Recommendation Systems for Learning – Challenges and Solutions
Marie Sacksick

To cite this version:
Marie Sacksick. Recommendation Systems for Learning – Challenges and Solutions. AISR2017, May 2017, Paris, France. <hal-01640308>

HAL Id: hal-01640308
https://hal.archives-ouvertes.fr/hal-01640308
Submitted on 20 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Recommendation Systems for Learning – Challenges and Solutions

Marie Sacksick

Abstract— A recommender system is used to guide a user so as to find an item when there are too many of them. In case of learning, it aims at recommending the best learning content for the user given his parameters (which can be his objective, his habits, his emotions, his prior knowledge etc.). In this article, we highlight the differences between the challenges faced when constructing a “classical” recommender system and a personalized recommender system for learning, and how the solutions to these challenges must be different. This will be useful for people who want to build a learning recommender system, to foresee some challenges that are not treated, or not adequately, in roadmaps to build “classical” recommender systems.

Index Terms—Education, Recommendation System, Adaptive Learning, Learning.

I. INTRODUCTION

A recommender system (RS) is a system that allows to recommend interesting resources for the users, that can be used to help users to find out and select their information and knowledge sources [1]. RS are of help to find interesting items when there are too many for the user to know them all. Thus, it is used in many domains among which the most well-known are e-commerce websites and cultural consumption (such as Amazon or Netflix).

A personalized recommender systems (PRS) capture the traces that users leave in an environment. Such PRS can be used to the educational domain: the objects recommended are therefore pedagogical. In addition, they can be used for learning. In that case, PRS are more specific than those that are only recommendation systems to education, - such as recommending classes or recommending research articles -, since they focus on finding objects that will improve learning, i.e. the goal of the recommendations is to make the user learn and reach a pedagogical objective. Therefore, PRS for learning (called LPRS later in the article) requires personal learning data as well as “epistemic communities, that is groups of agents sharing common knowledge concerns” [2].

LPRS are useful for at least two reasons. The first one is that, as other domains, the amount of content one can access is growing exponentially: the drawback of such plenty is that users are not able to detect which contents are of interest according to their pedagogical objective and needs. The second reason is that the “one-size-fits-all” frame is not accepted anymore: people expect to learn with the best-suited content for them, avoiding boredom and withdrawal.

While building a LPRS, one will encounter several challenges that are specific, or that have a special twist, to/in the learning domain, which are not fully answered in the literature. The goal of the paper is to highlight those challenges and their specific side, to introduce some possible solutions and show the remaining blanks.

II. WHAT IS A RECOMMENDATION SYSTEM FOR LEARNING?

First of all, a recommendation system is a system which, thanks to a utility function $f$, is able to find the most useful object $o_{\text{max}}$ within the set of objects $O$ for any user $u$ of a given platform:

$$o_{\text{max}} = \arg\max_{o \in O} f(o, u)$$

When it comes to learning, this utility function will aim, among other criteria, at maximizing learning. This is why it could be used for Adaptive
Learning such as in [3], where feedback is recommended to users adaptively.

A personalized recommendation system for adaptive learning is composed of the following pieces:

- **User Model**: this represents what a user knows, what are his/her objectives, what he likes to learn, and so on, thanks to an aggregation of the traces left on the environment by the user. There are many technical choices to build a user model [4].
- **Domain Model**: this represents the structure of the objects on the platform. It can be almost absent with just little information on each object, or it can be very advanced and precise.
- **Recommendation algorithm**: that part is in charge of finding the next best learning object for the user taking into account the domain model and the user model.
- **Evaluation system**: the goal of that part is to understand the user’s actions to update the user model. For instance, one the user succeeded an exercise on the subject, the user model on that point is updated to mastery.

### III. THE CHALLENGES OF RECOMMENDATION SYSTEMS APPLIED TO THE LEARNING DOMAIN

The PRS can work thanks to personal data: the first obstacle is the users’ confidence concerning their data. Thanks to their data, it will be possible to organize them to construct a user model which describes the user. When the user arrives on the system, his/her user model is empty: the system must work despite a lack of data. Once everything is installed, the system will start making recommendation, but what is a good recommendation? Finally, after some use, it will be interesting to evaluate the whole system.

