perspective of e-commerce as a tool that helps users search through records of knowledge which is related to user interests and preferences [20, 56]. It handles the problem of information overload that users normally encounter by providing them with personalized, exclusive content and service recommendations.

RSs attempt to discover user's preferences [47] , and to learn about them in order to anticipate their needs. Broadly speaking, a RS provides specific suggestions about items (products or actions) within a given domain, which may be considered of interest to the given active user [53]. They are beneficial to both service providers and users [51]. They reduce transaction costs of finding and selecting items in an online shopping environment. RSs have also proved to improve decision making process and quality [49]. In e-commerce setting, RSs enhance revenues, for the fact that they are effective means of selling more products [51]. In scientific libraries, RSs support users by allowing them to move beyond catalog searches. In e-learning systems [8, 11, 18–20, 23, 37], RS is a significant part for personalization and recommendation of appropriate materials to the learner.

Recently, various approaches [1–3, 7, 8, 15, 18, 19, 28, 31, 34, 35, 38, 43, 54, 57] for building RSs have been developed, which can utilize either content-based filtering, collaborative filtering, sequential pattern mining or hybrid filtering.

Content filtering based recommender. Content-based recommendation systems [38] try to recommend items similar to those a given user has liked in the past. Indeed, the basic process performed by a content-based recommender consists in matching up the attributes of a user profile [47] in which preferences and interests are stored, with the attributes of a content object (item), in order to recommend to the user new interesting items. Following that technique in e-learning systems, the conclusion of the utility of a material for a learner is made on the basis of the utility assigned by the learner to the materials similar to the given one.

A variety of content-based recommender systems exist. LIBRA [41] implements a naÄśve Bayes text categorization method for book recommendation that exploits the product descriptions obtained from the Web pages of the Amazon on-line digital store. Re:Agent [10] is an intelligent email agent that can learn actions such as filtering, downloading to palmtops, forwarding email to voicemail, etc. using automatic feature extraction. Re:Agent users are required only to place example messages in folders corresponding to the desired actions. Re:Agent learns the concepts and decision policies from these folders. Citeseer [9, 22] assists the user in the process of performing a scientific literature search, by using word information and analyzing common citations in the papers. INTIMATE [39] recommends movies by using text categorization techniques to learn from movie synopses obtained from the Internet movie database. In order to get recommendations, the user is asked to rate a minimum number of movies into six categories: terrible, bad, below average, above average, good and excellent. In the same way, Movies2GO [42] learns user preferences from the synopsis of movies rated by the user. The innovative aspect of the system is to integrate voting schemes, designed to allow multiple individuals with conflicting preferences arrive at an acceptable compromise, and adapt them to manage conflicting preferences in a single user.

Despite the success of the content filtering approach, several limitations have been identified. The main limitation is that only materials with a high degree of agreement with the learner's preferences will be recommended. Others limitations associated with content-based filtering techniques are limited content analysis, cold-start problem, overspecialization and sparsity of data [2, 6, 17, 38, 43]. The limited content analysis occurs when content-based techniques have a limit in the number of features that are associated, resulting in unsuitable suggestion. The overspecialisation problem occurs when a user is only recommended items that are similar to items that were rated or bought before. The cold-start problem occurs when dealing with new users and new or updated items in web environments. Since the user has not rated or purchased items or the items has not been rated yet, it is difficult to find a group of similar users or items. In other word, the RS cannot draw any relation between users and items because of the absence of information about the user or item present. The sparsity problem occurs when the number of user ratings from the user-item rating dataset are insufficient for identifying similar users. This arises from an insufficient quantity of coitem ratings for similarity calculations between users or items. As a consequence, it is not possible to make reliable recommendations due to an initial lack of ratings. In other word, the lack of rating history causes the data sparsity. All these problems limit the quality of recommendations and the applicability of content filtering approach in general.

Collaborative filtering based recommender. Collaborative filtering technique [7, 8, 26, 28, 32, 37, 54] is the most mature and the most commonly implemented. Collaborative filtering recommends items by identifying other users with similar taste; it uses their opinion to recommend items to the active user. Following that technique in e-learning systems, the utility of a material for a learner is evaluated based on utilities assigned to the material by learners who resemble the given learner. Thus, if the recommender is guided only by learner preferences, it cannot recommend a material until it gets a sufficient number of evaluations.

Collaborative recommender systems have been implemented in different application areas. GroupLens [2] is a news-based architecture which employs collaborative methods in assisting users to locate articles from massive news database. Ringo [14] is an online social information filtering system that uses collaborative filtering to build users profile based on their ratings on music albums. Amazon [61] uses topic diversification algorithms to improve its recommendation. The system uses collaborative filtering method to overcome scalability issue by generating a table of similar items offline through the use of item-to-item matrix. The system then recommends other products which are similar online according to the purchase history of the user.

Despite the success of this filtering technique, it suffers from some limitations [2, 6, 17, 43] such as cold-start, sparsity and scalability problems. These problems usually reduce the quality of recommendations and they could be solved using hybrid methods [15, 17]. The scalability problem occurs when the number of users or items grows. The algorithms cannot respond immediately to online requirements or make recommendations for all users.

Sequential pattern based recommender. Sequential pattern mining [25, 27, 50, 52, 60] aims at analysing ordered or timed data to extract interesting patterns. Broadly, a pattern is considered interesting if it occurs frequently in the data, i.e. the number of its occurrences is greater than a fixed given threshold. Based on the intuition that frequent patterns can be used to predict the next few items that users would want to access, sequential pattern mining-based recommendation algorithms [23, 37, 59] have performed well in empirical studies including online product recommendation. In e-learning setting, there are some intrinsic orders for learning material in users' learning processes. The learning processes usually have some time-dependency relationships and are repeatable and periodic. The time-dependency relationship between learning materials in a learning process can reflect learner's latent material access pattern and preference. Then, using these sequential patterns [23–25, 27, 35, 37, 44, 50, 52, 60], we can predict the most probable material that a learner will access in near feature.

Hybrid recommender . A hybrid recommendation system is composed of two or more diverse recommendation techniques, including content-based filtering, collaborative filtering and sequential pattern mining techniques. The main goal of this approach is to improve performance in terms of recommendations and to overcome some of the issues that plague recommender systems, such as the cold-start and sparsity problems. Different hybridization methods [2, 12, 13, 15, 17, 17, 33, 36] have been proposed, such as the use of weighted criterion (the scores of different recommendation components are combined numerically), the use of a switching mechanism (the system chooses among recommendation components and applies the selected one) or even the presentation of the two recommendations together, leaving the decision in the user's hands. Nevertheless, a common problem with these methods is that the parameters controlling the hybridization have to be tuned.

3. A traditional-learning time predictive approach

This section presents the contribution of the paper. It is organized as follows. Subsection 3.1 presents the three types of learning sessions along with related learning contexts, learning material parameters and contextual parameters. Subsection 3.2 discusses how the learning server is aware of traditional learning sessions that have taken place. Subsection 3.3 studies the problem of estimating traditional learning times. For this purpose, a mathematical approach, which is grounded into some theoretical considerations, is introduced in a detailed way. Subsection 3.4 is concerned with implementation issues. Thus, the traditional time-estimation approach presented in subsection 3.3 is translated into algorithms.

