The Use of Time Dimension in Recommender Systems for Learning

Eduardo José de Borba[1], Isabela Gasparini[1] and Daniel Lichtnow[2]
[1]Graduate Program in Applied Computing (PPGCA), Department of Computer Science (DCC),
Santa Catarina State University (UDESC), Paulo Malschitzki 200, Joinville, Brazil
[2]Polytechnic School, Federal University of Santa Maria (UFSM), Av. Roraima 1000, Santa Maria, Brazil

Keywords: Recommender System, Context-aware, Time, Learning.

Abstract: When the amount of learning objects is huge, especially in the e-learning context, users could suffer cognitive overload. That way, users cannot find useful items and might feel lost in the environment. Recommender systems are tools that suggest items to users that best match their interests and needs. However, traditional recommender systems are not enough for learning, because this domain needs more personalization for each user profile and context. For this purpose, this work investigates Time-Aware Recommender Systems (Context-aware Recommender Systems that uses time dimension) for learning. Based on a set of categories (defined in previous works) of how time is used in Recommender Systems regardless of their domain, scenarios were defined that help illustrate and explain how each category could be applied in learning domain. As a result, a Recommender System for learning is proposed. It combines Content-Based and Collaborative Filtering approaches in a Hybrid algorithm that considers time in Pre-Filtering and Post-Filtering phases.

1 INTRODUCTION

Nowadays, there are distinct educational approaches, e.g. online learning, blended learning, face-to-face learning, etc. All these approaches can benefit from the learning management systems, (also called e-learning systems) for the administration, documentation, tracking, reporting and delivery of electronic educational technology. In e-learning systems when the number of available materials and learning objects is huge, students may feel lost and may not find relevant objects to study. Moreover, there is great probability that some learning materials never get studied.

For this purpose, researchers have applied personalization techniques to select the best items for each student, considering student's knowledge, goals, preferences and needs (Brusilovsky, 1998). Recommender Systems (RS) can help with these problems, suggesting items to users using information they have about users and items and about how item characteristics meet users' needs.

Context-aware RS (CARS) are an evolution of traditional RS that apply context information to improve the quality of recommendations. Among all dimensions that represent context, time has the advantage of being easy to capture and has the potential to improve the quality of recommendation (Campos et al., 2014).

This work aims to identify how Time-aware RS (Context-aware RS that uses time context) can be used in the learning domain. For this purpose, seven categories of how time can be used in RS are presented, based on previous works in the area (Borba et al., 2017). Based on these seven categories, the application of time in the e-learning domain is presented through different scenarios.

This work is organized as follows. Section 2 presents Background of this research. Section 3 details the seven categories on the use of time in RS. Section 4 discusses the Related Works. Section 5 defines scenarios for each of the categories introduced in section 3. Section 6 presents a Proposal that applies time context in Recommender Systems and Section 7 presents Conclusions and Future Work.

2 BACKGROUND

This section presents the main concepts related to Recommender Systems for Learning. Firstly, the definition of Recommender Systems and their traditional approaches is presented. Followed by

Borba, E., Gasparini, I. and Lichtnow, D.
The Use of Time Dimension in Recommender Systems for Learning.
DOI: 10.5220/0006312606000609
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 2*, pages 600-609
ISBN: 978-989-758-248-6

Context-aware Recommender Systems and Time-Aware Recommender Systems.

2.1 Recommender Systems

Recommender Systems are computational tools that provide personalized suggestions to users (Ricci et al., 2011). This means that as recommendation each user receives a different set of items based on his/her preferences. In recent years, interest in Recommender Systems applications is growing strongly (Adomavicius and Tuzhilin, 2005; Beel et al., 2016). Examples of these applications are recommendation of Books, CDs, DVDs, etc., in e-commerce like Amazon or EBAY, recommendation of movies like MovieLens or Netflix; recommendation of songs in music websites like Last.fm or Spotify; friend's recommendations in social networks like Facebook.

Recommender Systems emerged as an independent area in the mid-1990s (Adomavicius and Tuzhilin, 2005). Others areas are usually involved, e.g., Information Retrieval, Approximation Theory, Artificial Intelligence, etc.

Recommender Systems are formally represented as follows:

$$F: U \ x \ I \rightarrow R$$

Where F is the function that predicts the rating for an unknown item, U represents the users, I represents the items and R denotes an ordered set of predicted ratings.

Traditional approaches of Recommender Systems are (Adomavicius and Tuzhilin, 2005): Content-based, Collaborative Filtering, Knowledge-based, Demographic and Hybrid.