#### A. Personal Data

LPRS will come to use personal data, as other recommendation systems. Instead of memorizing that the user bought that book and like that kind of literature, it will memorize that the user completed that exercise and is at ease with grammar rules. The personal data protection problem is more significant here than in other domains for two reasons. The first is that it more likely than in other systems to deal with children. The second is that an analysis of the data can show how the user learns, how he/she forgets, what is known, and with didactic analysis, what are the reflexional process.

Users are aware of that sensibility and they are more and more demanding on the subject [5]. There are several solutions among which there are: informing well the public on data security, explaining the use made of the data, and who will access these data.

Moreover, as far as Europe is concerned, the European Commission made a new regulation on personal data to strengthen the legal framework around personal data and therefore reassuring consumers.

#### B. User Modeling

It is possible to make a recommendation system without user modeling (which does not mean no personal information), by mostly relying on content proximity: they are content-based. However, when it comes to learning, that method may fall short quickly because it means no learning goal. A more sophisticated system will require user modeling. This will lead to two new challenges: the first one deals with the learning profiles, which is a tricky subject in cognitive psychology; the second one deals with the fast evolution of the learner.

1) **Learning Styles**

The learning-style hypothesis consist in saying that, since people learn differently, then individualization instruction that takes into account the learning style will lead to better learning outcomes [6]. If that hypothesis were to be true, learning styles should be incorporated in the user model.

A study has been conducted in [7], to measure the effects on sixty students of receiving content adapted or not to their learning style, measured by the declarative Index of Learning Styles questionnaire. They showed that the students who learned through content matching their learning style were significantly more satisfied, though they did not show any effect on the learning outcomes.

In the state-of-the-art led by [6], no reliable study was able to show the impact of the learning styles on the learning outcomes. Several problems weaken
the learning styles hypothesis according to [8]: the declarative aspect to detect the learning style is not reliable, that style may depend on the subject studied, and finally even though every learner has its own way of learning, it seems laborious to make a short classification of theses.

Therefore, the learning styles should not be the principal piece of the user model in a LPRS.

2) Knowledge Evolution

In a classical context, the user modeling is quite stable: if the user likes rock music, it is highly likely that he/she will still like that kind of music after a couple of month, or after hearing a set of rock songs. Therefore, the user model must not be too sensitive to sudden changes.

In a learning case, the situation is completely different since the goal of a recommendation is to trigger an evolution of the learning profile towards more knowledge. Sudden changes are also important since they can show cheating, disengagement, or misconception of a concept. The challenge is therefore to be able to analyze quickly and subtly the interaction of the user with the environment.

This challenge has a particular impact on time computation of a recommendation, and on the interpretation that has to be given to the use following (or not) successfully (or not) the recommendation. Indeed, some systems are already hampered by time computation and cannot be used in real life because of the waiting time it would mean for the user [9].

Two solutions have been proposed by [10]: the first one is simply to have a temporal window, and the second, more efficient but more weighty, is to have two user model, one “recent” and one “old”.

C. The cold start problem

The cold-start problem is the fact that “new users start off with nothing in their profile and must train a profile from scratch” [11]. It can also concern newly added objects. This problem can be encountered in all recommendation systems which use a user modeling.

As far as objects are concerned, the problem can be settled quite rapidly: either the administrator does not mind that this object is not recommended (in the case of a content added by a user for instance), either a system of tag can be used to qualify the object, which allows the system to recognize a proximity with the user’s needs and/or tastes.

In the learning domain, the cold-start problem on user is slightly bigger because the user model is more complex to draw (it is about knowing everything the user knows and does not know!) and disengagement due to inappropriate content comes faster and may have more important consequences.

The classical tracks to solve are numerous, among them can be found: rule-based induction [12], implicit rating to enrich the user model faster [13], prior or self-assessment [14], and stereotyping.

Another help to solve the cold start problem in learning comes from the ontologies, knowledge spaces and/or knowledge graphs. These structures organize content and knowledge, and they allow to infer mastery of a concept thanks to prerequisite relationships. For instance, using knowledge spaces, the stochastic assessment algorithm [15] can describe the knowledge of a learner quickly and precisely.

D. What is a good recommendation?

Usually a good recommendation is defined as an object that the user will like, and that he does not know, as illustrated by Fig. 1 [16]. These two factors will contribute to the utility of the object, and the objects with the higher utility will be recommended to the user.