3.1. The context of a learning environment and the context of a learning material

As stated above, stemming from various factors such as the performances of both the Internet and the electricity network, the standard of living and computer vision syndrome, we consider three types of learning sessions, namely *online learning session*, *offline learning session* and *traditional learning session*.

1. An *online learning* session can take place only when the performances of both the Internet and the electricity network are good. During an *online learning* session, the learning server collects a number of informations and the learner has the opportunity to interact with instructors and other learners.

2. An *offline learning* session can take place only when the performance of the Internet is poor and the performance of the electricity network is good. During an offline session, the learner uses a client software to access to learning materials that have been downloaded from the learning server during an online session, and the client software collects session informations (offline learning time, number of learning material used during the offline session, ...) that will be send to the learning server during the next online session. However, the learner cannot interact with instructors, nor with other learners.

3. A *traditional learning* session can take place when the performances of both the Internet and the electricity network are poor. During a traditional session, the learner uses hard copies of learning materials that have been downloaded and printed during others types of sessions.

In our study, we assume that a learner can move from one type of learning to another one, depending on various factors such as the performances of both the Internet and the electricity network, the standard of living and the effects of computer use on eye health and vision. Thus, we consider that the *learning context of a learner* consists of three parameters:

1. The percentage of the cumulative learning time over all its online sessions, also called the percentage of its online learning time for sake of simplicity.

2. The percentage of the cumulative learning time over all its offline sessions, also called the percentage of its offline learning time.

3. The percentage of the cumulative learning time over all its traditional sessions, also called the percentage of its traditional learning time.

The sum of these three percentages should be equal to 100%. similarly, we also consider the *learning context of a learner for a given learning material*. It also consists in three parameters: (1) the percentage of the cumulative online learning time for that material, (2) the percentage of the cumulative offline learning time for that material, (3) the percentage of the cumulative traditional learning time for that material. As for the learning context of a learner, the sum of these three percentages should be equal to 100%.

We also consider a number of contextual parameters which dertermine the context of a learning material : (1) *the set of semantically related concepts*: keywords, synonyms, (2) *the number of figures per size unit*, (3) *the number of figures related to African context per size unit*, (4) *the number of illustrative examples per size unit*, (5) *the number of illustrative examples related to African context per size unit*, (6) *level of interactivity*: we consider four levels denoted by 1, 2, 3 and 4, (7) *level of conviviality*: as for the interactivity parameter, we consider four levels, (8) *size*: we do not consider the number of pages for presentation (ppt) and text-based (pdf) formats, nor the time for audio (mp3) and video (mp4) formats, rather, we consider the number of lessons of the learning material for any type of format, assuming that all the lessons have equal teaching times (hypothesis 2 in section 3.3), (9) *format*: there are four formats, audio (mp3), video (mp4), presentation (ppt) and text-based (pdf), represented by digits 1, 2, 3 and 4 respectively, (10) *printability* : there are two values, non-printable and printable, represented by digits 0 and 1 respectively, (11) *downloadability* : as for the printability parameter, there are two values, non-downloadable and downloadable, represented by digits 0 and 1 respectively.

Note that, if the value of the printability parameter is 0, traditional learning cannot take place. Similarly, if the value of the downloadability parameter is 0, offline learning cannot take place.

3.2. Detecting traditional learning sessions

For the sake of scoring learning materials, the learning server should know whether or not a learner has performed traditional sessions. This can be obtained in a number of ways, including:

1. Explicitely from the learner by asking direct questions.

2. Implicitely from the learning context of the learner. If the percentage of the traditional learning time is different from zero, we might consider that he will study parts of learning materials during traditional learning sessions

3. By analyzing the behavior of the learner during the online and offline sessions.

    (a) If a learner has downloaded and printed a learning material, then we might consider that he certainly performed traditional learning sessions.

    (b) If a learner has never studied some parts of a learning material (that he has choosen) over all its offline and online sessions, i.e. either the cumulative visit time of these parts over all its online and offline sessions is close to zero or he skipped all these parts over all its offline and online sessions, we might consider that he studied these portions during traditional learning sessions

    (c) If a learner partially studied parts of a learning material over all its online and offline sessions, i.e. the cumulative learning time of these parts over all its online and offline sessions is low compared to the average time, we might consider that he also studied these portions during traditional learning sessions.

    (d) Assume that each learning material contains a course and exercices distributed equitably following the keywords of the course. Also consider a learner and a learning material for which the learning server is aware of its answers to exercices. If the percentage of exercises done by this learner is greater than the percentage of course studied during offline and online sessions, we might consider that he studied portions of that course during traditional learning sessions. Note that exercices can be done during any type of learning session. But, the answers of exercices done during traditional learning sessions should be uploaded during online learning sessions. Similarly, the answers of exercices done during offline learning sessions are automatically send to the learning server by a client software during online learning sessions.

    (e) If the learner has made fair contributions, on parts of a learning material that he has never studied online/offline, during an interaction with the instructor or during a meeting with other learners under the supervision of the instructor, we might consider that he studied these parts during traditional learning sessions. Note that only the instructor is allowed to appreciate the quality of a learner contribution.

    (f) Similarly, if the learner has made very good contributions, on parts of a learning material that he has partially studied online or offline, during an interaction with the instructor or during a meeting with other learners under the supervision of the instructor, we might consider that he also studied these parts during traditional learning sessions.

3.3. Estimating traditional learning times

This subsection introduces a mathematical approach grounded into some theoretical considerations for the problem of estimating traditional learning times. It is organized as follows. First, a number of hypotheses based on empirical observations of the learning process are introduced. They are exploited to construct the traditional time-estimation procedure proposed in this paper. Second, a number of mathematical notations to denote learning times of the three types of learning, lesson neighborhood and learner neighborhood are introduced in a detailed way. Third, a five-steps procedure, based on a number of hypotheses, learning times, lesson neighborhood, learner neighborhood, content filtering technique [38], collaborative filtering technique [7, 8, 26, 28, 32, 37, 54], similarity between two lessons and similarity between two learners, is introduced to estimate the value of $\mathrm{EnablingTime}(Lr, Ls)$. It denotes the amount of lesson learning-time for lesson Ls it takes to enable learner Lr to do all the exercises of Ls. Fourth, the estimation of $\mathrm{EnablingTime}(Lr, Ls)$ is in turn exploited to estimate traditional learning times. Fifth, equations to compute the similarity between two lessons, based on lesson parameters, are provided. Sixth, equations to compute the similarity between two learners, based on their learning context, are provided.

Hypotheses. Each learning material describes a course and is divided into parts, where each of them contains a lesson and exercises. A course is a unit of teaching, led by one or more instructors, that typically lasts one academic term. It is made of a series of lessons in a particular subject, typically leading to a qualification. A lesson is an explanation for the purpose of helping learners understand a complicated and/or concrete sub-subject of a course-subject. It may range from a lecture, to a demonstration, to a discussion or a blend of some of these common presentation methods. Some lessons may involve work by the learner, such as reading and writing or creating something, perhaps when the instructor is not present. The learner may work independently or collaborate with others. The potential format and structure of a lesson is dependent upon factors such as culture, learning objectives and the style of the individual teacher. Exercises can be viewed as a means of practicing what learners learned during lessons.