Content-based is the approach where the user receives a recommendation of items similar to the ones he had interest in, in the past (Lops et al., 2011). It usually consists of comparing the description of the items (a set of keywords) to the users' profile (another set of keywords) and recommending the most suitable item(s). That is why this approach is related with Information Filtering techniques, like TF-IDF or Cosine (Adomavicius and Tuzhilin, 2005). The main advantages of Content-based approach are (Lops et al., 2011): (1) no dependence on an active community of users and (2) no item cold-start. The main drawbacks of this approach are (Lops et al., 2011): (1) user cold-start and (2) overspecialization.

Collaborative Filtering approach recommends items to a user based on what other users - with similar tastes - have interest in (Jannach et al., 2011). It is the automatization of "word of mouth", where the RS tries to predict item utility to the user based on the utility of this item to users with similar tastes to him/her. The main advantage of this approach is Serendipity (Jannach et al., 2011). The main drawbacks are (Jannach et al., 2011): (1) dependence on an active community of users, (2) User cold-start, (3) Item cold-start and (4) Black sheep.

Knowledge-based approach recommends items to users based on the knowledge about how item features matches user needs and how useful this item should be (Felfernig et al., 2011). This approach is usually applied to improve the recommendation precision or in cases where the other approaches have problems. This approach should be chosen where domain allows the representation of knowledge through structures easy read by computers, like ontologies (Adomavicius and Tuzhilin, 2005). The main drawback of this approach is that it needs the knowledge acquire (Adomavicius and Tuzhilin, 2005).

Demographic approach recommends based on the user's demographic profile, like age, gender, nationality, etc. This approach uses a recommendation by demographic classes, in which users are classified through stereotypes (Burke, 2002). It considers that different recommendations should be made to different stereotypes. The main advantage of this approach is to recommend items according users age, gender, culture, etc. (Burke, 2002). The main drawbacks are (Burke, 2002): (1) assuming that users with similar demographics have similar tastes, (2) there are few works in literature about this.

Hybrid approach combines the mentioned approaches to recommend items to users. The objective is to group the advantages of these approaches to improve the recommendation quality and with fewer drawbacks of any individual one (Burke, 2002). Burke (2002) suggests some combinations of the approaches, for example: *Weighted*, *Switching*, *Cascade* and *Mixed*. In *Weighted*, the predicted ratings of several recommendation techniques are combined and each one has a different weight. In *Switching*, the system changes through different recommendation techniques depending on the current situation. In *Cascade*, one recommender refines the recommendations given by another. In *Mixed*, all combined approaches are used and the results are presented in the same ranking.

2.2 Context-Aware Recommender Systems

Traditional RS considers only users and items to recommend, but it does not consider the context in which the users are. According to Dey (2001), context is any information that can be used to characterize the situation of an entity. In RS, entities can be the users and the items.

Context-Aware RS are formally represented as:

$$F: U \; x \; I \; x \; C \rightarrow R$$

Where F is the function that predicts the rating for an unknown item, U represents the users, I represents the items, C represents the context and R denotes an ordered set of predicted ratings.

Several authors defined different set of dimensions that could represent context (Schilit et al., 1994; Chen and Kotz, 2000; Zimmermann et al., 2007). In this work, we follow Schimidt et al. (1999) that defines the following dimensions:

- Information on the user, e.g., users' habits, users' emotional state, etc.;
- User's social environment, e.g., co-location with others users, social interaction in social networks, etc.;
- User's tasks, e.g., general goals, whether it is a defined task or random activity, etc.;
- Location, e.g., absolute position, whether the user is at home or office, etc.;
- Physical conditions, e.g., noise, light, etc.;
- Infrastructure, e.g., network bandwidth, type of device, etc.;
- Time, that could be categorical, e.g., Time of the day (Morning, Afternoon, Evening), or continuous, e.g., a timestamp like "June 1st, 2016 at 17:14:36".

Adomacivius and Tuzhilin (2011) define three paradigms of context in the recommendation process:

- Contextual Pre-Filtering, where the context filters the data that represents the user and then a traditional RS approach is applied;
- Contextual Post-Filtering, where a traditional RS approach is applied and then the result is filtered according to the context;
- Contextual Modelling, in which the context is applied directly in the recommendation algorithm.

