Learning Path Recommendation : A Sequential Decision Process

Zhao Zhang

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Learning Path Recommendation: A Sequential Decision Process

THÈSE

présentée et soutenue publiquement le 05 07 2022

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

ZHANG Zhao

Composition du jury

Président :	Davy Monticolo	
Rapporteurs :	Marie-Hélène Abel	Université Technologique de Compiègne
	Sylvie Calabretto	INSA Lyon
Examinateurs :	Nicolas Gutowski	Université d'Angers
	Davy Monticolo	Université de Lorraine
Directrices de thèse :	Armelle Brun	Université de Lorraine
	Anne Boyer	Université de Lorraine

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thesul.

Résumé

Au cours des deux dernières décennies, nous avons assisté à une adoption croissante du numérique dans le domaine de l'education. Cela est accompagné par un accroissement du nombre de ressources pédagogiques accessibles par les apprenants. Par conséquent, des systèmes de recommandation deviennent nécessaires pour aider les apprenants à trouver des ressources qui leur sont utiles. En particulier, cela inclut les systèmes de recommandation de parcours d'apprentissage qui visent par exemple à améliorer l'expérience d'apprentissage des apprenants, et notamment leur niveau de connaissance. Dans ce contexte, cette thèse se concentre sur le domaine des systèmes de recommandation de parcours d'apprentissage et sur l'évaluation de ces parcours d'apprentissage recommandés.

Cette thèse propose d'aborder la tâche de recommandation comme un problème de prise de décision séquentielle et considère les processus décisionnels de Markov partiellement observables comme une approche adéquate. Dans le domaine spécifique de l'éducation, la mémoire des apprenants est un facteur très important qui doit être pris en compte, et cela a été proposé dans la littérature et utilisé pour promouvoir des recommandations liées à de la révision. Cependant, peu de travaux ont été menés pour la recommandation basée sur des POMDP, et les modèles proposés sont complexes et requièrent beaucoup de données. Cette thèse propose deux modèles de recommandation basés sur POMDP qui considèrent la mémoire des apprenants, tout en limitant la complexité et le volume de données requis.

l'évaluation de la recommandation d'un parcours d'apprentissage est une tâche difficile de la littérature, qui peut être effectuée soit en ligne ou hors ligne. L'évaluation en ligne est très populaire, mais elle repose sur des recommandations effectives de parcours aux apprenants, ce qui peut avoir des conséquences dramatiques si les recommandations ne sont pas de qualité. L'évaluation hors ligne repose sur des ensembles de données statiques des activités d'apprentissage des apprenants et simule les recommandations de parcours d'apprentissage. Bien que plus facile à exécuter, il est difficile de procéder à une évaluation hors ligne de l'efficacité d'une recommandation de parcours d'apprentissage avec précision. Ceci tend à justifier le manque de travaux de la littérature sur ce sujet. Pour résoudre ce problème, cette thèse propose également des mesures d'évaluation hors ligne simples.

Enfin, ces algorithmes et mesures sont évaluées sur deux jeux de données réels. Nous avons montré que les algorithmes de recommandation proposés ont une qualité de recommandation supérieure à ceux de la littérature, avec une augmentation de la complexité limitée, y compris sur un jeu de données de taille moyenne. En ce qui concerne les mesures d'évaluation, nous avons montré qu'elles permettent effectivement de caractériser et de différencier les algorithmes de recommandation.

Abstract

Over the past couple of decades, there has been an increasing adoption of Internet technology in the e-learning domain, associated with the availability of an increasing number of educational resources. Effective systems are thus needed to help learners to find useful and adequate resources, among which recommender systems play an important role. In particular, learning path recommender systems, that recommend sequences of educational resources, are highly valuable to improve learners' learning experiences. Under this context, this PhD Thesis focuses on the field of learning path recommender systems and the associated offline evaluation of these systems.

This PhD Thesis views the learning path recommendation task as a sequential decision problem and considers the partially observable Markov decision process (POMDP) as an adequate approach.

In the field of education, the learners' memory strength is a very important factor and several models of learners' memory strength have been proposed in the literature and used to promote review in recommendations. However, little work has been conducted for POMDP-based recommendations, and the models proposed are complex and data-intensive. This PhD Thesis proposes POMDP-based recommendation models that manage learners' memory strength, while limiting the increase in complexity and data required.

Under the premise that recommending learners useful and effective learning paths is becoming more and more popular, the evaluation of the effectiveness these recommended learning paths is still a challenging task, that is not often addressed in the literature. Online evaluation is highly popular but it relies on the path recommendations to actual learners, which may have dramatic implications if the recommendations are not accurate. Offline evaluation relies on static datasets of learners' learning activities and simulates learning paths recommendations. Although easier to run, it is difficult to accurately evaluate the effectiveness of a learning path recommendation. This tends to justify the lack of literature on this topic. To tackle this issue, this PhD Thesis also proposes offline evaluation measures, that are designed to be simple to be used in most of the application cases.

The recommendation models and evaluation measures the we propose are evaluated on two real learning datasets. The experiments confirm that the recommendation models proposed outperform the models from the literature, with a limited increase in complexity, including for a medium-size dataset. In addition, the measures proposed actually allow to characterise and differentiate the algorithms.

Je dédie cette thèse
à ma famille.

Contents

List of Figures

1

Introduction

1.1 Context and Motivation

Over the past couple of decades, there has been an increasing adoption of Internet technology in our everyday lives. We shop online, search information by using search engines, spend an important part of our social life online and even learn online. Since almost all services, products, information, etc. become available to every web user, our choices are becoming nearly unlimited. As a result, the need for effective recommendations to help users to find useful content has emerged.

The task of recommender systems is to turn users' online activities and interactions into predictions of their possible future likes and useful interests to provide them with personalized recommendations. The huge potential of recommender systems was first noticed by Goldberg et al. in the forefront of the information revolution [1]. The research in Recommender Systems (RSs) is at crossroads of several disciplines such as computer science, sociology [2], economics [3], finance [4], education [5], etc. In addition, while being a field originally dominated by computer scientists, recommendation calls for contributions from various directions and is now also a topic of interest for mathematicians, physicists, and psychologists [6].

A large number of works in the literature focused on the design of recommender system, that vary according to the type of data used (ratings, traces of activities, resource description, etc.), the type of approach (deep learning, matrix factorisation, Bayesian networks), etc. Several classifications of RSs have been proposed according to these elements: memory-based/model-based, collaborative/content/hybrid, etc. [7]. In these classifications, we can identify the one that considers the time horizon of the intended *goal* of the recommendation: short-term goal and long-term goal [8, 9].

Short-term recommender systems (ST-RSs) provide a single recommendation. This recommendation is intended to be the user next action, and the short-term goal is generally the user satisfaction once the recommendation is adopted, the satisfaction is thus immediate (short-term goal). It should be mentioned that the impact of the recommendation on the user's future behavior, preferences or satisfaction is not considered at all.
For example, in e-commerce, when users shop online, they generally face many choices and are drowned in the volume of resources. However, they aim at buying a resource with the best value for money, w.r.t. their preferences. The recommender system recommends one resource that meets these preferences. Notice here that a set of recommendations can be provided, where each resource meets these preferences and the user may choose one or several resources in this set, generally one resource. In the context of online music, users are recommended a song or a set of

songs they might like, based on their previous preferences or interactions.
ST-RSs are the first RSs that have been proposed in the literature. They are widely studied in both research and industrial domains [1] and are still of high interest [10]. The term of short-term was first used in [11].

Long-term recommender systems (LT-RSs) aim at reaching a goal, which is not immediate but occurs later. This goal can generally not be satisfied in one step (one recommendation), but can be reached through a series of recommendations. Thus, LT-RSs aim at building a sequence of resources (recommendations) that contribute to achieve the predefined goal, which cannot be reached in one step and that can occur far in the future.
For example, when learners study online, they may receive a recommendation of a sequence of pedagogical resources designed to faster acquire more knowledge and reach a pedagogical goal. In the cultural heritage, one specific painting may be appreciated by a beginning visitor only if he/she has some prerequisites. To ensure that this visitor appreciates this painting, a sequence of recommendations has to be recommended which may start with a series of easy-to-access paintings. Then more complex paintings can be recommended and finally this specific painting. Notice that even though a complete sequence of resources is built by LT-RSs, this sequence can be shown to users either resource by resource, or the entire sequence at one time.

The sequence of recommendations is defined in this PhD Thesis as a *path* in the RSs domain, where:

Definition 1 *A recommended path is a personalized sequence of resources that is recommended to a target user to achieve a predefined goal, including a long-term goal [12, 13, 14].*

The literature has focused on LT-RSs more recently than on ST-RSs [15]. Besides, ST-RSs and LT-RSs are different in two perspectives:

- ST-RSs recommend resources that are generally only related to the profile of the target user (the history of his/her interactions, his/her preferences), and the objective function is generally simple, such as the maximisation of the probability that the target user will like the resource. LT-RSs also recommend resources that have to be related to the target user's profile, but have, in addition, to contribute to reach the predefined goal. Notice here that the goal may be distant in terms of content from the user profile.

- Building a path that contributes to reach a predefined goal is more complex than recommending a single resource or a set of resource(s). Indeed, the output is not a single element, but a sequence of elements, where the elements that constitute this path have to be coherent with each other, and this path has to be coherent with the user profile and gradually contribute to reach the goal. To make this path accurate, more information may be required: about the users, the resources, etc. [16].

LT-RSs are thus quite different than ST-RS, not only in terms of output, of goal, but they are also more complex. As an important and emerging topic in recommender systems, LT-RSs still have several challenges, including the complexity of the models and their evaluation.

Inline with the constant presence of Internet in our everyday life, it is also gradually being used in the education field. Online education platforms such as Learning Management Systems (LMSs), Massive Open Online Course (MOOC) and their associated tools, are now embedded in many teachers' and learners' activities [13]. This general adoption in education also faces one limit: learners are drowned in the large number of learning resources they can access [17].

This prevents them from focusing on the adequate learning resources, i.e., the resources that fit their background, which limits their learning experience. In addition, learners may face cognitive overloading and disorientation, especially when they have to balance limited available learning time and multiple learning resources. Besides, in traditional face to face teaching, teaching is personalized at the group level: every learner of a group is taught the same way. RSs are an opportunity to propose more personalized teaching to each learner, depending on his background, memory capacities and performance. Therefore, personalized learning resource recommendation for learners in e-learning environments becomes an important requirement [18].

From our point of view, one critical aspect in education is to arrange the learning activities so that the learners will be guided to the right resource in the right place at the right time. In this context, educational recommender systems [19] aim at selecting and recommending learners learning resources to increase their learning experience. Such recommender systems may rely on the target learner profile, that may be built based on the traces of interaction of this learner with the LMS [20, 21], his/her academic performance [22], etc.