Stemming from empirical observations related to how people learn [16, 40, 45, 58], we consider a number of hypotheses stated as follows.

1. *Hypothesis 1:* The parts of a same learning material may have different numbers of pages, different lesson-lengths and different numbers of exercices.

2. *Hypothesis 2:* All the lessons of learning materials have equal teaching times. Note that: (a) this hypothesis contributes in bridging the gap between the amounts of learning-time per lesson needed by a given learner to reach a same goal for each lesson, and (b) highest similarities between lessons also contribute in bridging such gaps. Hypothesis 2 is exploited to define hypothesis 4.

3. *Hypothesis 3:* The more you learn, the more you understand and the more you acquired skills. This comes from the fact that the learning process is repeatable and periodic. Hypothesis 3 is exploited to define hypothesis 5.

4. *Hypothesis 4:* Stemming from hypothesis 2, given a learner, we assume that: (a) all the lessons require close amounts of learning time to enable her or him to do the same percentage of exercices for each lesson, and (b) the more the similarity between two lessons, the more such amounts are close. Hypothesis 4 is exploited to estimate the value of EnablingTime(Lr, Ls) through equations (6) and (7), which is in turn exploited to estimate traditional learning times.

5. *Hypothesis 5:* Stemming from hypothesis 3, we assume that the percentage of exercices done by the learner in a given part of a learning material is almost proportional to the amount of his or her learning time devoted to the corresponding lesson. Hypothesis 5 is exploited to estimate the value of EnablingTime(Lr, Ls) through equation (5).

Notations. The learning process is repeatable and periodic. The amount of lesson learning-time it takes a learner to do exercises, i.e. practicing what learned during lessons, depends on the learner's goals, interests, and abilities. Given a learner Lr and a lesson Ls of a learning material M, we consider a number of concepts and notations defined as follows:

1. OnlineTime(Lr, Ls): It denotes the cumulative online learning time of learner Lr for lesson Ls over all its online sessions. It is automatically calculated by the learning server.

2. OfflineTime(Lr, Ls): Similarly, it denotes the cumulative offline learning time of learner Lr for lesson Ls over all its offline sessions. It is automatically calculated by the client software that handles offline sessions.

3. TraditionalTime(Lr, Ls): Similarly, it denotes the cumulative traditional learning time of learner Lr for lesson Ls over all its traditional sessions. Contrary to the times of others types of learning, i.e OnlineTime(Lr, Ls) and OfflineTime(Lr, Ls), it cannot be calculated automatically. Thereby, either it is given by the learner or it is estimated by analyzing the behavior of the learner during others types of sessions, i.e. online and offline sessions.

4. CumulativeTime(Lr, Ls): It denotes the cumulative learning time of learner Lr for lesson Ls over all its learning sessions. We have :

$$\begin{aligned}\text{CumulativeTime}(Lr, Ls) = \\ \text{OnlineTime}(Lr, Ls)+ \\ \text{OfflineTime}(Lr, Ls)+ \\ \text{TraditionalTime}(Lr, Ls)\end{aligned} \tag{1}$$

If the real value of TraditionalTime(Lr, Ls) is unknown and different from zero, the real value of CumulativeTime(Lr, Ls) is also unknown and should be estimated. On the other hand, if learner Lr has not studied lesson Ls during any traditional session, i.e. TraditionalTime$(Lr, Ls) = 0$, we do not need to perform any estimation as we have:

$$\begin{aligned}\text{CumulativeTime}(Lr, Ls) = \\ \text{OnlineTime}(Lr, Ls)+ \\ \text{OfflineTime}(Lr, Ls)\end{aligned}$$

This equation holds if the value of the printability parameter is equal to 0 and the value of the downloadability parameter is equal to 1. Similarly, if the value of the printability parameter is equal to

1 and the value of the downloadability parameter is equal to 0, we have.

$$\begin{aligned}\text{CumulativeTime}(Lr, Ls) = \\ \text{OnlineTime}(Lr, Ls) + \\ \text{TraditionalTime}(Lr, Ls)\end{aligned}$$

If 0 is the value of these two parameters, we have:

$$\text{CumulativeTime}(Lr, Ls) = \text{OnlineTime}(Lr, Ls)$$

5. OnlineTime(Lr, M) : It denotes the cumulative online learning time of Lr for M over all its online sessions. Similarly, one can get the meaning of OfflineTime(Lr, M), TraditionalTime(Lr, M) and CumulativeTime(Lr, M). We have :

$$\begin{aligned}\text{CumulativeTime}(Lr, M) = \\ \text{OnlineTime}(Lr, M) + \\ \text{OfflineTime}(Lr, M) + \\ \text{TraditionalTime}(Lr, M)\end{aligned} \tag{2}$$

6. OnlineTime(Lr) : It denotes the cumulative online learning time of Lr over all its online sessions. Similarly, one can get the meaning of OfflineTime(Lr), TraditionalTime(Lr) and CumulativeTime(Lr). We have :

$$\begin{aligned}\text{CumulativeTime}(Lr) = \\ \text{OnlineTime}(Lr) + \\ \text{OfflineTime}(Lr) + \\ \text{TraditionalTime}(Lr)\end{aligned} \tag{3}$$

7. PerOfExe(Lr, Ls): It denotes the percentage of exercises of lesson Ls done by learner Lr. Similarly, PerOfExe(Lr, M) denotes the percentage of exercises of material M done by Lr. These percentages are automatically calculated by the learning server as it is aware of all the answers of learners to exercices.

8. EnablingTime(Lr, Ls): It denotes the amount of lesson learning-time for lesson Ls it takes to enable learner Lr to do all the exercises of Ls. The meaning of EnablingTime(Lr, M) is similar and we have:

$$\begin{aligned}\text{EnablingTime}(Lr, M) = \\ \sum_{Ls \in M} \text{EnablingTime}(Lr, Ls)\end{aligned} \tag{4}$$

9. LLsN(Lr, Ls) and LsN(Lr, Ls): LLsN and LsN stand respectively for *Local Lesson Neighborhood* and *Lesson Neighborhood*. The first notation denotes the neighborhood of lesson Ls for learner Lr in the material to which Ls belongs. It is made of the m lessons, belonging to the material containing Ls and having the highest similarity to lesson Ls, for which the cumulative learning time of Lr is known. It is also called the local lesson neighborhood of Ls for Lr. The second notation denotes the neighborhood of Ls for Lr in the set of all the learning materials. It is made of the m lessons, belonging to any learning material and having the highest similarity to lesson Ls, for which the cumulative learning time of Lr is known. Similarly to the first notation, it is also called the lesson neighborhood of Ls for Lr.