Verbert et al. (2012) say that, in e-learning, RS traditional approaches are not enough to recommend properly items to students, because this domain offers some specific characteristics that are not covered by these approaches. For example, it is much more dangerous recommend a bad material to a student, which could demotivate him/her to study, than recommend a bad product in an e-commerce system. According Verbert et al. (2012) this application domain requires a major level of personalization. Using some context dimensions is an alternative to improve the personalization of e-learning environments, recommending properly to actual student situation, e.g., Learning History, Environment, Timing and Accessible Resources (Verbert et al., 2012).

The next section presents a specific kind of Context-Aware Recommender Systems that uses time context to recommend. This kind of RS could also be used with others context dimensions.

2.3 Time-Aware Recommender Systems

Among all context dimensions, time has an advantage to be easy to capture, considering that almost every device has a clock that could capture the timestamp when an interaction occurs. Besides that, works in this area showed that the context of time has potential to improve recommendation quality (Campos et al., 2014). This kind of RS is called Time-Aware Recommender Systems (TARS). TARS are formally represented as:

$$F: U \; x \; I \; x \; T \rightarrow R$$

Where F is the function that predicts the rating for an unknown item, U represents the users, I represents the items, T represents time context and R denotes an ordered set of predicted ratings.

According to Merriam-Webster dictionary, time is "a non-spatial continuum that is measured in terms of events that succeed one after another from past through present to future" (Merriam-Webster, 2016). This enables to establish an order to time events.

As seen in section 2.2, time may be a continuous or a categorical variable. Continuous variables are those that represents the exact time at which items are rated/consumed (Campos et al., 2014). Categorical variables are calculated regarding time periods of interest in the recommendation (Campos et al., 2014). Also, it can be represented in several time units, e.g., seconds, minutes, hours, days, months, years, etc. Time units are hierarchical, e.g., 1 day has 24 hours, 1 hour has 60 minutes, 1 minute has 60 seconds.

3 USE OF TIME CATEGORIES

A Systematic Mapping was conducted in previous work (Borba et al., 2017), using Peterson et al. (2008) methodology, aiming to explore Time-Aware Recommender Systems. The main research question defined was: How the time is used in Context-aware Recommender Systems? To answer the main research question, three secondary research questions were defined: (1) How recommender algorithms use time? (2) What are the differences about the use of time in different application domains? (3) What others dimensions of context are used to be applied together with the time dimension?

It is important to emphasize that this previous work did not consider only e-learning recommender systems. After all the process of selection of papers, 88 papers were considered to answer the research question. Between another analysis, like recommendation approach (Content-based, Collaborative Filtering, Hybrid, etc.) or representation of time (categorical or continuous), in this previous work, it were observed seven categories of how the time is used in Recommender Systems. Here, an overview of these seven categories, following (Borba et al., 2017), is presented. Section 5 details each one in depth, and explains how it could be applied in RS for e-learning.

- **Restriction:** the time is used to restrain which items are recommended. Thus, the RS matches the user's available time with time required to use the item. Examples: recommend only restaurants that are open when user's going to have lunch; recommend a movie with length less or equal to user's available time.
- **Micro-profile:** the user has distinct profiles for each time. Here, time is usually categorical, so the user has a profile for weekdays and another profile for weekends, or the user has a profile for morning, a profile for afternoon and another for evening. Example: recommend a mobile app to the user at Sunday morning based only in apps used by this user in past Sunday mornings.
- **Bias:** time is the third dimension of a User *x* Item matrix. So, collaborative filtering has more information to compare users, find k-neighbours and predict user's rating to a non-viewed item. Example: Koren (2009) proposes a Tensor Factorization strategy using a User *x* Item *x* Time tensor.
- **Decay:** the time is used as a decay factor, in which old interactions are less important than new ones. Example: take into account the use of RS techniques in E-commerce, consider items the user searched recently more important for producing recommendation.
- **Time Rating:** time is considered by the RS to infer user's preferences, i.e., the more the user stays at the item, more he likes it. It means that time gives feedback of a user to an item implicitly, i.e., without need of user rate the item. Example: at a shopping mall, the more a user stays at some store, more he likes it, and the RS could recommend products of this store when it has sales.
- **Novelty:** only new items could be recommended. Thus, the RS has a threshold and items older than that are not recommended. Example: in news website, it's more precise to recommend news of, at least, one day ago.
- **Sequence:** the RS observe items that are usually consumed one after the other. So, if the first one of the sequence is consumed, the second one should probably be consumed too. Example: in music recommendation, songs of the same album are most likely to be heard together, so if the user selects one of them, the next one should be recommended.