Learning is a long-term activity. For example, it takes about ten years to become a doctor; learning a foreign language requires several years; learning cooking or gardening requires at least a few weeks, etc. In these activities, the learning goal is generally not immediate, but is a long-term goal. The associated RS aims thus at reaching this goal by generating and recommending a sequence of learning resources, a path. For example, this path can target the increase in knowledge of this learner, under a time constraint, the success at an exam, etc. In education, recommending a sequence of learning resources is referred to as a Learning Path Recommender System (LPRS) [23]. When learners adopt a recommended learning path, they may reach the expected long-term goal. For example, this path may allow a given learner to master new knowledge, it may even lead to career success, etc. [24, 25, 26]. A recommended learning path can be defined as follows:

Definition 2 *A Recommended Learning Path is a personalized sequence of learning resources, recommended to a target learner to help him/her to reach a predefined learning goal.*

To realise a LPRS is a high risk task. The sequential dimension of the recommendation makes the risk even higher, compared to traditional short-term educational recommender systems. Indeed, adopting a learning path may have serious consequences for the learner, if this learning path does not fit the learner profile or does not allow to reach the expected goal. For example, the consequence for college students may cause college student retention, even dropout from school. Therefore, achieving accurate LPRSs is a difficult task. As for path or sequence RS, LPRS may require a large amount of information about the learners to model more accurately the impact of the resources on the knowledge level, skills, motivation, learning preferences, etc. [27].
The literature has proposed a variety of learner models [28, 29], and educational ST-RSs [19, 30]. However, few works have been interested in LPRSs. In our view, learning path recommendation is not a new but less mature scientific research direction, while being central in education [12], [31], [18], [14].

A recommender system that exploits only the traces of interaction of users with the system has a partial view of the environment and problem. Even if additional information such as the context, description of the domain, etc. is available and used, the view of the situation remains incomplete and the system cannot be sure to provide the best recommendation, or take the correct decision. For example, the duration a learner is expecting to work the current day is not

known, knowing if a learner is still actively studying at a given moment is impossible (except if the camera is switched on), etc.
This lack of information results in uncertain information, such as the concentration level, motivation, level of knowledge, etc. Notice that lack of information and uncertainty are not specificities of online teaching, they are also limits of face to face teaching.
From our point of view, how to deal with partial information and uncertainty is an important concern in the educational domain.

Under the premise that the RS in the field of education is difficult to deal with, the memory capacities of the learners is also worth noting. Learners' memory capacities play an important role in the learning process [32]. There is a huge amount of evidence that our ability to learn/remain improves with review actions and decays with delay since last exposure: review makes the learners' knowledge more firmly grasped [33, 34, 35]. Reviewing plays a crucial role in the design of educational instruction, leading to a trade-off between teaching new material and reviewing what has already been taught [36]. Notice that learning new knowledge is usually based on old knowledge and requires time investment and significant cognitive abilities. Apart from that, hundreds of studies have demonstrated that properly structured review with the material over time produces superior long-term learning [22].

We found that in reality, most people choose learning strategies that promote short-term memory, which achieves the short-term goal in learning. In real education scenarios, most learners prefer to learn new knowledge rather than review. As early as 1967, Pimsleur noticed that there were almost no incentives to make learners review in textbooks or in the training of teachers and trainers [32]. Most people know that if one wants to learn something well, whether it is a set of facts, concepts, skills, or procedures, a single exposure is usually not enough to develop good long-term retention [35]. It is especially true that improving long-term memory and the sustainability of knowledge are key issues for any learning activities. However, few learners question the value of regularly reviewing previously acquired knowledge [37]. In additional, event hundreds of studies have demonstrated that properly structured review with the material over time produces superior long-term learning, few teacher or researcher actually put this strategy into practice to consolidate their knowledge in the satisfaction of long-term goal [22].

In cognitive psychology, a memory model based method was discovered more than 100 years ago [38]. Specifically, periodic knowledge review is called spaced repetition [39]. A well defined spaced repetition model allows learners to learn more, possibly in less time [37]. Spaced repetition is effective because it allows the use of psychological phenomena that contribute to learning and memory [39].

Although there is not much work in this area, based on the above facts, the role of memory ability in e-learning has attracted more and more attention of researchers in recent years. These works are mainly based on the following characteristics of memory capacity: (1) the forgetting curve shows that we can predict when a person will forget information; (2) the interval effect shows that when we predict forgetting Previous learning will have an exponential impact on the benefits of memory; (3) The test results represent the principle that testing yourself in the process will strengthen these benefits [39]. This is also one of our research focus.

In our point of view, LPRS is an emerging topic that still faces challenges, among which the management of the partiality and uncertainty of the information collected about learners, and that should also take into account learners' memory, with the objective of improving the

accuracy of the recommended learning path.

Moving more deeply with human-computer interaction, a good feed streaming recommender system should be able to contribute to user satisfaction, which can be concretized by a high click through rate, an acceptance of the recommendations proposed, etc. More generally, any recommender system, whether traditional (ST-RS) or path RS (LT-RS), cannot be designed without considering the evaluation of its performance. This is also true in education.

In line with traditional RS evaluation methodologies, the evaluation of the recommendation of a sequence or a learning path can be performed either online or offline [40]. Online evaluation is used in a live environment and focuses on the impact of the recommendations on the learners. Although highly informative, online evaluation is highly time consuming, requires the availability of real users and is often not completely reproducible.

At the opposite, offline evaluation focuses on the accuracy of the recommendations by relying on static datasets of learners' learning activities and simulates paths recommendations. It is thus less costly, which justifies its popularity, but provides a limited estimation of the accuracy of the recommendation model. Offline evaluation has been highly studied for traditional ST-RSs: MAE, RMSE, nDCG, click-through rate, precision, novelty, serendipity, etc. Although offline evaluation is easier to run, it is difficult to accurately evaluate offline the effectiveness of a learning path recommendation [41], [25]. Few evaluation measures and frameworks have been proposed for the evaluation of sequences of recommendations, especially for offline evaluation [40].

The offline evaluation of the effectiveness of a recommended learning path is thus a challenging task. From our point of view, it deserves more attention, especially as it is easy to implement and be can broadly used.

In summary, with the development of Internet, online education has gained more and more attention. How to achieve effective online education requires in-depth research on an emerging scientific research direction: LPRS. Most of the literature purely relies on statistical models, but we think that these models could be improved if they were designed to take into account the characteristics of learning, such as learners' memory. Besides, the evaluation of the LPRS is of the highest importance but still lacks of evaluation methods, especially offline.

This PhD Thesis highlights the correlations around LPRS from three perspectives: algorithm combination memory model, application with real world data, and new evaluation measures.

We formulate the problematic addressed in this PhD Thesis as two research questions:
Research Question 1 How to build a LPRS that considers educational characteristics?
This first RQ can be divided into two sub-questions:

1. As there is much lack of information and uncertainty on the characteristics of the online education-related data, the first sub RQ is "how to deal with the partial observation problem in a LPRS?"

2. As learning is a long-term process, that relies on memory, the second sub RQ is "how to deal with the trade-off between learning new knowledge and reviewing old knowledge?"

Research Question 2 How to define offline LPRS evaluation measures which can assess the effectiveness of a recommended LP and contribute to a more general adoption of LPRS?

1.2 Contribution

The main contributions of this PhD Thesis are the following:

Contribution 1 : Creation of a Learner's Memory-based POMDP Model for Learning Path Recommendation, that answers Research Question 1. A learning path is a sequence of learning activities performed by a learner. LPRSs are designed to help a target learner, achieving a given learning goal. In line with the literature, we propose to address the problem of LP recommendation as a decision problem (DP), more precisely as a sequential DP and consider Partially Observable Markov Decision Processing (POMDP) as an adequate approach. The originality of this work lies in the way the learners' memory is managed. We propose two models: the first one represents the learners' memory in the state definition, and the second one considers it in an external module that inserts resources in a learning path to form a learning path with some resource repetitions. To the best of our knowledge, there are no more than five works that integrated educational domain specificities into POMDP-based models, and we are the first to combine the spaced repetition and POMDP model into a new LPRS model.

The experiments conducted on two real-world datasets confirm that it is possible to manage learners' memory with a POMDP-based LPRS, while increasing the recommendation accuracy. In addition, managing it in an external module not only increases even more the accuracy, but is also a way to significantly decrease the complexity of the model.

Related Publication:

- **ZHAO ZHANG,** Armelle Brun, and Anne Boyer, " Hierarchical Partially Observable Markov Decision Processing Recommender System in E-learning for Long-term Goal". In 3rd Annual Learning learner Analytics Conference (LSAC), 2019.

Contribution 2 : Creation of New Measures for Offline Evaluation of Learning Path Recommender Systems Recommending learners useful and effective learning paths is highly valuable to improve their learning experience. The evaluation of the effectiveness of a recommended LP is a challenging task that can be performed either online or offline. A relatively small number of works in the literature have focused on offline evaluation of LPRS.

To tackle this issue, we propose a series of *general* and *simple* offline evaluation measures that can be used in combination. These methods vary according to the different learner profiles. Under the premise of satisfying the sequential characteristics of learning path, factors such as the self-organized learning path of the target learner (the learning path that learners used in reality) and the learning path of a defined mentor learner are taken into consideration. These measures are evaluated on two recommendation algorithms on a real learning dataset and we show that they actually allow to characterise and differentiate the algorithms.

Related Publication:

- **ZHAO ZHANG,** Armelle Brun, and Anne Boyer, "New Measures for Offline Evaluation of Learning Path Recommenders". In 15th European Conference on Technology Enhanced Learning, 2020, pp. 259-273.

1.3 Plan

This PhD Thesis is organized as follows:

- **Chapter 2** provides an analysis of the literature about RSs, path RSs, evaluation measures, as well as their specificities in the educational context. Further, the concepts and challenges in the research field of LPRS and the Markovian methods including POMDP employed, are presented.
- **Chapter 3** subsequently shows the work conducter on LPRS : two POMDP-based learning path recommendation models, that manage learners' memory. To limit the model complexity, and the associated volume of data required, we propose to model memory in two simple ways. First, memory is manged as an attribute of the states of the POMDP. Second, memory is managed in a post-hoc process.
- **Chapter 4** presents the difficulties of the evaluation of a learning path, especially in education learning path. After devising guidelines of new measures, we present the proposed measures and the associated experiments we propose to conduct.
- **Chapter 5** summarizes the contribution of the PhD Thesis and remaining issues. In addition, prospects for future applications and further research directions are also presented, as well as potential improvement plans.

2

Related Work

As pointed out in the introduction, this PhD Thesis aims at designing a LPRS, i.e., a RS that provides learners with sequences of learning resources, where a resource is a pedagogical material. Moreover, the RSs of interest here are not only goal-oriented, i.e., they aim to achieve a user predefined goal (in our case a pedagogical goal) but also aim at managing the uncertainty about the learners.

This chapter is structured around four different elements. It will first present RSs from a general point of view: their principle, the approaches of the literature. A specific focus on RSs proposed for the educational context will be made. Secondly, the research status of general path RS and LPRS will be presented. Thirdly, the way RSs are traditionally evaluated will also be investigated. The last section will present RS when they are viewed as decision processes, and will include a study of Markovian algorithms (Markovian Decision Process and Partially Observable Markovian Decision Process).

2.1 Recommender Systems

RSs appeared in the 90's [1]. RSs are intelligent systems, generally associated with numerical services, that provide personalized suggestions of resources (products, services, people, etc.) that users might be interested in [42]. As the choice of numerical services grows, so do the choices users have. As a consequence, RSs are becoming crucial tools and new approaches and techniques are constantly proposed [43, 44].

In this section, we will provide an overview of RSs technology, both in a general environment and for educational purposes.