10. LLrN(Lr, Ls) and LrN(Lr): LLrN stands for *Local Learner Neighborhood* and LrN stands for *Learner Neighborhood*. Notation LLrN(Lr, Ls) denotes the local learner neighborhood of learner Lr for lesson Ls made of the m learners having the highest similarity to learner Lr over the set of all learners for which the cumulative learning time of lesson Ls is known. Notation LrN(Lr) denotes the learner neighborhood of learner Lr made of the m learners having the highest similarity to learner Lr over the set of all learners for which the cumulative learning time of a lesson is known.

We have the following lemma.

Lemma 1. If there is no more limitation on the number of elements of the neighborhoods defined above, we have the following inclusions:

1. LLsN$(Lr, Ls) \subseteq$ LsN(Lr, Ls)
2. LLrN$(Lr, Ls) \subseteq$ LrN(Lr)
3. LsN(Lr, Ls) is included in the set of lessons for which there exists a learner whose cumulative learning time is known.

Proof. Note that each of the following neighborhoods, LLsN(Lr, Ls), LsN(Lr, Ls), LLrN(Lr, Ls) and LrN(Lr), contains at most m elements according to its definition. Now, let's remove this size limitation: (1) LLsN(Lr, Ls) becomes the set of lessons belonging to the material containing Ls for which the cumulative learning time of Lr is known, (2) LsN(Lr, Ls) becomes the set of lessons belonging to any learning material for which the cumulative learning time of Lr is known, (3) LLrN(Lr, Ls) becomes the set of learners for which the cumulative learning time of lesson Ls is known, and (4) LrN(Lr) becomes the set of learners for which the cumulative learning time of a lesson is known. Hence the result. □

Estimating function EnablingTime. If learner Lr has done all the exercises of lesson Ls, the value of EnablingTime(Lr, Ls) is equal to

CumulativeTime(Lr, Ls). This value is known if learner Lr has not studied lesson Ls during any traditional session, i.e. TraditionalTime$(Lr, Ls) = 0$. Otherwise, it can be approximated by considering the five-steps prediction approach described in the following which can be summarized as follows. First, it attempts to estimate EnablingTime(Lr, Ls) from the value of CumulativeTime(Lr, Ls). If that last value is unknown, it makes a second attempt, based on the narrowest neighborhood. If the second attempt fails, i.e. the narrowest neighborhood is empty, it makes a third attempt based on a slightly greater neighborhood. If the third attempt still fails, a fourth attempt is made from another neighborhood that is slightly greater than the previous one. If the fourth attempt still fails, the previous step is repeated once. If the fifth attempt still fails, the approach is unable to determine a good estimation.

The idea behind the estimation proposed in the first step is borrowed from hypothesis 5. The common idea behind the estimations proposed in the second and third (resp. fourth and fifth) steps is borrowed from the content filtering [38] (resp. collaborative filtering [7, 8, 26, 28, 32, 37, 54]) technique. Thus, the proposed estimation approach can benefit from the success of these two filtering techniques. The differences between the estimations proposed in the last four steps come from the type of the elements (either lessons or learners) and the locality (local or not) of the neighborhood used in the estimation process.

Case 1: The value of CumulativeTime(Lr, Ls) is known

From hypothesis 5, We have the following approximation:

$$\mathrm{EnablingTime}(Lr, Ls) \approx \frac{\mathrm{CumulativeTime}(Lr, Ls)}{\mathrm{PerOfExe}(Lr, Ls)} \tag{5}$$

Case 2: The value of CumulativeTime(Lr, Ls) is unknown and LLsN$(Lr, Ls) \neq \emptyset$

If the cumulative learning time of learner Lr for lesson Ls is unkown, we cannot get an approximated value of EnablingTime(Lr, Ls) from equation (5). On the other hand, assume that the local lesson neighborhood of Ls for Lr is not empty, i.e. LLsN$(Lr, Ls) \neq \emptyset$. Stemming from hypothesis 4.a, we can approximate the enabling learning time of Lr for Ls as the average of the enabling learning time of Lr for lessons belonging to LLsN(Lr, Ls). Thus, we have:

$$\mathrm{EnablingTime}(Lr, Ls) \approx \frac{\sum_{L \in \mathrm{LLsN}(Lr,Ls)} \mathrm{EnablingTime}(Lr, L)}{|\mathrm{LLsN}(Lr, Ls)|} \tag{6}$$

Stemming from hypothesis 4.b, approximation (6) can be improved by taking into account the similarity between lesson Ls and each lesson belonging to LLsN(Lr, Ls). To this end, it becomes a weighted average as follows.

$$\mathrm{EnablingTime}(Lr, Ls) \approx \frac{\sum_{L \in \mathrm{LLsN}(Lr,Ls)} \mathrm{EnablingTime}(Lr, L)\mathrm{sim}(Ls, L)}{\sum_{L \in \mathrm{LLsN}(Lr,Ls)} \mathrm{sim}(Ls, L)} \tag{7}$$

It is easy to see that the sum of all the weights is equal to 1.

Case 3: LLsN$(Lr, Ls) = \emptyset$ and LsN$(Lr, Ls) \neq \emptyset$

Lemma 2 explains why the contrary of the condition of case 1 has been removed from the list of conditions of case 3.

Lemma 2. If LLsN$(Lr, Ls) = \emptyset$, the value of Cumulative-Time (Lr, Ls) is unknown.

Proof. Let M denotes the learning material to which lesson Ls belongs. Assume that LLsN$(Lr, Ls) = \emptyset$. It means that there is no lesson belonging to M for which the cumulative learning time of Lr is known. Thus, as lesson Ls belongs to M, the value of CumulativeTime (Lr, Ls) is unknown. □

The following approximation is obtained by replacing in equation (7) the neighborhood LLsN(Lr, Ls) with LsN(Lr, Ls).

$$\mathrm{EnablingTime}(Lr, Ls) \approx \frac{\sum_{L \in \mathrm{LsN}(Lr,Ls)} \mathrm{EnablingTime}(Lr, L)\mathrm{sim}(Ls, L)}{\sum_{L \in \mathrm{LsN}(Lr,Ls)} \mathrm{sim}(Ls, L)} \tag{8}$$

Case 4: LsN(Lr, Ls) $= \emptyset$ and LLrN$(Lr, Ls) \neq \emptyset$

Lemma 3 explains why the first condition of case 3, i.e. LLsN$(Lr, Ls) = \emptyset$, has been removed from the list of conditions of case 4.

Lemma 3. If LsN$(Lr, Ls) = \emptyset$, LLsN$(Lr, Ls) = \emptyset$.

Proof. Assume that LsN$(Lr, Ls) = \emptyset$. It means that there is no lesson over the set of all the learning materials for which the cumulative learning time of Lr is known. This implies that the cumulative learning time of Lr for any lesson belonging to the material containing Ls is unknown. It comes that LLsN$(Lr, Ls) = \emptyset$. □

The following approximation is obtained by replacing in equation (8) the neighborhood LsN(Lr, Ls) with LLrN(Lr, Ls) and the similarity (sim(Ls, L)) between two lessons with the similarity (sim(Lr, L)) between two learners.

$$\text{EnablingTime}(Lr, Ls) \approx \frac{\sum_{L \in \text{LLrN}(Lr,Ls)} \text{EnablingTime}(L, Ls)\text{sim}(Lr, L)}{\sum_{L \in \text{LLrN}(Lr,Ls)} \text{sim}(Lr, L)} \quad (9)$$

Case 5: $\text{LsN}(Lr, Ls) = \emptyset$, $\text{LLrN}(Lr, Ls) = \emptyset$ and $\text{LrN}(Lr) \neq \emptyset$

Lemma 4. $\text{LLsN}(Lr\prime, Ls) = \emptyset$ and $\text{LsN}(Lr\prime, Ls) \neq \emptyset$ for all learner $Lr\prime \in \text{LrN}(Lr)$.