Figure 1 presents recommendation process in CARS. **Extraction** and **Recommendation** phases are common to every RS, while **Pre-Filtering** and **Post-Filtering** are more related to CARS.

In this model, Time Rating category appears in the first phase, as information extraction in an implicit approach. Decay and Micro-profiles appears in Pre-Filtering phase, and in this phase, we could include many other Pre-Filtering strategies using other context dimensions. Bias category appears in Recommendation phase, always applied with Collaborative Filtering, or Hybrid RS that uses Collaborative Filtering. Finally, Novelty, Restriction and Sequence categories are classified in Post-Filtering phase, and in this phase, we could also include other Post-Filtering strategies using other context dimensions.

It is worth saying that is not necessary to use all these categories of the use of time. It is possible to implement a TARS that uses just one category, or a combination of two or more strategies.

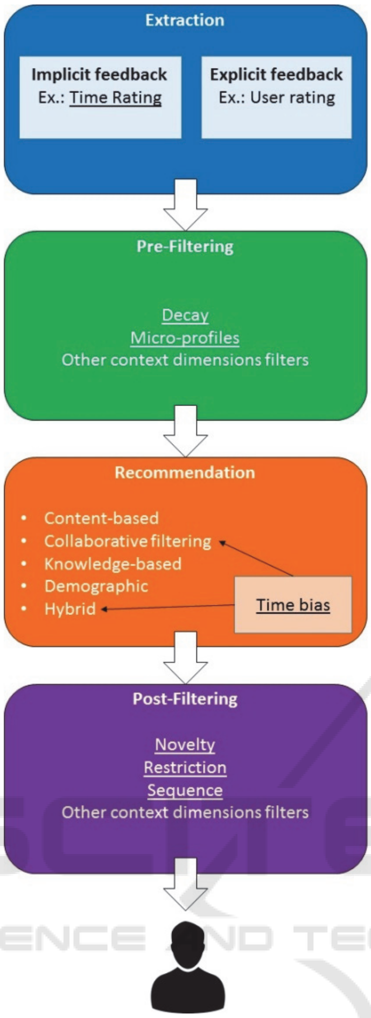

Figure 1: Recommendation model with use of time categories.

4 RELATED WORK

In this section are presented works that apply time context in the recommendation process. These works were chosen to represent some of the seven categories described in section 3, but not all categories have works in e-learning domain.

The first work is described in Gallego et al. (2012) that define a proactive Context-Aware RS that recommends learning objects to teachers and scientists that will produce learning resources the students will consume. The recommendation process is divided into three phases: (1) Generation of social context information related to the users in the environment; (2) Current situation is analysed, considering User's social environment, Location, Time and User's task; (3) Suitability of learning materials to be recommended is analysed. Time in this work is used like Restriction; the RS tries to find learning materials that matches actual user's time.

Chen et al. (2012) propose a hybrid RS to recommend learning items in users' learning processes. The proposed method consists of two steps: (1) discovering content-related items using Collaborative Filtering approach; and (2) applying the item sets to sequential pattern mining algorithm to filter items according to common learning sequences. Time in this work is used as Sequence, RS compares items in common learning sequences in order to find items that are usually consumed together and decide which one recommend.

Luo et al. (2009) propose a Context-Aware resource recommendation model and relevant recommendation algorithm to pervasive learning environments. The calculation of relevant items to be recommended can be divided in two: (1) Content-based and Collaborative Filtering are combined together, meanwhile learners' historical sequential patterns of resource accessing are also considered to further improve the accuracy of recommendation; (2) Connection type and Time satisfaction degree are calculated, considering other relevant contexts. The two parts are combined and results in Top-N items recommended. Time in this work is used as Sequence, considering learning materials order to improve recommendation, and Decay, giving less weight to older accesses.

Benlamri and Zhang (2014) propose a knowledge-driven recommender for mobile learning on the Semantic Web. This work uses an approach for context integration and aggregation using an upper ontology space and a unified reasoning mechanism. Time is used as Restriction, where learning resources have expected learning time and the approach considers this to recommend more properly to each user.

In Related Works, it is possible to observe that, in learning, most common uses of time is related to Restriction, Sequence and Decay categories. We do not find other categories of use of time in the learning domain.

5 USE OF TIME IN LEARNING RECOMMENDER SYSTEMS

In this section, we describe in depth each category described in section 3 and see how each of them could be used in RS for e-learning. For this purpose, we use scenarios to represent possible interactions of learners with an e-learning system. Each category is demonstrated in a distinct scenario.