Several classifications of RSs have been proposed in the literature [45, 46]: nature of the data, structure, temporality, application domain, etc. We propose to present different approaches according to the structure of data they exploit. We will first quickly present the approaches used when data is made up of user-provided ratings, then we will make a thorough focus on the case where data represent user behavior, which we call sequential RSs. Finally, we will consider works related to recommendations in education.

2.1.1 Rating-based Recommender Systems

Rating-based RSs have a long history, they are even the first type of RSs that have been proposed [1]. They are still widely used in many domains such as e-commerce, movie recommendation, etc., and are still of interest for researchers [47].

Collaborative Filtering Recommender System (CFRS) is the dedicated term for rating-based RSs. The task of a CFRS is to predict a target user's (tu) rating on each resource (item) he/she did not rate, based on other users' ratings on a set of resources. CFRS assumes that users who had similar interests in the past will still have similar interests in the future. CFRS then chooses which resources should be recommended to tu [48].

The set of user-resource ratings is represented as a rating matrix. Ratings are numerical values, from a predefined minimum value (for example 1) to a predefined maximum value (for example 5). In this case, the recommendation problem is formulated as a matrix completion problem [49]. Table 2.1 shows an example of rating matrix for a set of users $U = \{t_1, \cdots, t_m\}$ on a set of resources $Res = \{res_1, \cdots, res_n\}$. $rating_{ji}$ is the rating given by user j on resource i.

Table 2.1: A rating matrix R

R: Resource / User	res_1	res_2	...	res_i	...	res_n
u_1						
u_2						
...						
u_j				$rating_{ji}$		
...						
u_{m-1}						
u_m						

Two popular techniques have been proposed by the CFRS literature: memory-based and model-based RSs.

The memory-based CFRS directly exploits data and proceeds in two steps. First, for a given target user t, the ratings about resources that he/she did not rate are predicted by either identifying similar resources according to their associated ratings (item-based approach) [50] or by identifying similar users according to their ratings (user-based approach) [51]. Second, CFRS recommends a set of top k resources i.e., those with the highest estimated rating, where l is a parameter determined manually. In both these prediction and recommendation steps, the evaluation of the similarity plays an important role. It can be evaluated by different measures such as Jaccard Similarity, Cosine similarity, Pearson correlation, etc. [50, 51].

For example, in the prediction step of a user-based approach, the predicted rating of target user tu on i_{th} resource can be evaluated as presented in equation (2.1).

$$\hat{rating}_{tu,res_i} = \overline{rating_u} + \frac{\sum_{u=1}^{m}(rating_{u,res_i} - \overline{rating_u})w_{tu,u}}{\sum_{u=1}^{m} w_{tu,u}} \tag{2.1}$$

where $rating_{u,res_i}$ is the rating of user u on i_{th} resource. $w_{tu,u}$ is the weight associated with the pair tu, u and it is often the similarity between the target user tu and a user u. $\overline{rating_{tu}}$ is the average rating of target user tu. m is a predefined parameter that represents the number of neighbor users, i.e., the m most similar users.

The model-based approach first uses the rating matrix to compute a model, which is in turn used to estimate unrated resources. The great majority of recently proposed models are based on deep neural networks [52]. Once the models are built, they have proven their high performance and their ability to provide recommendations in real-time. The time-consuming part dedicated to the learning of the model is performed offline. The recommendation step is similar to one

of the memory-based approaches, the top k resources (with the highest estimated ratings) are recommended.

We would like to highlight the fact that hybrid recommendation algorithms are still popular [53], [54]. They combine multiple recommendation algorithms together to not only avoid the limits of individual algorithms such as cold start, but also propose more accurate recommendations to users, thus increasing their experience [48].

2.1.2 Sequential Recommender Systems

With the enrichment and generalization of the Internet, more and more services have broken away from the rating mechanism, to limit user solicitations. Thus, rating-based recommendation approaches are becoming gradually limited.

Sequential Recommender Systems (SRS) are designed to manage users' behavior, especially their interaction with resources. Getting behavioral data is fully transparent for the users, so it has no impact on their experience, and much data can be collected. Behavioral data are, in essence, sequential or temporal data, hence the term SRSs.

Notice that some works consider these interactions as implicit preferences and infer users' preferences by exploiting these interactions over time [55, 56]. The problem comes then down to a rating-based RS. An example of SRS, inspired from [15], is presented below. Jimmy is an online service user who regularly uses the referral service. These services correspond to his sequential behavior, and the traces of these service interactions are automatically collected. Jimmy first booked a flight, then a hotel, and finally a car. The question an SRS aims to answer is: what next interaction (action) should be recommended to Jimmy? To answer this question, SRS considers that actions in human behavior are all sequentially inter-related, especially each action is related, more precisely depending on the previous ones, so the recommended resources are defined based on the previous resources the target user interacted with. From the sequential behavior point of view, we can suppose that Jimmy is going on a self-driving tour and the next action that can be recommended is visiting a place such as a museum.

We propose to define a SRS as follows:

Definition 3 *A Sequential Recommender System is an RS that infers users' potential needs based on the explicit or implicit characteristics of their past behavior (history of their interactions). They recommend resources that may fit users' needs or that are in line with their behavior, by modeling the sequential dependencies embedded in their interaction sequence.*

The sequential recommendation task can be described as: given a set of all users U and a set of resource Res, for a target user $tu \in U$, tu has a historical sequence of interactions $S^{tu} =< res_1^{tu}, res_2^{tu}, res_3^{tu}, ..., res_n^{tu} >$. Given the set of sequential interactions for all users $u_i \in U$, $S = \{S^{u_1}, S^{u_2}, S^{u_3}, ..., S^{u_{|U|}}\}$, the objective is to recommend one resource or a set of resources to the target user tu that are in line with the sequence of his/her previous learnt resources.

The set of recommendations is generally made up of the top resources under the constraint of a utility function, for example the likelihood of resources [15], a conditional probability [57], a score [48], etc.

SRSs can be traced back to 20 years ago. We can for example cite the work that introduces SWARS, Sequential Web Access based Recommender System [58]. In this work, a pattern tree is used to store sequential patterns, then this tree structure is used to match the target users'

current history and generate recommendations. As an early research work, they found and studied the sequential problem, but as an initial work, the method is primitive and lagging as this method only used basic sequential pattern mining.

Since then, several models have been proposed, among which sequential pattern mining, Markovian approaches, factorization model, embedding models, deep neural networks, etc. [59, 15]. In the following, we will detail some of these models.

Markovian Approaches

Markovian algorithms are an important part in sequence-related problems, including Markov Chain, Markov Decision Process, Partially Observable Markov Decision Process [60, 61, 62]. The most simple Markovian algorithm is the Markov Chain (MC). Even though it has a long history and it is simple enough, it is still in the state of the art in the SRSs [63, 64, 65, 66].

MC have been first defined in [67] as following:

Definition 4 *A Markov chain or a Markov process is a stochastic model describing a sequence of possible events in which the probability of each event (event can be treated as consuming a resource) depends only on the state attained in the previous event.*

It can be presented as:

$$\begin{aligned} &Given\ Pr(X_1 = x_1, X_2 = x_2, ..., X_n = x_n) > 0, \\ Pr(X_{(n+1)} = x_{(n+1)} | X_1 = x_1, X_2 = x_2, ..., X_n = x_n) &= Pr(X_{(n+1)} = x | X_n = x_n) \end{aligned} \tag{2.2}$$

where the possible values of X_i form a countable set called the state space of the chain. The transition matrix describes the probabilities of particular transitions, and an initial state (or initial distribution) across the state space.

The definition 4 is 1-order MC. Besides, in *n-order* MC, the probability of each event depends only on the n previous states:

$$\begin{aligned} &Pr(X_{(m+1)} = x_{(m+1)} | X_1 = x_1, X_2 = x_2, ..., X_m = x_m) \\ &= Pr(X_{(m+1)} = x | X_{m-n} = x_{m-n+1}, ..., X_{m-} = x_{m-1}) \end{aligned} \tag{2.3}$$

where $n < m$.

Shani et al. [63] proposed to use a traditional MC process with an appropriate state space representation as to its prediction, thus recommendation model. Different to Definition (3), they use a 3-*order* MC with the transition probability function defined as follows:

$$\begin{aligned} Pr(< x_4 > | < x_1, x_2, x_3 >) &= Pr(< x_1, x_2, x_3, x_4 > | < x_1, x_2, x_3 >) \\ &= \frac{count(< x_1, x_2, x_3, x_4 >)}{count(< x_1, x_2, x_3 >)} \end{aligned} \tag{2.4}$$

where *count* represents the number of occurrences in all the sequence.

Rendle et al. [64] went a step further: starting from a personalized MC, they added a factorization model as a pre-posed process to improve their sequential recommendation model. In more detail, their model, called factorized personalized MC (FPMC) uses a combination of matrix decomposition and decomposed first-order MC as their recommender, which captures users' long-term preferences as well as resource-to-resource transitions.

In the context of e-commerce, a purchase *Basket* is made up of several resources and can be described as $B =< res_1, res_2, res_3, ... >$. The purpose of the recommendation is: based on

the previous purchasing history, which is the most adequate resource to be recommended for the next purchase?

In this context, the number of possible transitions between baskets would be $2^{|Res|} \times 2^{|Res|}$, where Res is the set of possible resources.

As other work presented, a basket B is defined such that absolute time points are not considered, only the sequential order is represented. Under this definition, a state represents a possible basket (a sequence of resources) and a 1-order MC is defined as:

$$Pr(B_t|B_{t-1}) \tag{2.5}$$

where t is the current timestamp. Given a previous purchased resource res_{pre}, for each resource in the next Basket, the transition probability of a resource res_i in the next purchasing basket can be defined as:

$$Pr(res_i \in B_t|B_{t-1}) = \frac{1}{|B_{t-1}|} \sum_{res_{pre} \in B_{t-1}} Pr(res_i \in B_t|res_{pre} \in B_{t-1}) \tag{2.6}$$

So the next Basket transition would be:

$$Pr(B_t|B_{t-1}) = \prod_{res_i \in B_t} Pr(res_i|B_{t-1}) \tag{2.7}$$

The modification is that they used a concept of Basket including resources, the Markovian chain linked the target Basket to its neighbor Basket and then add sequential resources into a target Basket.

To solve the data sparsity issue, which is a common difficulty in the RS domain, the authors used Tucker Decomposition in a cube A with three dimensions: based on the resource to resource transition matrix, they added a user dimension. The tensor of A then can be factorized as follows:

$$A := C \times V^u \times V^i \times V^n \tag{2.8}$$

where C is the core tensor of A, V^u is the features of users, V^i is the features of resource, and V^n is the features matrix of the next possible resources. Another Canonical Decomposition is added on A to further reduce the data sparsity. In this way, the pairwise interaction on the three tensors is computed, i.e. the relationship among user u, resource i, and resource n.