Proof. Let $Lr\prime$ denotes a learner that belongs to $\text{LrN}(Lr)$. This implies that there exists a lesson $Ls\prime$ for which the cumulative learning time of learner $Lr\prime$ is known. It comes that $Ls\prime \in \text{LsN}(Lr\prime, Ls)$. Thus, $\text{LsN}(Lr\prime, Ls) \neq \emptyset$. On the other hand, $\text{LLrN}(Lr, Ls) = \emptyset$ means that the the cumulative learning time of lesson Ls is unknown for all learner. This implies that $\text{LLsN}(Lr\prime, Ls) = \emptyset$. Hence the result. □

Given a learner $Lr\prime$ belonging to $\text{LrN}(Lr)$, lemma 4 implies that the value of $\text{EnablingTime}(Lr\prime, Ls)$ can be approximated with equation (8). Once this approximation is done for all learner of $\text{LrN}(Lr)$, an approximation of $\text{EnablingTime}(Lr, Ls)$ is obtained from the following equation which is obtained by replacing in equation (9) the neighborhood $\text{LLrN}(Lr, Ls)$ with $\text{LrN}(Lr)$.

$$\text{EnablingTime}(Lr, Ls) \approx \frac{\sum_{L \in \text{LrN}(Lr)} \text{EnablingTime}(L, Ls)\text{sim}(Lr, L)}{\sum_{L \in \text{LrN}(Lr)} \text{sim}(Lr, L)} \quad (10)$$

Case 6: $\text{LrN}(Lr) = \emptyset$

Lemma 5 states why the first two conditions of case 5, i.e. $\text{LsN}(Lr, Ls) = \emptyset$ and $\text{LLrN}(Lr, Ls) = \emptyset$, are not considered as conditions of case 6.

Lemma 5. If $\text{LrN}(Lr) = \emptyset$, $\text{LsN}(Lr, Ls) = \emptyset$ and $\text{LLrN}(Lr, Ls) = \emptyset$.

Proof. Assume that $\text{LrN}(Lr) = \emptyset$. This means that there is no learner for which the cumulative learning time of any lesson is known. It comes that there is no learner for which the cumulative learning time of lesson Ls is known on one hand, i.e. $\text{LLrN}(Lr, Ls) = \emptyset$, and that the cumulative learning time of learner Lr for any lesson is unknown on the other hand, i.e. $\text{LsN}(Lr, Ls) = \emptyset$. □

In this case, the value of the cumulative learning time of any lesson is unknown for all learner. If $\text{LrN}(Lr) \neq \emptyset$, the value of $\text{EnablingTime}(Lr\prime, Ls)$ is approximated following the approach described from case 1 to case 5. Hence, we have the following lemma.

Lemma 6. Consider a learner Lr and a lesson Ls for which the value of $\text{EnablingTime}(Lr, Ls)$ is unknown. If $\text{LrN}(Lr) \neq \emptyset$, i.e. there exists a learner for which the cumulative learning time is known for some lesson, the value of $\text{EnablingTime}(Lr, Ls)$ can be approximated following the approach described from case 1 to case 5.

Estimating function TraditionalTime. Once an approximation of $\text{EnablingTime}(Lr, Ls)$ is calculated, we can obtain an approximation of the cumulative learning time from equation (5) as follows.

$$\text{CumulativeTime}(Lr, Ls) \approx \text{EnablingTime}(Lr, Ls)\text{PerOfExe}(Lr, Ls) \quad (11)$$

This approximation of $\text{CumulativeTime}(Lr, Ls)$ enables us to get an approximation of $\text{TraditionalTime}(Lr, Ls)$ from (1). Once all the unknown values of traditional learning times of learner Lr for lessons belonging to a same learning material M are approximated, we deduce an approximated value of $\text{TraditionalTime}(Lr, M)$ from equation (2). After doing this for each learning material that learner Lr has choosen, we deduce an approximated value of $\text{TraditionalTime}(Lr)$ from equation (3). Hence, we have the following lemma.

Lemma 7. If $\text{LrN}(Lr) \neq \emptyset$ for some learner Lr, an estimation of function TraditionalTime can be deduced from the approach described in section 3.3 and equations (1), (2), and (3).

Similarity between two lessons. We consider a number of contextual parameters which dertermine the context of a lesson. They are the same as those introduced in section 3.1 for learning materials. Note that the value of the eighth parameter is 1 for all lesson. Thus, we do not consider that parameter to determine the closeness between two lessons.

In [38, 55], the following assumptions are made, based on empirical observations regarding text. They can influence the definition of similarity of two lessons.

1. *Assumption 1*: *Rare terms are not less relevant than frequent terms*. Terms that occur frequently in one lesson, but rarely in other lessons, are more likely to be relevant to the topic of the lesson. Following that assumption, (1) the set of keywords of a lesson should be as large as possible, and (2) the set of keywords of a learning material should contain the set of keywords of all its lessons. This assumption is related to the first parameter of learning materials.

2. *Assumption 2*: *Multiple occurrences of a term in a document are not less relevant than single occurrences*. Following that assumption, the number of occurrences of keywords should be taken into account in the measure of the similarity between

two lessons. This assumption is exploited to improve the measure of the similarity between two lessons. It leads to equation (14).

3. *Assumption 3: Long documents are not preferred to short documents*. This leads to a normalization assumption. It is taken into account by hypothesis 2 of section 3.3 and by the eighth parameter of learning materials. That hypothesis states that all the lessons of learning materials have equal teaching times while that parameter measures the size of a material as the number of its lessons.

In the following, $p = 11$ denotes the number of contextual parameters. The parameter numbered i is assigned a weight denoted w_i. We assume that the sum of these weights is equal to 1.

$$\sum_{i=1}^{p} w_i = 1 \tag{12}$$

Let Ls and $Ls\prime$ be two lessons. Denote $p_i,\ i = 1,\ 2,\ \dots\ p$ (resp. $p\prime_i,\ i = 1,\ 2,\ \dots\ p$) the contextual parameters of Ls (resp. $Ls\prime$). The similarity between Ls and $Ls\prime$ is defined as follows.

$$\mathrm{sim}(Ls, Ls\prime) = \frac{w_1|p_1 \cap p\prime_1| + \sum_{i=2}^{p-4} w_i \min(p_i, p\prime_i) + \sum_{i=p-2}^{p} w_i \mathrm{equal}(p_i, p\prime_i)}{w_1|p_1 \cup p\prime_1| + \sum_{i=2}^{p-4} w_i \max(p_i, p\prime_i) + 3} \tag{13}$$

Where function min returns the minimum of its two arguments, function max returns the maximum of its two arguments and function equal returns 1 if its two arguments are equal and 0 otherwise.