We consider that recommender system can help in different educational approaches and situations. For example, in online learning, there are courses that are open and do not have schedule to start or to finish (i.e. classes that student can study whenever he/she wants). But also there are courses where there is a schedule to start and to finish. Both cases can benefit of recommender systems. In blended learning of face-to-face learning, the e-learning system can be used as a support of the classes, e.g. in the classroom – with technological resources, or out of the class – students homework, additional study, etc.

We consider this distinction of the different learning situations important, to present some scenarios in the next sections.

5.1 Restriction

The Restriction category uses the time to restrain which items are recommended. To understand how this category could be applied in learning domain, we consider a student who is going to study. The environment asks the student about how much time he intend to study. The student indicates that he is going to study for 3 hours (it is possible to thing in a system that uses information about user to infer this without ask the user about available time for studying).

The RS knows about the items that the student already accessed, so it should suppose that these are no longer need by the user. The RS also knows that the learning style of this user is visual, so it tries to recommend only videos to him/her. If no video is available, then the RS recommends other types of items. After applying a traditional RS approach to select the videos that best matches the user profile, the RS filters the list of recommendation, removing all videos that goes over 3 hours.

Thus in ***scenario 1***, the student watches a video that is 1h45min longer. Then, the student has 1h15min left. The next time the student asks for recommendations, the RS filters videos that go up to 1h15min. This process goes on and on until student's available time is over.

5.2 Micro-profile

Micro-profile category uses time to create different profiles of a user. Then, the user must have two or more profiles, depending on time, and the RS selects which profile are going to be used, based on some criteria.

Thus, in ***scenario 2*** there is a RS in an learning management system that uses Content-based approach. This RS represents students' profile as a set of keywords and each of the items another set of keywords. Student' keywords come from the items he/she liked (rated positively). However, item's keywords are the words that most appeared in the material and it is discovered through an algorithm called TF-IDF (Term Frequency – Inverse Document Frequency).

In this scenario, a teacher uses the learning management system described above to support his/her in-person classes (face-to-face learning). He/she provides papers, presentations, links, games, etc., that may help student while studying.

The RS using Micro-profile strategy could split student's profile in three. One for the time (period) of the classes, other for weekdays (regarding the time out of the class), and other for the weekends. The RS knows what items the student access in each of these time periods. Then, it will recommend items during the face-to-face classes based on items the student accessed during classes, will recommend to him in weekdays out of class based on items accessed in this period and recommend in weekends based on weekend's access, using Content-based approach.

The RS might found out that, for example, one of the students likes to see complementary materials, like presentations while in the classroom to go along with teacher presentation. However, he likes more complete and complex materials to study in depth the subject while on weekdays. Moreover, the student wants short videos in the weekends where he/she will not spend much time studying. These preferences are reflect in each of students' profiles, so the recommender system is going to understand them and improve its recommendations.

5.3 Bias

In Bias category, time is the third dimension of the User x Item matrix, and it is only applied in Collaborative Filtering approach. Time improves the process of finding the k-users more similar to the one that will receive recommendation and the prediction of ratings to non-viewed items.

Thus, in ***scenario 3*** there is an online course, available for six months, and totally non-presential. Despite of there is a schedule to end the course, the system allows students entry at any time. Two users started the course in different times: John started two months ago and Stuart started five months ago. Also, both use to study 2 hour by day. Items that Stuart accesses now are probably different of the items John accesses now, because Stuart is forward in the course.

In this case, the RS using time as Bias category compares John's profile today with Stuart's profile of three months ago (when he was also on the second month of the course). When comparing these two profiles, the RS finds out that John and Stuart are similar users, the RS can use Stuart ratings of three months ago to recommend items that John might like. Using this strategy, the RS does not recommend items to John based on what he is studying now, that could be too advanced to him. Instead, recommends items that are probably on the same topic of what John is studying, based in the assumption that these two users are considered similar.

5.4 Decay

Decay category uses time as decay factor to user's interaction, i.e., the older the interaction, less important it is.

Thus, in ***scenario 4,*** student Frank is enrolled in a discipline of Data Structure, that lasts one semester and that has four main topics (stack, queue, list and tree). In this discipline, there are four tests, one for each topic. Also, suppose that the RS in this discipline uses Content-based approach, i.e., recommends items similar to the ones the student accessed.

Before the first test, Frank only studied items about stacks, so he might receive only recommendations about stack. After the first test, Frank starts studying the second topic: queues. If the RS keeps recommending only stacks, the user will probably not like the recommendations he receives.