He et al. [60] also used a combination of MC and matrix factorization to solve the recommendation problem. First of all, the model learns an resource-resource similarity matrix W where W_{res_i,res_j} is the similarity of resource res_i to resource rec_j. The matrix W can be factorized as:

$$W = PQ^T \tag{2.9}$$

This factorization is explained as matrices P and Q, which are two low-ranked matrices. Mathematically, these two separate matrices will not have as much data sparsity as the original matrix W. Given the two matrices P and Q, the model proposed is defined as follows:

$$Pr_{tu}(res_j|res_i) = \overbrace{\frac{1}{|Res^+_{tu} - 1|} \sum_{res'_j \in Res^+_u / res_j} < P_{res'_j}, Q_{res_j} >}^{preference\ of\ user\ tu} + \overbrace{< M_{res_i}, N_{res_j} >}^{p(res_j|res_i)} \tag{2.10}$$

where $Pr_u(res_j|res_i)$ is the transition probability from the last consumed resource res_i to the resource res_j for the target user tu, Res^+_u is the user consumed resource set, vector M_{res_i} and

vector N_{res_j} come from matrices M and N, which are two matrices that factorized from a higher transition matrix of the user. Note that this form can change to a higher order MC.

Both [64, 60] use a factorization and MC based model to perform sequential recommendation. [60] use a higher order-MC which considers more sequential information in the model. Besides, [60] use factorization based on similarity matrix W rather than a simple transition matrix.

He et al. [66] also propose to use a higher-order MC and apply a novel transformation-based structure (transformation space) that bounds the previous resource, next resource, and the user together which solves the problem that traditional MC only has the relationship between resources.

Despite the popularity of Markov chain models, they suffer from two main drawbacks in the frame of RSs:

- The Markov property assumes that the current interaction depends only on the most recent interaction(s) (which is decided by the order of the Markov chain). Compared with other methods, although it is more suitable for dealing with sequential problems, it cannot really consider long-term goals.

- In general, they can only capture point-wise dependencies and ignore a part of the collective dependencies of user-item interactions, i.e., user is ignored generally ignored in the traditional MC, only non-personalized dependencies are modeled.

The details of MDP and POMDP, which are also Markov-based approaches, will be presented in the Section 2.4.

Deep Learning-based Approaches

These years, deep learning based methods are highly studied in many research areas, including in RS. For example, recurrent neural network (RNN) have been used to form sequential recommendations [68, 69]. To the best of our knowledge, Hidasi et al. [68] is the first work that uses RNN to model user sequential behavior, and achieves significant improvement in the recommendation accuracy. A few years later, Ying et al. [70] used a two-layer neural network to simultaneously consider the short and long-term preferences of users. As a neural network-based model, it requires huge data (a lot of user-system interactions), which limits their use to some specific applications. Besides, the RNN they propose has the drawback that it is a uni-direction deep neural network.

To meet the requirements of sequential recommendation, Zhu et al. [71] proposed a deep neural network, that exploits an attention mechanism to activate local historical behaviors for a given target resource, which can extract potential temporal interest from the user's historical behavior sequence and model the evolutionary process of user's interests. Their proposed model is called Deep Interest Evolution Network (DIEN) which is used for click-through-rate prediction through capturing users' interests as well as models the interest evolution. To achieve this, DIEN uses an interest extraction and an interest evolution layer. In the interest extraction layer, interest sequences are extracted based on behavioral sequences, and an auxiliary loss is calculated to improve the accuracy of interest expression; the interest evolution layer models the interest evolution process related to the target resource. Despite the good performance, this model could have been improved as the unidirectional structure still limits the ability of implicit representation in the sequence of user actions.

To cope with these drawbacks, Sun et al.[72] proposed a new sequential recommendation model which uses a deep bidirectional self-attention mechanism to model user behavior sequences.

Their model is composed by L bidirectional transformer layers. All of these layers are stacks. On each layer, their model iteratively corrects the representation of each position by exchanging information across all positions of the previous layer in parallel using the transformer layer.

Although deep learning has shown high prediction accuracy in practice, we highlight two main drawbacks. First, the time required for training the models is really high and requires huge computational resources. Second, it requires a large amount of training data as it has to learn a large number of parameters. Such a volume of data may not be available in some cases.

2.1.3 Repeated Recommender Systems

Repeated consumption is a natural behavior in our daily life activities: seeing a movie several times, revisiting a museum, rereading the news, repeating a web query, re-listening a song, etc. As a consequence, a user may appreciate the recommendation of resources that he/she already knows, has seen, or purchased in the past. Repetition is thus a critical factor that has to be taken into consideration by RSs. Nevertheless, the interest for repeated RSs has emerged recently.

On one side, repetition is inherently managed by models that purely model user behavior such as sequential patterns [73] or Markovian models [63] and deep learning-based models [72]. Indeed, if users tend to repeat an action (the purchase of a specific resource for example), the recommendation model, will represent such repetitions and recommendations can contain resources that the target user has already consumed or resources that the system has already recommended.

On the other side, repetition can be explicitly managed by the RS, with the goal control this repetition (what and when). For example, in [73] the repeated recommendation task is viewed as a survival analysis problem. A proportional hazard modeling method is applied in survival analysis, and a new opportunity model is proposed, which explicitly incorporates time into the recommendation system, to foster repetitions.

In some specific contexts, repetition may not occur in the data, it may not be a natural element, but be all the same important. For example, users may have forgotten some resources they have seen in the past, and may even be no more aware of them. The repetition can be used here as an intervention to help the users to remember these resources. The intervention is called a *reminder*. In [74], examples of such repetitions are presented: repetition can be a movie that users have seen years ago, news read the week before, etc. In these cases, the timing of the recommendation is of the highest importance to recommend a resource at a good moment, and at an appropriate temporal distance from the last moment the user has interacted with the resource.

[75] propose to model repeated recommendations in the frame of recurrent sequential user activities, by predicting the next returning-time of users to the resources. To do this, they capture the recurrent temporal patterns by connecting self-exciting point processes and low-rank models. [76] introduced Hawkes Process to model short-term and long-term temporal dynamics and provide accurate repetitions. In these works, the recommendations provided are a trade-off between novel and already consumed resources.

2.1.4 Recommender Systems in Education

As previously mentioned, RSs are used in many domains: tourism, commerce, health, etc. They are also more and more popular in education [77, 78]. The overall purpose of educational RSs is to provide learners with a list of educational resources around a topic, such as [79] who tried to find the right actions that the learner should do to learn primary mathematics.

However, education has specificities that make educational RSs different from traditional ones. In this section, we will focus on these specificities.

Considering rating-based approaches, learners with similar interests (or similar outcomes) may have different learning habits, different memory capacities, and different learning capacities, thus it is difficult for RSs to make effective recommendations based on traditional recommendation methods. More importantly, consuming a learning resource (studying a lecture, taking an exam, etc.) has an impact on the learner and on the entire learning process. This impact is not considered in traditional algorithms, such as in e-commerce, and we assume that they should be considered in educational RSs. One important question that is raised in the literature is: how to evaluate the impact of a learning resource on a learner or learning process?

The impact of resources on learning is directly reflected in the learner's knowledge level and the knowledge level is based on the performance of a learner. To model this impact, some researchers used the Item Response Theory(IRT) [24, 80, 25]. Concerning the learner profile, the literature confirms that it could be used to estimate learners' performance so IRT plays an important role in some educational RSs. The IRT model assumes that the probability that a question is answered correctly follows a logical or normal forget curve. In short, the IRT is based on the relationship between a learner's ability and correct response rate. The IRT can be presented as:

$$Pr(\theta_i, \beta_j) = c_i + \frac{1 - c_i}{1 + e^{-a_i(\theta_i - \beta_j)}} \tag{2.11}$$

where c_i is a guessing parameter that denotes the probability the learner randomly guess the right answer for resource i, a_i is the discrimination representing how steeply the rate of success of individuals varies with their ability for resource i, and β_j is the difficulty of the resource i, θ_i is the mater degree of the related knowledge for the resource. The θ_i here could be defined by expert such as a professor. Base on these we can have the recall probability for learner j of resource i as $Pr(\theta_i, \beta_j)$. Note that this IRT is called three parameter logistic model, when the c_i and a_i are considered as a constant, it changed to two parameter logistic model(2PL, also called Rasch model as presented before). The Rasch model [81] of IRT is defined as follows:

$$Pr(\theta_i, \beta_j) = \frac{e^{(\theta_i - \beta_j)}}{1 + e^{(\theta_i - \beta_j)}} \tag{2.12}$$

where the notations are the same as we presented in Equation 2.11. IRT has been widely used in e-learning. For example, [82] implemented a tutoring system based on deep learning that used IRT to recommend learning resources to learners. Recently, [25] used IRT to estimate a learner's learning efficiency.

Pliakos et al. [83] proposes a hybrid model that also associates Rasch model and a decision tree to improve the performance in education RSs.

Besides IRT, knowledge tracking [84] is another important technique for building adaptive education systems. It is also used to accurately predict learners' mastery of knowledge, and then do accurate pushing, path planning for learner learning or knowledge mapping construction based on sequences of learner behavior to predict the level of learner mastery of knowledge.

However, despite widespread use of IRT and knowledge tracing in e-learning, they are reassuring in terms of education for people who are afraid that AI replaces teachers: they require comprehensive data, such as the difficulty of the learning resources to estimate learner's answer ability, and this needs a lot of teacher's work.

Generally, RSs in education are sequential RSs. In educational RSs, there is often a goal used by the recommending process. This is even more true when the recommendation problem

is treated as a policy planning problem such as in [85, 86, 79, 87], etc. All the step of recommendation should consider all the next possible actions to reach this goal. The goal implicitly used in traditional sequential RSs is characterized by the maximization of the probability for the user's most interesting resources. In education, the goal may be explicit and may not be in line with the learner's profile or habits.

In this frame, the literature focuses on two main points: the definition of the learner model and the design of the recommendation technique. By way of illustration, [36, 29] are learner model based RSs and [31, 88] are recommendation technique based.

As learners cannot be treated in a uniform way [89] the literature has proposed special learners' models, such as in the work [90, 91, 29], etc. The most intriguing of these is the work of Choffin et al. [29]. They build a learner model dedicated to the learners' memory capacity, by using the learning and forgetting curve, which varies from one learning skill to another. Based on this memory model, the recommendations provided to a learner are more personalized, as it not only considers the characteristics of the target learner, but also includes different learning skills (through the memory modeling).

Since learning is a continuous and long run process with stronger prerequisites than general domains [24], it is important that RSs contribute to maintaining the continuity in the learning process. Based on this specificity of learning, some researchers focused on studying the knowledge model of learning resources through the relationship between courses, between knowledge points within courses, between resources, till the details of the relationship between questions, using topic model [88], knowledge tracing (we talked above) [84, 92, 91] cognitive structure [24], etc. The RS proposed by Samin et al. [88] works at the topic level. It shows that the use of a probabilistic topic model allows the generation of appropriate course and tutor recommendations. In detail, natural language processing techniques are used to find the features and relationships between the learners' study proposal and the faculty's resume. Since the granularity is large (topic level), we can consider that their method is close to the traditional RS. [92] propose to use a knowledge graph, that represents a large information set from different domains.

Liu et al. [24] use a prerequisite relations based graph to consider the cognitive structure and organize the order of recommendations. This kind of structure, including the knowledge level and knowledge structure, is specific to the field of education, it is less important and used in other domains such as e-commerce for example. On the other hand, they use a recurrent neural network to track learners' changing levels of knowledge at each learning step.