Equation (13) can be improved by taking into account the number of occurrences of keywords as stated in assumption 2. Denote $\mathrm{O}(Ls, k)$ the number of occurences of keyword $k \in p_1$ in lesson Ls. Equation (13) becomes

$$\mathrm{sim}(Ls, Ls\prime) = \frac{w_1 mi + \sum_{i=2}^{p-4} w_i \min(p_i, p\prime_i) + \sum_{i=p-2}^{p} w_i \mathrm{equal}(p_i, p\prime_i)}{w_1 ma + \sum_{i=2}^{p-4} w_i \max(p_i, p\prime_i) + 3} \tag{14}$$

where

$$\begin{cases} mi = \sum_{k \in p_1 \cap p\prime_1} \min(\mathrm{O}(Ls, k), \mathrm{O}(Ls\prime, k)) \\ ma = \sum_{k \in p_1 \cup p\prime_1} \max(\mathrm{O}(Ls, k), \mathrm{O}(Ls\prime, k)) \end{cases}$$

Similarity between two learners. Three percentages are used to calculate the similarity between two learners: (1) the percentage of the online learning time with respect to the sum of the online and offline learning times, (2) the percentage of the offline learning time with respect to the sum of the online and offline learning times, and (3) the percentage of exercices done. Note that there is a slight difference between the first two percentages and those defined in section 3.1. Those introduced in section 3.1 take the traditional learning time into account while the ones introduced here do not do it for two reasons: (1) here, the traditional learning time is unknown, and (2) here, similarities between two learners are used to estimate the value of the traditional learning time.

The parameter numbered i is assigned a weight denoted we_i. We assume that the sum of these weights is equal to 1.

$$\sum_{i=1}^{3} we_i = 1 \tag{15}$$

Let Lr and $Lr\prime$ be two learners. Denote $pe_i,\ i = 1,\ 2,\ 3$ (resp. $pe\prime_i,\ i = 1,\ 2,\ 3$) the parameters of Lr (resp. $Lr\prime$). We have:

$$\begin{cases} pe_1 = \dfrac{\mathrm{OnlineTime}(Lr)}{\mathrm{OfflineTime}(Lr) + \mathrm{OnlineTime}(Lr)} \\ pe_2 = \dfrac{\mathrm{OfflineTime}(Lr)}{\mathrm{OfflineTime}(Lr) + \mathrm{OnlineTime}(Lr)} \end{cases} \tag{16}$$

The similarity between Lr and $Lr\prime$ is defined as follows.

$$\mathrm{sim}(Lr, Lr\prime) = \frac{\sum_{i=1}^{3} we_i \min(pe_i, pe\prime_i)}{\sum_{i=1}^{3} we_i \max(pe_i, pe\prime_i)} \tag{17}$$

3.4. Algorithms

In this subsection, the traditional-learning time predictive approach presented in subsection 3.3 is translated into algorithms.

Calculating the similarity between two lessons.

Function : sim(Lesson Ls, Lesson $Ls\prime$) : Calculation of the similarity between Ls and $Ls\prime$ according to equation (14)

Data contained in Ls and $Ls\prime$: The contextual parameters of Ls and $Ls\prime$: p_i and $p\prime_i$, $i = 1,\ 2,\ \dots\ 11$
The number of occurences of keyword $k \in p_1$ (resp. $k \in p\prime_1$) in lesson Ls (resp. $Ls\prime$) : $\mathrm{O}(Ls, k)$, $\mathrm{O}(Ls\prime, k)$)

Constants related to the function : The weights of the contextual parameters following equation (12): w_i, $i = 1,\ 2,\ \dots\ 11$

Output : The value of equation (14)

Statements :

```
//Calculating the value of mi
mi=0
for k ∈ p_1 ∩ p'_1 do
  mi = mi + min(O(Ls, k), O(Ls', k))
end for
//Calculating the value of ma
ma=0
for k ∈ p_1 ∪ p'_1 do
  mi = mi + max(O(Ls, k), O(Ls', k))
end for
p=11
//Calculating the numerator of equation (14)
n = w_1 mi
for i=2, 3 ... p-4 do n = n + min(p_i, p'_i) end for
for i=p-2 ... p do n = n + w_i equal(p_i, p'_i) end for
//Calculating the denominator of equation (14)
d = w_1 ma + 3
for i=2 ... p-4 do d = d + max(p_i, p'_i) end for
//Calculating and returning the final result
sim = n/d
return sim
```

Calculating the similarity between two learners.

Function : sim(Lesson Lr, Lesson $Lr\prime$) : Calculation of the similarity between two learners Lr and $Lr\prime$ according to equation (17)

Data contained in Lr and $Lr\prime$: The percentages of the online and offline learning times of Lr and $Lr\prime$ according to equation (16), and the percentages of exercices done by each of them : pe_1, pe_2, $pe_1\prime$, $pe_2\prime$, pe_3, $pe_3\prime$

Constants related to the function : The weights assigned to the three percentages following equation (15): we_i, $i = 1, 2, 3$

Output : The value of equation (17)

Statements :

```
//Calculating the numerator of equation (17)
n=0
for i=1, 2, 3 do n = n + we_i min(pe_i, pe'_i) end for
//Calculating the denominator of equation (17)
d=0
for i=1, 2, 3 do d = d + we_i max(pe_i, pe'_i) end for
//Calculating and returning the final result
sim = n/d
return sim
```

Calculating an estimation of EnablingTime(Lr, Ls).

Function : EnablingTime(*Learner Lr, Lesson Ls*) **:** Eestimation of the value of EnablingTime(Lr, Ls)

Global data : The set of lessons and the set of learners

Output : An estimation of EnablingTime(Lr, Ls) following the five-steps procedure introduced in subsection 3.3.

Statements :

```
//Case 1:
if (the value of CumulativeTime(Lr, Ls) is known)
  Estimate EnablingTime(Lr, Ls) from formula (5)
//Case 2:
else if ( LLsN(Lr, Ls) ≠ ∅)
  for L ∈ LLsN(Lr, Ls) do
    Estimate EnablingTime(Lr, L) from (5)
  end for
  Use the previous estimations of
  EnablingTime(Lr, L), L ∈ LLsN(Lr, Ls), to
  estimate EnablingTime(Lr, Ls) from formula (7)
//Case 3:
else if (LsN(Lr, Ls) ≠ ∅)
  for L ∈ LsN(Lr, Ls) do
    Estimate EnablingTime(Lr, L) from (5)
  end for
  Use the previous estimations of
  EnablingTime(Lr, L), L ∈ LsN(Lr, Ls), to
  estimate EnablingTime(Lr, Ls) from formula (8)
//Case 4:
else if (LLrN(Lr, Ls) ≠ ∅)
  for L ∈ LLrN(Lr, Ls) do
    Estimate EnablingTime(L, Ls) from (5)
  end for
  Use the previous estimations of
  EnablingTime(L, Ls), L ∈ LLrN(Lr, Ls), to
  estimate EnablingTime(Lr, Ls) from formula (9)
//Case 5:
else if (LrN(Lr) ≠ ∅)
  for Lr' ∈ LrN(Lr) do
    for L ∈ LsN(Lr', Ls) do
      Estimate EnablingTime(Lr', L) from (5)
    end for
    Use the previous estimations of
    EnablingTime(Lr', L), L ∈ LsN(Lr', Ls), to
```

36. estimate EnablingTime($Lr\prime, Ls$) from (8)
37. **end for**
38. Use the previous estimations of
39. EnablingTime($Lr\prime, Ls$), $Lr\prime \in$ LrN(Lr), to
40. estimate EnablingTime(Lr, Ls) from (10)
41. //Case 6: The estimation procedure has failed
42. **else**
43. return
44. **end if**
45. //Returning the estimation
46. return the estimation of EnablingTime(Lr, Ls)

Calculating an estimation of TraditionalTime(Lr, Ls).