The RS, using Decay category, gives less weight to the old items that Frank accessed about stacks and gives more weight to the new items about queues. There still possibilities that Frank receives recommendations about stacks, but the RS are going to prioritize the new items about queues to recommend materials.

5.5 Time Rating

Time rating category is related to the fact that RS uses time to understand user's preferences. For example, if the user stays a long time in an item, this means to the RS that the user likes it or has interesting in it. But if the user stays small time in the item, it means that the user doesn't like it or does not have interesting in it.

Thus, in ***scenario 5,*** there is a RS in an e-learning environment that uses Collaborative Filtering approach. This approach requires an active community of students and requires feedback of the Students to the items. The feedback is usually made through explicit rating, e.g., from 1 to 5 stars.

Suppose that this environment have an active community, but the students rarely rate the items that they access. In this scenario, time is useful to receive implicit feedback about how the student liked this item.

To exemplify, considering two students Anna and Bruce and that all items were created with the same length and therefore the students will spend the same time to use an item. A problem that must be considered by the RS is that some students usually stay more in some items, while others stays less in the same items. Anna has an average time of 30 minutes per item. Bruce has an average time of 5 minutes per item. If Anna stays 15 minutes in the item *I*, this probably means she did not likes much this item. If Bruce stays 15 minutes in the item *I*, this probably means he liked this item.

To treat this problem the RS could be use the following equation to calculate how much the student *u* like item *i*, similar to the bias strategy used by Koren (2009):

$$R_{u,i} = \frac{t_{u,i}}{\overline{t_u}}$$

Where $R_{u,i}$ is the rating calculated of user u to item i, $t_{u,i}$ is the time (in minutes) that the user u stayed in the item i, and $\overline{t_u}$ is the average time of user u in the items of the system. This equation express that if the user u stays in the item i more than its average, the rating calculated is more than 1, so this means the user like it. But, the user u stays in item i less than its average, the rating is between 0 and 1, meaning the user didn't like it.

5.6 Novelty

Novelty category uses time to filter only new items to be recommended. The RS knows when the items were created and compares them with actual time to decide if the item should or not be recommended.

Thus, in ***scenario 6*** the student Fernando signs-up to a course about new technologies, like HTML 5, CSS 3, Ruby on Rails, Angular, etc. This subject is in constant changing, because these actual technologies are been updated and upgraded very frequently. This course is updated every time one of these technologies changes and the course administrator tells the system in metadata when this item was created.

The RS, using Novelty category, would recommend only items that are new to the system and up-to-date with the technologies. This kind of RS in e-learning does not define a threshold like news recommendation, because does not make sense to ignore items just from it timestamps. But if two items are similar to each other, the newer will be recommended. Also, if some item is too old comparing with the others, it would never be recommended.

5.7 Sequence

Sequence category observes items that are used in a sequential pattern to improve recommendation process. If the RS observes a set of items that is accessed in a specific order, the system should recommend them in this order to the user.

This, in ***scenario 7*** there is a short course of Algorithms that occurs one time for semester and lasts a month. This short course has 30 materials numbered from 1 to 30, within papers, links, images, videos, etc.

In the first semester of 2016, 80 students enrolled the short course and RS observed that the learning path most frequent was:

$$1 \rightarrow 9 \rightarrow 7 \rightarrow 15 \rightarrow 23 \rightarrow 12 \rightarrow 25$$

In the second semester of 2016, the RS, using Sequence category, starts the short course recommending the item 1, followed by item 9, and so on. This means RS uses the learning path learnt from the last semester to recommend material to new students.

6 PROPOSAL SYSTEM

Taking into account the scenarios presented in section 5, it is possible to define a recommender system architecture that consider time in its recommendation process. In this sense, the system can use time in distinct ways aiming to improve the quality of recommended items.

6.1 Architecture Overview

The components of system architecture consists of a module to maintain learning objects (include, remove, update and assign metadata) and a recommender module.

6.1.1 Maintain Learning Objects Module

This module allows inserting and maintaining learning objects. The learning objects are described using LOM – Learning Object Metadata (IEEE, 2002). Taking into account LOM metadata allows improving the recommendation in some scenarios presented in section 5.

Thus, metadata 2.3.3 of LOM (*Date*, that stores when a learning object were created and indicate how old a learning object is), is very important to Novelty that considers more important to recommend newer items. Besides metadata 4.7 (*Duration Time*, continuous time a learning object takes when played at intended speed) and 5.9 (*Typical Learning Time*, approximate or typical time it takes to work with or through this learning object for the typical intended target audience) are very important in Restriction.