Chen et al.[91] propose to use deep knowledge tracing. They propose to incorporate information from the knowledge structure into the model, specifically by considering the back-and-forth relationship of incoming knowledge. Their core idea is that: given two knowledge points k_1, k_2 if k_1 is the antecedent of k_2, the mastery of k_2 is less than or equal to k_1, i.e., the mastery of the posterior is less than or equal to the mastery of the anterior.

Umemoto et al.[27] designed a RS to help learners upgrade their skills easily and more efficiently. It relies on skill improvement and resource difficulty modeling, used to choose action sequences. One specificity of this work is that they consider that the skill level of a learner never declines, which does not represent reality and may limit the performance of the model, especially when the learning process is long enough. Furthermore, they manage the inner construction of each learning resource which will be difficult to achieve in some experimental/real situations.

Since learners' learning is closely related to their memory, in the next section we highlight the current state of scientific research on some memory models in the field of education.

Human Memory Modeling

The literature about human memory modeling is highly active. The most popular memory model is the exponential forgetting curve. It has been over 100 years since it was first proposed by Ebbinghaus [38]. The forgetting curve, which represents the recall probability of a knowledge concept, is described as follows:

$$recall(res, tl) = e^{-\frac{time(t)\cdot\theta}{s(tl)}} \tag{2.13}$$

where $recall(res, tl)$ is the recall probability of a target learner for a resource, $time(t)$ is the time slot between the last time this resource has been learned and $s(tl)$ is the memory strength of the target learner.

Repetition is not a focus of traditional RS, it is also a research direction highlighted in the educational RSs literature and pedagogical works [22, 39, 33, 93]. Exploiting learners' memory is very important in deciding how to do the repetition as it can help the learner to retain knowledge and the ability to retain (new) pieces of information is an essential component in learning. The literature considers that a learning resource has to be repeated (or reviewed) when it (or resource associated concepts) is forgotten by a learner. This repetition is intended to ensure that the learner memorizes the resource (and/or its associated concepts) through time. Thus, repetition is related to both the learner's memory and knowledge state. The associated recommendation models are thus generally based on the learner's memory capacity [39].

In this frame, a significant amount of works focused on the highly popular spaced repetition approach, which aims at improving long-term retention by using a spaced repeated review of content. Spaced repetition relies on two elements: the probability of recalling a resource, which is related to the repeated exposure to this resource; and the delay, which is the time since this resource was last reviewed. Compared with previously mentioned methods, spaced repetition has a memory control dedicated to contributing to learners' learning more effectively.

Besides, repetition is also popular in many educational learning platforms and software, such as Duolingo[1], where spaced repetition is used to improve the learner's knowledge retention. In such platforms, resources that need to be reviewed are automatically popped up to consolidate learners' knowledge. We can note that researches that manage memory models require a great quantity of tests or quizzes [36], [33], [29] and are evaluated on data collected from such websites. Kang et al. [22] and Teninbaum et al. [39] described the spaced repetition from the pedagogical view: newly introduced and more difficult resources are shown more frequently, while older and less difficult resources are shown less frequently in order to exploit the psychological spacing effect. Spaced practice benefits memory, spaced practice improves generalization and transfer of learning, etc.

Tabibian et al. [93] use spaced repetition with stochastic differential equations with jumps. The reviewing process can be viewed as a Markovian process: each state can be renewed when a review action is done since when a review happens, the memory of a knowledge point is only related to the current state and it can be updated after each review action.

Settles et al. [33] designed an algorithm called half-life regression. It extends the exponential forgetting curve by discarding resource difficulty and increasing the exponential model of memory intensity (see Equation 2.14).

$$R(tl, res) = 2^{-\frac{\Delta}{h}} \tag{2.14}$$

where $R(tl, res)$ is the recall probability for a target learner on a resource, Δ is the time slot between the last time the resource has been learned, and h is the half-life or measure of strength

[1] https://www.duolingo.com

in the learner's long-term memory. When $\Delta = 0$, $R = 1$ which means that the resource has just been learned. When $\Delta = h$ the time slot is equal to the half-life, the resource is on the verge of being forgotten. When $\Delta \gg h$, the resource has not been reviewed for a long time relative to its half-life, so it has probably been forgotten. The half-life regression is an approach that relies on a complex loss function, which requires much training data. Compared with their method, Reddy et al. [36] used Poisson distribution to replace the half-life regression. As a consequence, it may not be utilizable on some mid-size platforms.

Reddy et al. [36] also proposed a queuing network model based on the Leitner system for reviewing flashcards (flashcards are card-boards consisting of a word, a sentence or a simple picture [94]), along with a first in first out (FIFO) structure that admits a tractable optimization problem for review scheduling, under the form of an exponential forgetting curve. It considers three elements: resource difficulty, the time elapsed since the resource was last seen by the learner, and learner's memory strength.

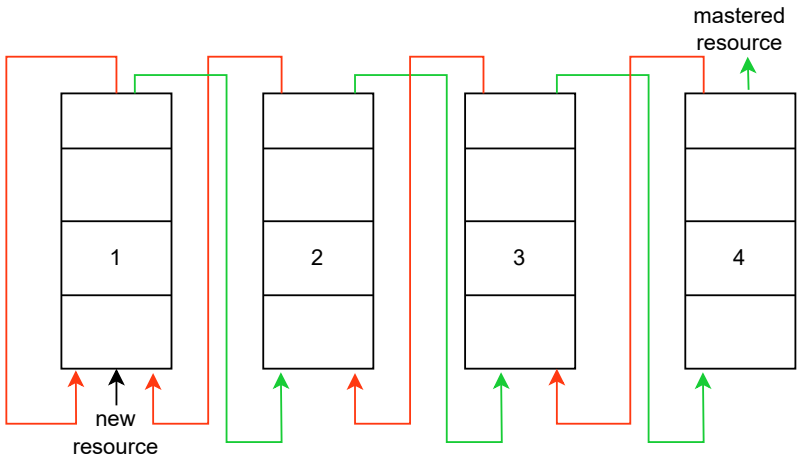


Figure 2.1: The Leitner Queue Network: Each queue represents a deck in the Leitner system. New resources enter the network at deck 1. Green arrows indicate transitions that occur when a resource is correctly recalled during the review, and red arrows indicate transitions for incorrectly recalled resources.

Figure 2.1 presents the structure of this model. In details, a deck (deck is an ordered set of flashcards). If one resource in deck k is memorized by the target user following Equation 2.15, the resource will be moved to deck $k+1$, otherwise, it will be moved to a previous deck following Equation 2.16. Given $\Delta_k = t - T_{k,1}$ is the delay (time slot of the resource last viewed) of the first learning resource in deck k, the transition probabilities of card moves are:

$$P(k \to k+1) = e^{\frac{-\theta\Delta_k}{k}} \tag{2.15}$$

$$P(k \to max\{k-1, 1\}) = 1 - e^{\frac{-\theta\Delta_k}{k}} \tag{2.16}$$

From our point of view, the main drawback of this model is related to the moving from one deck to another: the FIFO structure between desks is not flexible enough in the case of completely single-line processing. Indeed, there may be cases where the flashcards in the backward part of the FIFO sequence are forgotten faster. Note that retention is resource-dependent, not all resources

are forgotten at the same speed. We think that if the model compares several flashcards retention in the same deck during the same time, and then decides which one should be reviewed, it will be more efficient.

Choffin et al. [29] proposed the DAS3H model (resource Difficulty, learner Ability, Skill, and learner Skill practice History), that not only manages learners' memory but also their knowledge level about resources that can deal with several concepts. Note that it is exactly an adaptive spaced repetition model. This model requires resource description, as well as traces of learners' interactions, including evaluations. This model is a mixed blessing: it accurately estimates the learner's state in the learning process, but it requires much information about learners.

Quite recently, Burashnikova et al. [95] investigated the effect of long-term memory on the learn ability of SRSs containing implicit user feedback. Their work is based on two characteristics of SRS: 1. If user satisfaction does not change over a period of time, it will not change all the time, which is called *homogeneity*; 2. The current user's choice depends on the individual's entire interaction history, which is called *persistent*. Their work proved that the user's memory does make sense in several domains in RS.

In education, the literature on RSs uses various methods, but few of them are generally applicable [88]. Generally speaking, in each domain or furthermore in each different RS model, there is one and another with specific characteristics, so a customized RS generally consists of two parts: one or several core algorithms form the main part of RS, and one or several well-established complementary algorithms are used to complement the deficiencies in the core algorithm or in the dataset. Such as in [60] and [64], where MC is used as the core part associated with a factorized part to mine the resource-user relationship and to find a deeper resource-resource relationship. Besides, as a research point, human memory is also used by more and more researchers in personalized RSs. With the changing of the online technology, RSs are becoming more and more personalized and multifarious in very special domains.

2.2 Path Recommender Systems

2.2.1 Overview of Path Recommender Systems

As presented in the previous sections, RSs are traditionally designed to propose a user, referred to as the target user, one resource or a set of resources that fit his/her profile, preferences, etc. Most of these RSs thus have an immediate goal, i.e. they focus on the immediate value of the recommendations they provide.

With the generalization of RSs, users' needs are becoming more diverse. As a consequence, the methods traditionally proposed by RSs may not meet the broad spectrum of users' needs. In particular, users may not use RS for an immediate purpose only, but for a longer-term purpose. In such a case, a recommendation should not act in a single action, and should not take the form of the recommendation of a unique resource or a unique set of resources, but act in several steps, i.e. in sequence, and take the form of a sequence of recommendations, or a sequence of sets of recommendations. This sequence of recommendation is called a path, and the associated RS is called a Path Recommender System (PRS).

In PRS as in traditional sequential RSs, a user is usually known by his/her interactions with the resources over time: the sequence of his/her past interactions (see section 2.1.2). Let us recall that such RSs can rely on Markov models [55], recurrent neural networks [68], sequential patterns [15], etc.PRS does not necessarily differ by the way they exploit users' past interactions but differ by the way they form the recommendations. PRS can be divided into two approaches: step-by-step SRSs and entire path RSs, which will be detailed in the following sections.

Step-by-step sequential recommender systems

Let us first focus on step-by-step SRS. A step-by-step SRS aims to achieve a predefined goal, that is far from the present point in time. The path is built in multiple steps by relying on one resource from a set of candidates Res. For a SRS, the reward could be computed by Equation 2.17. It maximizes the reward for the next step by exploiting the learner's history interactions. The associated recommendations thus do not consider a long-term goal, only an immediate goal.

$$reward = arg\ max\ f(Res_c) \tag{2.17}$$

where the f is a utility function for a list of candidate resource Res_c [15]. Note that, it is just for one resource recommendation, it is like the traditional RS but with stronger and explicit sequence dependencies between the recommended resource and its history.

Such a SRS has the advantage of not being complex. However, the literature has shown that a step-by-step recommendation is cumbersome and inefficient, and the system usually becomes quite large and the resulting sequence may not be adequate [18, 25].

Entire path sequential recommender system

Let us now focus on the entire path RS, which builds the path in one run of the algorithm, not a sequence of run (which was the case in the previous paragraph). The literature, highlights two types of PRSs.