![Fig. 1. Classical classification of a recommendation](image-url)
Utility can contain other criteria: the system is then multi-criteria. For instance, another criterion could be the price in a classical system.

However, in a learning context, it is interesting to recommend what is called here “useless recommendation”, which are objects presumed too easy for the learner: indeed, this way the system can tackle misconceptions by presenting new ways of explaining a concept to the learner. On the other side of the difficulty, presenting objects presumed too difficult challenge the learner for him/her to go over his/her limits.[17]

That said, it becomes obvious that in a learning context, it is not possible to evaluate a lonely recommendation because it is part of a continuum which aims at teaching something to the user which might be far from what he/she knows. An average person would not choose on its own to go too far from his/her current knowledge, nor to go back on something he/she believes he/she masters, even though both situations are beneficial for the learner [18].

Therefore, to make an efficient learning recommendation system that is able to tackle the needs of the different learning phases, several utility functions must be designed to create various pools in which the system smartly digs when required.

E. The system evaluation

Finally, once all these challenges have been answered and the recommendation system is set, one can focus on the evaluation of the system so as to be compared with others. The question of system evaluation in a general case is still an ongoing subject [13], [19] and it is even less clear when it comes to education and learning.

1) What can be evaluated?

As a classical recommender system, a learning recommendation system can be evaluated in technical terms (accuracy, recall, coverage of items and execution time for instance), in terms of user-centric effects (do they feel an improvement of their user experience?), and compared to the goal [20].

2) How to evaluate?

It is possible to evaluate the technical aspect of a recommendation system off-line, with pre-registered data-set or a simulated data-set, or on-line, with users in a laboratory or in real life. The main advantage of the off-line method is that the evaluation can be done very fast, and has little cost. However, that method has been criticized a lot, such as in [21], because of its results which are not consistent with real-life experimentations. Furthermore, even if they were, it has been shown that with various data sets, the results of the same algorithm can be very different, and there is a lack of public and well-stock data sets which could be used by various systems. Therefore, off-line evaluation can be used when the system is still under development so as to gain time, but it is not reliable enough to be used to compare different algorithms between each others.

As far as the user-centric effects are concerned, it is mostly evaluated thanks to questionnaires, and sometimes interviews [20]. This evaluation has weight and meaning when the system has been tested in real-life conditions. Unfortunately, few systems are developed enough to be launched outside laboratories, and among those, few were tested through user’s opinion. When they passed all these barriers, they are not tested one the same points: [3] asked the users their opinion on the usefulness of the recommendations, while [22] oriented the evaluation to the benefits and the enjoyments. Furthermore, measuring user-centric effects is skating on thin ice because one must not mix evaluating the recommendation algorithm itself and its shell allowing to present recommendations to the learner. It shows that it is still required to structure the user-centric measures through a questionnaire for instance which would be very general. With such a method, algorithms could be clearly measured side by side.

Finally, the goal of a learning system being to make users learn, their progress should be evaluated. To know if the goal has been reached, one must “compute” the delta of users’ knowledge between before and after using the system. To do so, it requires knowing the state of the knowledge of the users at the beginning; however, setting a systematic placement test is too costly, and, for now, one cannot hope to find enough users who already have a user model in the learning domain. It would also require to have a unit for knowledge. To date, the best solution is to estimate, via an ending test, which number of learners successfully reached a desired level when they used the system compared to learners who did not use it. From this, it is possible to get the percentage of learners who succeeded and who might without the system (or
the opposite); and if the shell of the algorithm (most likely an e-learning platform) can measure time-on-task, the percentage of time same or wasted.

IV. CONCLUSION AND FURTHER WORK

In this article, we have underlined the common challenges which rise at the construction of a LPRS, and why they are different and cannot accept the solutions offered in “usual” recommender systems. We also presented some of the solutions that already exist and can be applied to that situation, and the points on which further work is needed.

In future work, we will present how our LPRS is designed, what are the solutions used to tackle the challenges pointed here. Furthermore, a proper evaluation of the system will be done, and a referential questionnaire to evaluate a LPRS will be proposed.

ACKNOWLEDGEMENTS

This work was supported by Charles Tijus from the laboratory CHArt (EA 4004) and Sebastián Ventura from the laboratory KDIS.

REFERENCES