6.1.2 The Recommender Module

The Recommender Module consists of a module that uses a Hybrid Approach to generate the recommendation. This hybrid approach combines Collaborative Filtering and Content-Based approach.

This is necessary because we consider distinct use: (1) in short courses with few users/students and (2) long duration courses with many students where it is possible to build a user community. In (1) content-based approach seems more suitable, because it is more difficult to have enough users to generate the recommendation. While in (2) Collaborative Filtering can be applied and may improve recommendation quality as time goes by.

Taking into account Burke (2002), the hybridization method chosen is mixed, where recommendation from several different recommenders are presented at the same time. Burke (2002) emphasizes that "mixed hybrid avoids the "new item" start-up problem: the content-based component can be relied on to recommend new shows on the basis of their descriptions even if they have not been rated by anyone." It does not get around the "new user" start-up problem, since both the content and collaborative methods need some data about user preferences to get off the ground, but if such a system is integrated into other source of information (e.g. social network, digital television, etc.) it could track user's behaviour and build his profile accordingly.

The Proposal System applies Collaborative Filtering and Content-Based approaches separately, each of them applying Pre-Filtering and Post-

Filtering with time dimension. For Pre-Filtering, the RS uses Decay category and, for Post-Filtering, uses Restriction and Novelty. The last one (Novelty) is only used if specified by the course manager, because it is specific to some subject.

In Decay Pre-Filtering, each user interaction (consumption or rating) are evaluated in terms of how old it is. The older the interaction, less weight the system gives to it. In this way, it considers more interactions that happened recently and items the user is studying in the last few days. In Restriction, the items will be evaluated in terms of Duration Time and Typical Learning Time metadata of LOM, matching it with user's available time. Moreover, in Novelty the items recommended will be evaluated in terms of age, if this is indicated in the system configuration, taking in account Date metadata of LOM and current data.

6.2 A Scenario of Use

In this scenario, we take into account the Proposal System, applied to a short online course about Introduction to Algorithms. It's a course that, in average, is two months long, but students can join in every time of the year and each one goes in his/her own speed. There is no tests to evaluate students, but there is plenty exercises to each user evaluate himself/herself. In this course, there are more than 300 active users every day. Also, there are 150 learning objects available and characterized with LOM metadata. It's possible to tell that both Collaborative Filtering and Content-Based approaches can be applied in this scenario.

Suppose a user of this system is already in the middle of the course. This user access twice a week the course and spends, in average, one hour and a half each time.

The Proposal System will show recommendations to this user that do not exceed one hour and a half, using time as Restriction to the list of recommendations. In Content-based list, we have items similar to the ones the user last accessed, because of using the time as Decay. While in Collaborative Filtering, even giving more weight to last items, it's possible to have surprising recommendations, because it is calculated based on items that other users with similar tastes liked. In this scenario, there is no need to apply Novelty category.

Figure 2 shows a prototype of an interface of the Proposal System, where recommendations are divided into **Based on contents you accessed** (Content-Based approach) and **Based on users similar to you** (Collaborative Filtering approach). Note that, although Hybrid approach is used, Content based and Collaborative Filtering are calculated apart and shown apart in interface. Also, it's important to explain to the user where these recommendations came from, so he/she might have more trust on the items recommended.

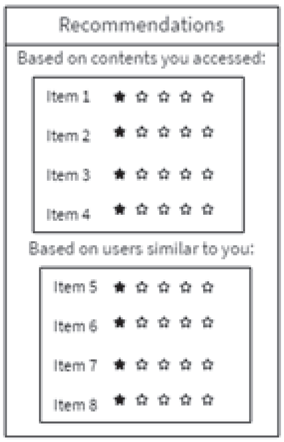

Figure 2: Prototype of Recommendations.

7 CONCLUSION

In this paper is described the use of time context in Recommendation Systems (RS) for learning. Time context has demonstrated its impact in RS, improving recommendation quality, but in learning situations, few works were found that uses this dimension.

In the present work, we take into account seven categories of how time can be used in Recommender System algorithms, based in our previous works. The proposed scenarios illustrate the use of time in learning situations and help to better explain each category of time.

Based in this work, a Recommender System architecture is proposed, that combines Content-Based and Collaborative Filtering approaches in a Hybrid algorithm. The system proposed uses time in three different ways (Decay, Restriction and Novelty) and takes advantage of some information of LOM, a set of metadata used to represent Learning Objects, for example, Date, Duration and Typical Learning Time.