In such an approach, a path can be formed by running iteratively (in sequence) such a RS, where each run considers the recommendation of the previous step as the last resource visited by the user. A cumulative reward function is generally used, that calculates for each step the possible revenue in the overall path. In this context, the general reward for one recommended path with n resource can be defined as:

$$reward(path) = \sum_{i=1}^{n} \lambda^{i-1} reward(res_i) \tag{2.18}$$

where n is the length of the recommended path, λ is a discount factor and $reward(res_i)$ is the reward of each step. For a recommended path with a possible candidate set R, the number of choice at step i can be noted as $|R|^i$, i.e. it grows exponentially, the further back in the recommended path the greater the uncertainty. So in general, the discount factor is set to less than one.

Let us secondly focus on the continuation-oriented based RSs that view the path recommendation task as a simple *list continuation* recommendation problem [96]. Such a problem aims at recommending a sequence of resources that is the logical following of the most recently accessed sequence of resources [97].

Goal-oriented RSs are relatively rarely studied in the literature, and are an emerging topic. They aim at forming and recommending personalized sequences of resources that contribute to reaching a predefined goal [9]. When the goal is the traditional user satisfaction, the problem can be simply viewed as a prediction problem, in line with the works presented above. When the goal is different, the problem has to be viewed as a prescriptive problem. A double challenge arises in this case: how to define the utility function and how to form the sequence that allows maximizing this function? The utility function can not consider each resource of the sequence independently, but has to consider the sequence as a whole, as each resource has to contribute to the maximization of the function. For example, the utility function can be defined based on the

diversity of the elements of the sequence, the general popularity of the resources, the coherence within the sequence, etc. [96].

Such a RS is not goal-oriented even though it is path-based. Vall et al. [98] proposed a RS that recommends the continuation of music playlists, i.e. extends the current playlist. Two different methods are used:
1. Profile-based playlist continuation, which is a neural network based song-to-playlist classifier. The network analyses the song features, and uses it as input features to decide which playlist is suitable for the target user.
2. The membership-based playlist recommendation considers the continuation of the listening process.
More specifically, the continuation playlist is constructed based on the previous existing playlists. It relies on a feature matrix based deep neural network. Each playlist is represented as a matrix: each song x is presented as a vector and a playlist with length n is a matrix of dimension $n|x|$. A playlist p can be presented as X_p, where each line is a song. The model evaluates to what extent a given playlist and a given song fit together. Based on the given playlist p, the matched song s will be added to the recommended playlist p_r as a continuation playlist and the p_r will be added to p to extend the playlist.

At last, the goal-oriented based RSs. Such RSs recommend sequences of resources (path), formed so that they contribute to achieve a predefined goal, such as gaining knowledge in education, reach the destination in navigation software [99], plan a trip for a target user [100], etc.

Since this thesis focuses on goal-oriented path recommendation in the frame of education, I present below the state of the art of goal-oriented path RSs, more precisely goal-oriented learning paths RSs.

2.2.2 Learning Path Recommender Systems

In education, a sequence of learning resources is called a learning path (LP), and the associated RSs are called learning path recommender systems (LPRS). With the development of e-learning, the problem of LPRS i.e. recommending sequence(s) of learning resources has emerged. The literature highlights that these RSs are most of the time goal-oriented, not continuation-oriented systems [12]. They recommend sequences of learning resources (lectures, exercises, etc.) to help learners achieve particular learning goals [12]. Such RS can also be viewed as a proactive RS [74].

The main difference between the path recommendation and LP recommendation lies in the goal: in education the goal is generally defined (at least characterized), whereas in other domains, the goal is more often history-oriented, the problem is thus generally viewed as a list continuation problem. An example of a goal can be learner knowledge acquisition. As for path recommendation, LPRS use the learning history of a target learner, and each element of the recommended LP can be associated with a reward, evaluated by the Equation 2.17, and the whole reward for a recommended learning path is evaluated by Equation 2.18.

The specificity of education is related to the specificities of learners' profiles and the expected impact of recommendations. Indeed, the history of interactions of learners may have a mitigated importance. In the specific case of a learner who faces difficulties, the recommendations provided should not directly meet their profile (history), as we can suppose that the sequence of actions performed by the learner do not contribute to a high level of knowledge. On the opposite, for such learners, the recommended LP should contribute to reach the overall enhancement of the knowledge along the whole learning path [24]. The reward associated with Equation 2.18 should

be high (higher than for successful learners).

The definition of goal in a learning process is very important. It is most of the time described as the success at an exam or more generally an enhanced learning outcome [24].

One additional specificity of the learning context is the evaluation of the impact of a recommendation (if adopted) on the learner. Indeed, it can only be evaluated through tests or examinations, which are obviously limited. Thus, researchers use different strategies to perform this evaluation to help learners improve their performance.

Forming an LP that contributes to reaching a predefined goal is complex. Some works do all the same rely on continuation-oriented recommendation approaches. In our perspective, such approaches do not guarantee that the goal is reached. Indeed, these works first generate a set of possible paths and second select those that allow to reach the goal [80, 25, 18]. Nabizadeh et al. [80] applied a depth first search algorithm in a resource graph to find all possible paths for a given target learner, along with the associated performance. The path with the maximum performance for both time and score is then recommended. Based on the work of [80], Nabizadeh et al. in [25] went further: their model relied on a two-layered course graph (one for lessons, one for learning resources). The proposed algorithm processed in two steps. It first identified a path of lessons by depth first search to reach the objective by considering the knowledge background of a learner and then forms a path within each lesson. The path that maximizes the learner's score is then recommended. This approach is of high interest due to its low complexity.

In order to find an effective personalized LPRS, some researchers propose to use models to simulate the gains of a target learner who adopts a recommended LP. For example, Zhou et al. [18] proposed to generate personalized LPs, by using a long short-term memory model (LSTM), along with their associated performance. They face the well-known problem of the evaluation of learning efficiency which will be presented in section 2.3.3, and the estimation of both the learning effect of a learning resource and a path on the goal achievement. They use a function to calculate the personalized features of learners to cluster these learners in several groups. Their model is designed to recommend learning paths that allow the learner to make the most progress in the nearest estimated quiz. It is worth noting that they experimentally compare their approach with a next-step recommendation approach. Their conclusion is mitigated : if the learning path is short, the next step recommendation is more favorable. Once the LP is long enough, the goal-oriented approach is more accurate. The main limitation of the model proposed lies in the huge amount of data required to train the model.

Zhu et al. [26] introduced a multi-constraint LP recommendation algorithm with the goal to improve the efficiency of learning. Information about learners (logs, attention degree, behavior features), as well as information about resources (average learning duration, frequency, interval and knowledge map) are considered by the algorithm under the form of constraints. In their model, LPs are under the constraint of different learning scenario and resources have a prerequisite in the knowledge map. Notice that the extensive experiments conducted confirmed the intra- and inter-learner variability: a learner has different learning path requirements in different learning scenarios and different learners have similar learning path requirements in a same learning scenario, which confirms the need for the recommendation of personalized learning paths.

Xie et al. [14] studied the specific issue of identifying a suitable learning path for a group of learners. They propose a profile-based framework for the discovery of group learning paths.

Zhao et al. [101] managed the historical interactions of the learners and a pre-existing knowledge graph to generate a path. At the opposite of many works, this work has the advantage to not rely on any analysis of the content of the learning resources, which makes the approach applicable to many contexts.

Belacel et al. [102] proposed a graph-based approach, where the graph represents the prereq-

uisite relations between vertices. After eliminating the resources irrelevant to reach the goal, the shortest path is found in the reduced the search space, by using a branch-and-bound algorithm. In line with this work, Son et al. [103] also used a knowledge graph based model to recommend a learning path to the target learner in a MOOC context. In their opinion, it is not easy to use the traditional methods to deal with the education RS problem because knowledge dependencies and learner's preferences.

Sequential pattern mining (SPM) is a highly active domain but a less popular approach for LPRS [17]. Frequent sequential activities and associated sequential rules can be identified, possibly under some constraints: number of resources of a pattern, span time, utility of a pattern, etc. In the educational context, sequential patterns represent frequent learning activities, and learning paths can be recommended by combining and concatenating the patterns based on the user's current path and goal. The main advantage of this approach is that it does not require much information, except the traces of activities of the learners, and many SPM algorithms have been proposed in the literature that tackles memory and efficiency issues. For example, Su et al. [104] combined sequential pattern mining, clustering of learners, and a decision tree on a learner's profile to recommend an LP to learners, Hsieh et al. [105] used SPM to find the learners learning habits in courses pattern and then used formal concept analysis algorithm generates the recommended LP.

The literature also views sequence recommendation as a sequential planning problem. Markov Decision Processes, including Partially Observed Markov Decision Processes, are thus popular approaches for LP-RS. They will be presented in detail in Section 2.4.2.

An overview of these works' main idea is presented in Table 2.2.

2.3 Evaluation of Recommender Systems

As highlighted in the two previous sections, a large variety of recommendation approaches and techniques have been proposed in the literature, with various goals and in different contexts. The associated evaluation of these RSs is an actively discussed topic in the RS community. Generally speaking, there are three main approaches to evaluate a RS: user studies including user-dedicated questionnaires, online evaluation and offline evaluation [108, 109]. In this PhD Thesis we consider user studies as a kind of online evaluation.

Most of the evaluation measures used and proposed in the literature have been thought for rating-based or SRSs, few of them are designed to evaluate RSs that recommend sequences of resources, i.e., path or learning path RSs.

In this section, we focus on the evaluation of RSs including PRSs, for both the online and offline evaluation settings. We will start by studying traditional RSs and then we will focus on path RSs.

2.3.1 Traditional Evaluation Measures

In traditional RSs, systems aim at recommending users the next resource or set of resources, for different application contexts such as e-commerce, music, etc. For these different contexts, many standard classification and ranking metrics can be used for performance evaluation [110].

The main evaluation scenario is offline evaluation, which is much more popular in the RS community than online evaluation. Offline evaluation (simulation-based) conducts experiments on datasets made up of historical ratings or implicit feedback [111]. In this section, we introduce some common evaluation measure methods and then present examples of some applications in both offline and online cases in reality.

Paper	Years	Methods
A.H. Nabizadeh et al. [25]	2020	Based on [80], used a two-layers graph to find possible LPs.
A.H. Nabizadeh et al. [80]	2019	Use a pedagogical theory: item response theory to estimate the time and the score that a learner may get for all possible LPs.
Y. Zhou et al. [18]	2019	Use LSTM to predict LP
Q. Liu et al. [24]	2019	Use knowledge tracing to estimate the knowledge level of learners and build a Markov decision process based deep learning to generate LPs.
H. Zhu et al. [26]	2018	LPs are under the constraint of different learning scenario and resources have a prerequisite in the knowledge map.
H. Xie et al. [14]	2017	Use topic graph, knowledge and learning preference to recommend LP for groups of learners
S. Reddy et al. [85]	2017	Treat the LP recommendation as a policy planning problem. Used a partially observable Markovian decision process combined spaced repetition model-free method in a deep learning environment to plan a review considered policy.
Z. Zhang et al. [101]	2017	Used a hierarchic planning strategy model to recommend LP
F. Colace et al. [106]	2014	Define a new model based on [107] with knowledge space, user model, the observations model and the adaptation model.
M. Salehi et al. [17]	2014	Used pattern mining and collaborative filtering to recommend learning resources.
N. Belacel et al. [102]	2014	Used a strategy planning to recommend LP on a prerequisite based graph.