Function : TraditionalTime(*Learner Lr*, *Lesson Ls*) **:** Estimation of the value of TraditionalTime(Lr, Ls)

Global data : The set of lessons and the set of learners

Output : An estimation of TraditionalTime(Lr, Ls) following equations (11) and (1).

Statements :

1. // From equation (11) we have:
2. ct = EnablingTime(Lr, Ls)PerOfExe(Lr, Ls)
3. /* Recall that OnlineTime(Lr, Ls) and OfflineTime(Lr, Ls) are automatically calculated by the learning server and the client software respectively. From equation (1) we have: */
4. $tt = ct$ − OnlineTime(Lr, Ls) − OfflineTime(Lr, Ls)
5. return tt

4. Conclusion

Material recommender system is a significant part in web-based educational systems that recommends appropriate materials to the learner. However, the existing recommendation algorithms do not apply to environments that involve the three types of learning, namely online learning, offline learning and traditional learning. As the rating process of learning materials of several recommendation algorithms are based on learning times and that online learning times and offline learning times can be obtained automatically while traditional learning times should be obtained either directly from learners or from a predictive algorithm, the problem of estimating traditional learning times is an interesting issue in such environments. To address these drawbacks of existing recommendation algorithms in this paper, a traditional-learning time predictive approach is proposed. This approach is based on a number of hypotheses inspired from some empirical observations regarding learning and text, an analysis of lesson and learner neighborhoods, content filtering technique [38] and collaborative filtering technique [7, 8, 26, 28, 32, 37, 54]. Thus, the proposed estimation approach can benefit from the success of these two filtering techniques. The proposed approach uses three types of context: (1) the context of a learning material, (2) the learning context of a learner for a given learning material, and (3) the learning context of a learner. The paper also describes how to detect traditional learning sessions and algorithms related to the proposed approach. Therefore the contribution of the paper is threefold: (1) a set of ways to detect traditional learning sessions, (2) an approach to predict traditional learning times, and (3) a translation of the proposed approach into algorithms. In the future, we intend to incorporate these algorithms into learning systems to improve the quality of recommendations for challenging environments.

Acknowledgements. The author would like to thank the editor and anonymous reviewers for their valuable comments to improve this paper.

References

[1] A. Abbas, L. Zhang, and S. U. Khan. A survey on context-aware recommender systems based on computational intelligence techniques. *Computing*, 97(7):667–690, 2015.

[2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, 17(6):734–749, 2005.

[3] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In *Recommender Systems Handbook*, pages 191–226. 2015.

[4] M. T. Banday. Web 2.0 in e-learning. *ICST Trans. e-Education e-Learning*, 1(3):e2, 2013.

[5] A. Belahcen, M. Abik, and R. Ajhoun. Decision making in the connected learning environment (CLE). *ICST Trans. e-Education e-Learning*, 3(9):e1, 2016.

[6] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. Recommender systems survey. *Knowl.-Based Syst.*, 46:109–132, 2013.

[7] J. Bobadilla, F. Serradilla, and J. Bernal. A new collaborative filtering metric that improves the behavior of recommender systems. *Knowl.-Based Syst.*, 23(6):520–528, 2010.

[8] J. Bobadilla, F. Serradilla, and A. Hernando. Collaborative filtering adapted to recommender systems of e-learning. *Knowl.-Based Syst.*, 22(4):261–265, 2009.

[9] K. D. Bollacker, S. Lawrence, and C. L. Giles. Citeseer: An autonous web agent for automatic retrieval and identification of interesting publications. In *Agents*, pages 116–123, 1998.

[10] G. Boone. Concept feature in re: Agent, an intelligent email agent. In *Agents*, pages 141–148, 1998.

[11] J. Bourdeau, T. Forissier, Y. Mazabraud, and R. Nkambou. Web-based context-aware science learning. In *Proceedings of the 24th International Conference on World Wide*

Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume, pages 1415–1418, 2015.

[12] P. Brusilovsky, A. Kobsa, and W. Nejdl, editors. *The Adaptive Web, Methods and Strategies of Web Personalization*, volume 4321 of *Lecture Notes in Computer Science*. Springer, 2007.

[13] R. D. Burke. Hybrid recommender systems: Survey and experiments. *User Model. User-Adapt. Interact.*, 12(4):331–370, 2002.

[14] L. Chen, F. Hsu, M. Chen, and Y. Hsu. Developing recommender systems with the consideration of product profitability for sellers. *Inf. Sci.*, 178(4):1032–1048, 2008.

[15] W. Chen, Z. Niu, X. Zhao, and Y. Li. A hybrid recommendation algorithm adapted in e-learning environments. *World Wide Web*, 17(2):271–284, 2014.

[16] N. R. Council. *Knowing What Students Know: The Science and Design of Educational Assessment*. The National Academies Press, Washington, DC, 2001.

[17] L. M. de Campos, J. M. Fernández-Luna, J. F. Huete, and M. A. Rueda-Morales. Combining content-based and collaborative recommendations: A hybrid approach based on bayesian networks. *Int. J. Approx. Reasoning*, 51(7):785–799, 2010.

[18] P. Do, H. Nguyen, V. T. Nguyen, and T. N. Dung. A context-aware recommendation framework in e-learning environment. In *Future Data and Security Engineering - Second International Conference, FDSE 2015, Ho Chi Minh City, Vietnam, November 23-25, 2015, Proceedings*, pages 272–284, 2015.

[19] H. Drachsler, H. G. K. Hummel, and R. Koper. Personal recommender systems for learners in lifelong learning networks: the requirements, techniques and model. *International Journal of Learning Technology (IJLT)*, 3(4):404–423, 2008.

[20] P. Dwivedi and K. K. Bharadwaj. e-learning recommender system for a group of learners based on the unified learner profile approach. *Expert Systems*, 32(2):264–276, 2015.

[21] H. Elletson and A. MacKinnon, editors. *The eLearning Africa 2014 Report*. ICWE, Germany, 2014.

[22] D. G. Feitelson and U. Yovel. Predictive ranking of computer scientists using citeseer data. *Journal of Documentation*, 60(1):44–61, 2004.