As future work, the proposed system should be detailed, implemented and tested using a real environment with active users, as well as others combinations between Recommendations approach and uses of time.

REFERENCES

Adomavicius, G.; Tuzhilin, A. 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering, vol. 17, no. 6.

Adomavicius, G; Tuzhilin, A. 2011. Context-aware Recommender Systems. In: Ricci, F. (Org.); Rokach, L. (Org.); Shapira, B. (Org.); Kantor, P. B. (Org.). Recommender System: Handbook. Springer.

Beel, J.; Breitinger, C.; Langer, S.; Lommatzsch, A.; Gipp, B. 2016. Towards reproducibility in recommender-systems research. User Model User-Adap Inter, 26:69–101.

Borba, E. J.; Gasparini, I.; Lichtnow, D. 2017. Time-Aware Recommender Systems: A Systematic Mapping. HCI International, Springer. (to appear).

Brusilovsky, P. Methods and Techniques of Adaptive Hypermedia. 1998. Adaptive Hypertext and Hypermedia, Kluwer Academic, Publishers, p. 1-43.

Burke, R. D. 2002. Hybrid Recommender Systems: survey and experiments. UserModel UserAdapt Interact., v .12, n.4, p. 331–370.

Campos, P. G.; Díez, F.; Cantador, I. 2014. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. User Model User-Adapt Inter, vol. 24, pp. 67-119.

Chen, G.; Kotz, D. 2000. A Survey of Context-Aware Mobile Computing Research. Technical report.

Dey, A.K. 2001. Understanding and using context. Ubiquitous Comput, vol. 5, no. 1, pp. 4–7.

IEEE 2002. Draft Standard for Learning Object Metadata, Available at: http://grouper.ieee.org/groups/ltsc/wg12/20020612-Final-LOM-Draft.html..

Ferfernig, A.; Friederich, G.; Jannach, D.; Zanker, M. 2011. Developing Constraint-based Recommenders. In: Ricci, F. (Org.); Rokach, L. (Org.); Shapira, B. (Org.); Kantor, P. B. (Org.). Recommender System: Handbook. Springer.

Jannach, D.; Zanker, M.; Felfernig, A.; Friedrich, G. 2011. Recommender Systems: An Introduction. New York, USA: Cambridge University Press.

Koren, Y. 2009. The bellkor solution to the netflix grand prize. Netflix prize documentation.

Koren, Y; Bell, R. 2011. Advances in Collaborative Filtering. In: Ricci, F. (Org.); Rokach, L. (Org.); Shapira, B. (Org.); Kantor, P. B. (Org.). Recommender System: Handbook. Springer.

Lops, P.; Gemmis, M. de; Semeraro, G. 2011. Content-Based Recommender System: State of the Art and Trends. In: Ricci, F. (Org.); Rokach, L. (Org.); Shapira, B. (Org.); Kantor, P. B. (Org.). Recommender System: Handbook. Springer.

Merriam-Webster. Last access: 2016. Time. Available at: http://www.merriam-webster.com/dictionary/time,

Petersen, K.; Feldt, R.; Mujtaba, S.; Mattsson, M. 2008. Systematic Mapping Studies in Software Engineering. 12th International Conference on Evaluation and Assessment in Software Engineering, vol. 17, no. 1.

Ricci, F.; Rokach, L.; Shapira, B. 2011. Introduction to Recommender System Handbook. In: Ricci, F.; Rokach, L.; Shapira, B.; Kantor, P. B. Recommender Systems: Handbook. Springer.

Schmidt, A.; Beigl, M.; Gellersen, G. H. 1999. There is More to Context than Location. Computers and Graphics, vol. 23, no. 6, pp. 893-901.

Schilit, B.; Adams, N.; Want, R. 1994. Context-Aware Computing Applications. Proc. First Workshop Mobile Computing Systems and Applications (WMCSA ’94), pp. 85-90.

Verbert, K.; Manouselis, N.; Ochoa, X.; Wolpers, M.; Drachsler, H.; Bosnic, I.; Duval, E. 2012. Context-Aware Recommender Systems for Learning: A Survey and Future Challenges. IEEE Transactions on Learning Technologies, vol. 5, n. 4.

Zimmermann, A.; Lorenz, A.; Oppermann, R. 2007. An Operational Definition of Context. Proc. Sixth Int’l and Interdisciplinary Conf. Modeling and Using Context (CONTEXT ’07), pp. 558-571.