Table 2.2: A brief overview of LP-RS models

Haruna et al. [108] compared online and offline evaluation. They consider that the online evaluation is more convenient than offline evaluation because through the online evaluation, researchers are able to visualise the impact of recommendations and thus determine whether they are sufficiently effective. However, it has a high cost and a low percentage of practical cases can use such an evaluation. By way of illustration, in education, it is hard to use the online evaluation since the knowledge recommended to a learner is hard to be evaluated by a questionnaire. Although online evaluation is the most trusted experiment evaluation method, it is more time consuming than any other evaluation methods. On the other hand, offline evaluation is the simplest and most convenient evaluation method, which explains its popularity. The authors also elaborated that in terms of volume, the works that use offline evaluation are more than ten times larger than those using online evaluation. The popularity of offline evaluation was also confirmed by other researchers, such as in [40], etc.

So, in the following part we will give a brief introduction to the online evaluation and mainly focus on the offline evaluation.

Online Evaluation Measures

Since the online evaluation is costly and it is not our focus, we briefly present a few examples of how the literature proposed to realize online evaluation measures in real world settings.

Retention is a useful metric used for online evaluation of RSs [112, 113]. Online experiments can be performed as A/B testing. Users are divided into two groups, g_c is the control group and g_t is the test group. The measure evaluates the difference between the two groups. Retention is evaluated as follows:

$$retention = f(g_c) - f(g_t) \tag{2.19}$$

where the $f()$ is a function that evaluates users' performances, those in parameter, under the form of a numeric score. $f()$ is highly dependent on the application case and has to be redefined for each new case. A high value of retention means that the recommendations are more favored by users. To ensure a good recommendation, the evaluation should be a maximized retention.

Peska et al. [109] set an online A/B testing on a travel agency's production server. During the online evaluation, they monitor which objects were recommended to the user, whether she/he clicked on some of them and which objects she/he visited. They use two metrics: click-through rate (CTR) and rate of visit after recommendation (VRR) [2]. Both metrics are used in the utility function $f()$ in Equation 2.19. The combination of retention and CTR has also been used in [114, 115] in an online setting.

Focusing on the MovieLens dataset[3], Rossetti et al. [116] organized a user study with the aim of comparing offline and online evaluation metrics. They designed a series of questionnaires based on a series of movies as part of their online evaluation measures and used *precision* to compare the recommended results with the questionnaires. In their experiment, the questionnaires can be personalized. When a movie is recommended to the target user, if this recommendation is not in the rating history, the offline evaluation metrics will treat this recommendation as negative, but online evaluation can know if the user is interested in this movie and decide if this recommendation is positive or negative.

Beel et al. [117] also used online evaluation on the recommendation for a literature management software called Docear. They have more than $1,000$ users and about $50,000$ recommendations. To evaluate recommendations are favored by users or not, they used CRT and mean

[2] the value of VRR is traditionally lower than CTR because it represents the rate of users who finally visited after recommendation

[3] http://files.grouplens.org/datasets/movielens/ml-1m.zip

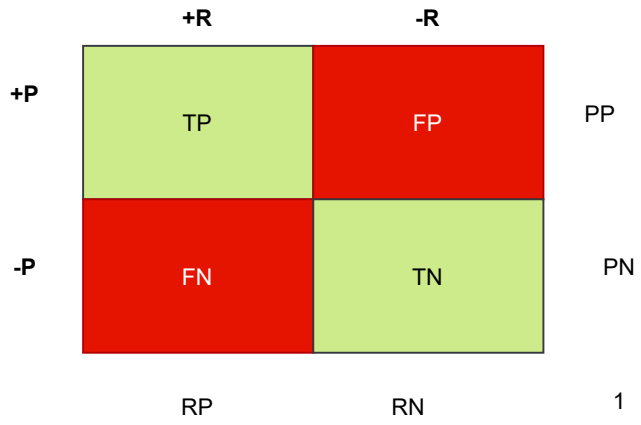


Figure 2.2: Example of a binary contingency table.

average precision as metrics. And in their experiments, the CRT and mean average precision results coincide highly. They claim that online evaluations are for different situations and are slightly less restrictive.

Offline Evaluation Based on Traditional Metrics

Here we present several traditional offline evaluation metrics that they can use in both rating based RS and SRS. The three most popular offline valuation metrics are the traditional precision, recall and F-measure [108, 118], which are measures used in many domains, such as information retrieval, speech recognition, etc.

Besides, other evaluation metrics, derived from the previous ones, are commonly used: average precision, mean absolute error, discounted cumulative gain, etc. In the following, we also focus on these popular metrics. Note that most of the traditional metrics are issued from the field of information retrieval, although the goal of RS and information retrieval is quite different. In information retrieval, the goal is to evaluate a query efficiency whereas in RSs the goal is to evaluate if the recommended resources are appreciated by users without any explicit query, or if they can have a positive impact on the users [118].

Here we divide all these measures into several groups according their characteristics.

Traditional Statistic Based Metrics In this part, we introduced four metrics: *recall precision*, F_n and root mean square error.

Recall and precision rely on a contingency table, as presented Figure 2.2. In this table, +P and -P represent the precision of positive (PP) and negative (PN) whereas +R and -R are recall of positive (RP) and negative(RN). In this table, at the cross of these elements we can find TP: true positive, FP: false positive, TN: true negative and FN: false negative. TP, FP, FN, TN and RP, RN and PP, PN refer to joint and marginal probabilities. The sum of the four contingent rate units and the two pairs of marginal probabilities is 1.

Precision and Recall

Precision, also known as a positive predictive value, is commonly used for classification, prediction and recommendation evaluation purposes [119, 110]. Precision represents the fraction of

recommendations that are relevant, i.e. the predicted positive cases that are actually positive:

$$Precision = TP/PP \tag{2.20}$$

As two very basic evaluation measures, precision and recall are still widly used ,such as in [119, 110, 52].

Recall, also known as sensitivity, is the fraction of relevant instances that were actually retrieved, i.e. the TP cases that are correctly predicted as positive:

$$Recall = TP/RP \tag{2.21}$$

In RSs, recall requires the definition of a parameter k to evaluate $recall@k$. $recall@k$ exploits the top k recommendations and computes the recall based on these k recommendations

As for precision recall is commonly used in RSs [120, 121, 52], etc.

Let us focus on a work that uses recall: Liu et al. [122] built a Markovian based context-aware recurrent neural network to model the user's behaviors and provide recommendations. In their experiments, they used two famous datasets: Taobao dataset[4] and MovieLens[5]. They used $recall@k$ as one of their metric, with $k = 1, 5, 10$. Note that there also exist other $@k$ metrics such as $precision@k$.

F-measure

The F-measure is calculated from the precision and the recall. F_1 is the harmonic mean of precision and recall.

$$F_1 = \frac{2}{recall^{-1} + precision^{-1}} = \frac{TP}{TP + 1/2(FP + FN)} \tag{2.22}$$

A more general formula can be represented as F_n, which applies additional weighting, that allows the weight of precision higher than recall.

$$F_n = (1 + n^2)\frac{recall \cdot precision}{n^2 \cdot precision + recall} \tag{2.23}$$

The F-measure is also commonly used in recommendation domains to finish evaluation [122].

Mean absolute error

The mean absolute error(MAE) is also a widely used evaluation metric. It could be used to evaluate a top N recommendation. It is presented as:

$$MAE = \frac{\sum_{i=1}^{n} |e(res_i)|}{n} \tag{2.24}$$

where the n is the number of recommendations and $|e(res_i)|$ is the absolute error comparing one recommendation with the standard resource.

Root mean square error

The Root Mean Square Error(RMSE) is defined as the square root of the expected value of the difference between the $\hat{rating}$ and $rating$, where the $\hat{rating}$ is the predicted recommendation position of the recommendation in an ordered user favorite list:

$$RMSE = \sqrt{E((rating - \hat{rating})^2)} \tag{2.25}$$

[4]https://tianchi.shuju.aliyun.com/competition/index.htm collected from https://www.taobao.com/

[5]http://files.grouplens.org/datasets/movielens/ml-1m.zip

When the recommendation contains Top N, it can be represented as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{1}^{n=1}(rating_i - \hat{rating_i})^2} \tag{2.26}$$

RMSE has been used in many works such as [123, 47, 124]. A new work of S. Rendle et al. [124] used RMSE as their unique evaluation metric. This leads to the conclusion that this method is still consistently applied as a popular metric.

Improved Recall and Precision Based Metrics The recall and precision based metrics are very traditional. When faced with evaluation of some different RSs, they need to make some adjustments. The following metrics were born.

Average precision

Precision and recall are single metrics that exploit the recommended resource for each recommendation case. For the specific case where a recommendation is a set of resources, each resource is associated with a rank in the set, it is adequate to consider this order in the evaluation measure [125]. In traditional recommendation domains, most RSs recommend an ordered set of resources that are likely to be of interest to the user as the next choice, e.g. in YouTube, Amazon, etc. Precisely, a precision-recall curve can be plotted to represent the precision. The Average Precision (AP) is the area under the exact precision-recall curve. This integral is replaced in practice by a finite sum over each position in the recommendation sorting sequence. Average Precision is defined as follow:

$$AvgP = \int_0^1 precision(recall(res_i))d\ recall(res_i) \tag{2.27}$$

where the $precision(recall(res_i))$ is the precision function of recall of each res_i.

Mean reciprocal Rank

The MRR is also a statistical evaluation of the results of a set of recommendations on the correctness of the ranking, it can be presented as:

$$MRR = \frac{1}{|U|}\sum_{i=1}^{|U|}\frac{1}{rank(res)} \tag{2.28}$$

where $|U|$ is the user's number of user set U, the $rank_i$ refers to the rank position of the first relevant recommendation for the $i-th$ user, the res is the top-ranked recommended resource. MRR has been used in [121, 126, 127, 128].

Mean average precision

Mean Average Precision (MAP) represents the mean of the average precision of a set of recommendations [129]. Note that in RS, *set* here represents a set of top-k recommended resources. Under these prerequisites, the MAP is defined as:

$$MAP = \frac{\sum_{resource_i}^{k} AvgP(resource_i)}{|k|} \tag{2.29}$$

where k is the number of the top-k recommendations, and $AvgP$ is the average precision. Since MAP is usually used for top-k recommendation [122] used $MAP@k$ just like $recall@k$ where $k = 1, 5, 10$ separately.

normalised Discounted cumulative gain

Normalised Discounted Cumulative Gain(nDCG) is also used in the situation that a Top-k set of

recommendations recommends to a target user, such as in e-commerce, movie recommendations, etc. [122, 120, 130, 47]. Some researchers use the DCG, a ranking-based measure: the more a resource appears at the top of the recommendation set, the higher the importance of this resource [131]. DCG measures the usefulness of a resource based on its position in the recommendation set. The gain is accumulated from the top of the result list to the bottom, with the gain of each result discounted at lower ranks:

$$DCG_{res} = \sum_{i=1}^{k} \frac{f(res_i)}{\log_2(i+1)} \tag{2.30}$$

where k is the size of the ordered top-k recommended set, $f(res_i)$ is the utility function of a resource presented in 2.17 that represents the relevance of the recommended resource at position i in the recommendation set. Notice that if different RS models propose sets of recommendation with different lengths, it makes no sense to compare the DCG values. The normalised CDG (nCDG) is designed to tackle this limit. In the premise of CDG, an ideal DCG (IDCG) is the score of $DCG = 1$ and it can be set by experts in very good order. nDCG is evaluated as follows:

$$nDCG_{res} = \frac{DCG}{IDCG} \tag{2.31}$$

MAP and DCG are two metrics with a similar idea: both of them exploit the rank of recommendations.