[23] P. Fournier-Viger, U. Faghihi, R. Nkambou, and E. M. Nguifo. Exploiting sequential patterns found in users' solutions and virtual tutor behavior to improve assistance in ITS. *Educational Technology & Society*, 13(1):13–24, 2010.

[24] P. Fournier-Viger, U. Faghihi, R. Nkambou, and E. M. Nguifo. Cmrules: Mining sequential rules common to several sequences. *Knowl.-Based Syst.*, 25(1):63–76, 2012.

[25] K. Gouda, M. Hassaan, and M. J. Zaki. Prism: An effective approach for frequent sequence mining via prime-block encoding. *J. Comput. Syst. Sci.*, 76(1):88–102, 2010.

[26] A. Hernando, J. Bobadilla, and F. Serradilla. Collaborative filtering based on choosing a different number of neighbors for each user. In *Handbook of Social Network Technologies and Applications*, pages 317–330. 2010.

[27] Y. Hirate and H. Yamana. Generalized sequential pattern mining with item intervals. *Journal of Computers (JCP)*, 1(3):51–60, 2006.

[28] Z. Huang, X. Lu, and H. Duan. Context-aware recommendation using rough set model and collaborative filtering. *Artif. Intell. Rev.*, 35(1):85–99, 2011.

[29] S. Isaacs and D. Hollow, editors. *The eLearning Africa 2012 Report*. ICWE, Germany, 2012.

[30] F. Isinkaye, Y. F. b, and B. O. Resnick. Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 16:261âĂŞ273, 2015.

[31] M. N. Jelassi, S. B. Yahia, and E. M. Nguifo. A personalized recommender system based on users' information in folksonomies. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013, Companion Volume*, pages 1215–1224, 2013.

[32] R. Jin, L. Si, and C. Zhai. A study of mixture models for collaborative filtering. *Inf. Retr.*, 9(3):357–382, 2006.

[33] O. Kassák, M. Kompan, and M. Bieliková. Personalized hybrid recommendation for group of users: Top-n multimedia recommender. *Inf. Process. Manage.*, 52(3):459–477, 2016.

[34] Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

[35] H. Kosorus, P. Regner, and J. Küng. On context- and sequence-aware document enrichment and retrieval towards personalized recommendations. In *Future Data and Security Engineering - First International Conference, FDSE 2014, Ho Chi Minh City, Vietnam, November 19-21, 2014, Proceedings*, pages 1–15, 2014.

[36] Q. Li and B. M. Kim. Clustering approach for hybrid recommender system. In *2003 IEEE / WIC International Conference on Web Intelligence, (WI 2003), 13-17 October 2003, Halifax, Canada*, pages 33–38, 2003.

[37] Y. Li, Z. Niu, W. Chen, and W. Zhang. Combining collaborative filtering and sequential pattern mining for recommendation in e-learning environment. In *Advances in Web-Based Learning - ICWL 2011 - 10th International Conference, Hong Kong, China, December 8-10, 2011. Proceedings*, pages 305–313, 2011.

[38] P. Lops, M. de Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*, pages 73–105. 2011.

[39] H. Mak, I. Koprinska, and J. Poon. INTIMATE: A web-based movie recommender using text categorization. In *2003 IEEE / WIC International Conference on Web Intelligence, (WI 2003), 13-17 October 2003, Halifax, Canada*, pages 602–605, 2003.

[40] R. Marra and B. Palmer. Encouraging intellectual growth: Senior college student profiles. *Journal of Adult Development*, 11(2):111–122, 2004.

[41] R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *ACM DL*, pages 195–204, 2000.

[42] R. Mukherjee, G. Jónsdóttir, S. Sen, and P. S. Dutta. MOVIES2GO: an online voting based movie recommender system. In *Agents*, pages 114–115, 2001.

[43] W. Nadee. *Modeling user profiles for recommender systems PHD filter*. PhD thesis, Science and Engineering Faculty, Queensland University of Technology, 2016.

[44] R. Nkambou, P. Fournier-Viger, and E. M. Nguifo. Learning task models in ill-defined domain using an hybrid knowledge discovery framework. *Knowl.-Based Syst.*, 24(1):176–185, 2011.

[45] C. on Developments in the Science of Learning, with additional material from the Committee on Learning Research, and N. R. C. E. Practice. *How People Learn: Brain, Mind, Experience, and School: Expanded Edition*. The National Academies Press, 2000.

[46] D. Osikoya and A. Oluwalani. An android application for animated lecture retrieval in e-learning. *ICST Trans. e-Education e-Learning*, 1(3):e4, 2013.

[47] R. M. Pai, S. V. Kolekar, and M. M. M. Pai. Web log pre-processing and analysis for generation of learning profiles in adaptive e-learning. *ICST Trans. e-Education e-Learning*, 3(9):e5, 2016.

[48] U. Panniello, A. Tuzhilin, and M. Gorgoglione. Comparing context-aware recommender systems in terms of accuracy and diversity. *User Model. User-Adapt. Interact.*, 24(1-2):35–65, 2014.

[49] B. Pathak, R. S. Garfinkel, R. D. Gopal, R. Venkatesan, and F. Yin. Empirical analysis of the impact of recommender systems on sales. *J. of Management Information Systems*, 27(2):159–188, 2010.

[50] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Trans. Knowl. Data Eng.*, 16(11):1424–1440, 2004.

[51] P. Pu, L. Chen, and R. Hu. A user-centric evaluation framework for recommender systems. In *Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011*, pages 157–164, 2011.

[52] J. Rabatel, S. Bringay, and P. Poncelet. Mining sequential patterns: A context-aware approach. In *Advances in Knowledge Discovery and Management - Volume 3 [Best of EGC 2011, Brest, France].*, pages 23–41, 2011.

[53] P. Resnick and H. R. Varian. Recommender systems - introduction to the special section. *Commun. ACM*, 40(3):56–58, 1997.

[54] M. Salehi, I. N. Kamalabadi, and M. B. G. Ghoushchi. Personalized recommendation of learning material using sequential pattern mining and attribute based collaborative filtering. *Education and Information Technologies (EAIT)*, 19(4):713–735, 2014.

[55] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1989.

[56] J. B. Schafer, J. A. Konstan, and J. Riedl. Recommender systems in e-commerce. In *EC*, pages 158–166, 1999.

[57] S. Wei, N. Ye, and Q. Zhang. Time-aware collaborative filtering for recommender systems. In *Pattern Recognition - Chinese Conference, CCPR 2012, Beijing, China, September 24-26, 2012. Proceedings*, pages 663–670, 2012.

[58] K. R. Wirthand and D. Perkins. Learning to learn. *http://www.macalester.edu/geology/wirth/CourseMaterials.html*, (accessed on 24 Jun 2016).

[59] G. Yap, X. Li, and P. S. Yu. Effective next-items recommendation via personalized sequential pattern mining. In *Database Systems for Advanced Applications - 17th International Conference, DASFAA 2012, Busan, South Korea, April 15-19, 2012, Proceedings, Part II*, pages 48–64, 2012.

[60] M. J. Zaki. SPADE: an efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1/2):31–60, 2001.

[61] C. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web, WWW 2005, Chiba, Japan, May 10-14, 2005*, pages 22–32, 2005.