Metrics Migrated from Traditional Business The above evaluation metrics are all based on the methods generated by information science. Besides, some metrics are generated from traditional business evaluation standards. Here we introduce the two most commonly used methods: hit rate and click-through rate.

Hit rate

This measure is close to precise. For a group of users, if the RS recommends each user a resource, the hit rate can be defined as:

$$hit\_rate = \frac{|hits|}{|U|} \tag{2.32}$$

where the $|hit|$ is the total number of hit recommendation users set, and the $|U|$ is the number of all the users. Note that the hit rate is different from the precision: the precision should compute the k into both numerator and denominator. Generally, for a given list of Top-k recommendations, the user can only hit one resource in the list.In the work of Kang et al. [132], it defined a top N RS in the next step recommendation and used the hit rate as one of their metrics. If the hit rate is 1.0, it means that the performance of the RS recommended the right resource for each user whereas 0.0 means the worst performance none of the users hits the recommendations. This simple evaluation method is also popular in the RS literature [133, 134, 132, 134], etc.

Click through rate

The click through rate (CTR) is another metric close to precision. CTR can be described as:

$$CTR = \frac{|hits|}{|k|} \tag{2.33}$$

where the *hits* represents the number of user clicks in N recommendations. CTR is mostly used in classical application domains such as e-commerce, music and movie recommendations [135, 136, 134].

Innovative Evaluation Metrics Besides these traditional widely used metrics, there are also some other personalized metrics. D. Sacharidis et al. [47] proposed two metrics to evaluate the novelty and diversity for some users. For a set of recommendations Res for the target user tu, and the history of this user is noted as H_u, the individual diversity is the average pairwise distance between the recommended resources. And the distance function for two resource i, j of tu can be described as $d(i, j)$. The diversity of one user is then presented as:

$$individual\_diversity(tu) = \frac{1}{|Res|(|Res|-1)} \sum_{res_i \in Res} \sum_{res_j \in Res} d(res_i, res_j) \tag{2.34}$$

And the novelty of one user is:

$$individual\_novelty(tu) = \frac{1}{|Res||H_u|} \sum_{res_i \in R} \sum_{res_j \in H_u} d(res_i, res_j) \tag{2.35}$$

From Equation 2.34 and 2.35 it can be deduced that the novelty and diversity within the community can be expressed as:

$$community\_diversity(tu) = \frac{1}{|Res_c|(|Res_c|-1)} \sum_{res_i \in Res_c} \sum_{res_j \in Res_c} d(res_i, res_j) \tag{2.36}$$

$$individual\_diversity(tu) = \frac{1}{|R_c|H_c} \sum_{rec_i \in R_c} \sum_{rec_j \in H_c} d(res_i, res_j) \tag{2.37}$$

where the Res_c and H_c are the set of recommendations and set of histories for users in the same community.

Another example is the average reciprocal hit rank proposed by Kang et al.[132]. The average reciprocal hit rank(ARHR) is an upgraded version of the hit rate. It solves this problem by rewarding each hit according to where it appears in the top-k recommendation list, which is defined as follow:

$$ARHR = \frac{1}{|U|} \sum_{i}^{|hits|} \frac{1}{rank_i} \tag{2.38}$$

where the $rank_i$ is the position of the recommended in the ranked Top N list for the $i-th$ hit. In Equation 2.38, a resource occurs earlier in the recommended list of top k. Compared with hits rate, the ARHR metric evaluates the intensity degree of a resource recommended to the user through the RS. [137] also used this metric.

Although these traditional evaluation methods presented here are still in use, and they can only identify whether the recommendation is well-performing or not, but they cannot dig deeper into the reasons why the recommendation is well-performing. Such as [47], through RMSE, we can see the performance directly, but through the novelty and diversity, we can get further information about the trend of the changing of recommendations and find a reason why RS's performance is good or bad.

Further, the traditional metrics do not fit the specificities of path recommendations, including LP recommendations. In the next two sections, we will describe the dedicated measures used in path RSs evaluation.

2.3.2 Evaluation for Path Recommender Systems

In this section, we focus on the measures proposed by the literature to evaluate the accuracy of a path recommendation.

We would like to first notice that the evaluation of PRSs can be performed through traditional measures such as those described in Section 2.3.1. So, we will first present these traditional measures and the way they are used in the specific frame of path evaluation. Second, we will present measures specifically designed for path RSs.

Traditional evaluation for path RSs

Most metrics evaluate path RSs as a top-k recommendation problem. As far as we can see, in this case, the metrics are very rigid: they cannot distinguish between good and bad sequences. For example, if a sub-path in the recommended path turns out to be right, but has a slight shift in position in the recommendation, traditional metrics would consider this recommendation to have a bad performance such as the work of Ziegler et al. [138]. In our view, there is a vague intersection between the development of PRSs and traditional RSs. Starting with music playlists, these two different recommendations are gradually differentiated.

In music recommendation, D. Jannach et al.[97] and A. Vall et al.[98] both recommend personalized music playlists. [97] evaluate the recommended playlist, in an offline setting, according to three traditional dimensions: accuracy, homogeneity and coherence with the history. For evaluating the accuracy, they used four-fold cross-validation to divide the dataset and the metrics used are hit rate and MRR. They did not distinguish the path and top-k set recommendation. They added the homogeneity and coherence, which are evaluated by the information that is associated with each music, such as artists and the social tags changed following time changing, the taste-trend of the user changed or not in a music list. In this work, they purposely not modified the traditional evaluation measures, but have proposed to use domain-related information to complete the evaluation of the accuracy of the recommended path. However, such a proposition is limited: additional information per resource may not be available in each context.

Vall et al. [98] conducted an offline evaluation to evaluate the performance of their recommendations. The proposed evaluation metrics are based on the work of [97], but it is still a retrieval-based traditional metric. They used $recall@k$ and $MRR@k$ as their metrics.

Besides in the recommendation of the music list, Y. He et al.[96] aim to recommend a list of personalized resources (a path) by considering the user's history. They used two valuation metrics: nDCG and hit rate. In their approach, they consider the resources that should match the pre-defined standard list and then evaluate the recommendation with the two metrics. From our point of view, this means that they did not consider to evaluate the dependency within a recommended list, i.e., they only evaluate path recommendation as top N recommendation. Their experiments also illustrate the lack of evaluation methods for path RS.

As introduced at the beginning of this section, the measures presented above are not specifically designed for path evaluation, nor for the educational domain, they are traditional metrics slightly modified to fit path recommendation.

Innovative Evaluation Methods for Path RSs

Some works have focused on the design of metrics specifically designed for path recommendations. Notice that they are all very recent works.

After having summarized the strengths and weaknesses of various traditional evaluation methods and analyzed the property of path recommendations, M. Diego et al. [40] proposed several

offline evaluation metrics. Before talking about the details of metrics, we defined some notation in the Table 4.1.

Notation	Description
res	a resource
Res	set of all the candidate sequences that formed by resources
$\|Res\|$	cardinality of the Res set
U	user set
$\|U\|$	cardinality of the user set
u_i	the i_th user in the user set
$rpath(u_i)$	recommended path for user u_i
RP	set of recommended path
RP_{test}	set of original path in test set
$path$	standard path to be compared with

Table 2.3: List of the main notations used throughout the chapter

Concretely, they proposed eight evaluation metrics for path recommendations:

1. Coverage metric represents the rate of resources that the recommended path RSs covers. For example, in e-commerce background, it is the ratio of products purchased by users in the recommendation to the total products in a fixed period. The coverage metric can be defined as:

$$coverage = \frac{1}{|U|} \sum_{i=1}^{|U|} \frac{|rpath(u_i)|}{|Res|} \tag{2.39}$$

   These metrics simply computed a coverage rate. It is similar compared with CTR, HR, and precision: simple to be used but could not very well evaluate the inner relationship of a recommended path. Its application scenarios are also relatively limited: in our opinion, the most suitable ones are in e-commerce, music, and movie RS domains.

2. A modified precision metric:

$$precision = \frac{1}{|RP_{test}|} \sum_{path \in RP_{test}} \frac{hit(path, rpath)}{min(|path|, |rpath|)} \tag{2.40}$$

   where $rpath$ is the recommended path, $path$ is the standard/original path to be compared with, $|path|$ and $|rpath|$ are the length of the recommended path and original path respectively, the hit is a function that returns the number of the resource in $rpath$ that are also part of $path$. The number of related resources is divided by the minimum value of the $|path|$ and $|rpath|$.

   This metric is very similar to the precision defined in Equation 2.20. This method is simple and useful and it is proposed to penalize a RS recommends with a too big length.

3. The Normalized Distance-based Performance Metric (NDPM) is an adaptation of nDCG. It changes the importance of measuring resources in nDCG. At the opposite of many other metrics, NDPM assumes that in a recommended path first resources are not always more important than following resources. It is defined as:

$$NDPM = \frac{1}{|RP|} \sum_{path \in RP} \frac{f^{-}(path, rpath) + f^{u}(path, \bar{rpath})}{2f(rpath)} \tag{2.41}$$

where three count functions are used f^-, f^u *and* f: t f^- (the $^-$ here means the opposite) returns the number of pairs in the recommended path that are in the opposite order with respect to the standard path *path*; f^u returns the number of pairs in the recommended path *rpath* for which the ordering is irrelevant; f counts the number of possible pairs in a recommended path. This metric measures the opposite of the performance of a recommended path and is inspired by the work of Yao et al. [139]. For the result of this metric, the smaller the value, the better the recommended path.

4. The path diversity can also be used as a metric, and is not new. Di et al. [140] proposed that in a path, a higher different rate of resources in a recommended path, probably it is good for the user, because users are exposed to more new resources that also meet the requirements, i.e., for two similar *rpath*, the one with higher diversity may be a better choice. The proposed diversity metric is defined as follow:

$$diversity = \frac{1}{|RP|} \sum_{rpath \in RP} \frac{\sum_{\forall res_i, \forall res_j : 0<i<j}^{k} 1 - sim(res_i, res_j)}{|rpath| \times (|rpath| - 1)} \tag{2.42}$$

where res is a resource in the recommended path, $sim()$ is a similarity function between two resources, for example it could be cosine similarity. In a recommended path with length $|rpath|$, if we combine resources into pairs, there exists $|rpath|(|rpath| - 1)$ possible pairs. The diversity metric computed all the diversities (the opposite of similarity) for all resources pairs in a recommended path.

In this diversity metric, high values represent a high diversity. In our point of view, this metric would be more appropriate in some areas, such as education, where RS can help learners learn new knowledge.

Even Di et al. [140] discussed that the diversity would be a good metric, they only computed the similarity. Here it changed the similarity to diversity which explicitly shows the indicator of diversity.

5. Novelty can also be used to measure the path recommendation:

$$novelty = \frac{-1}{|RP| \times |rpath|} \sum_{rpath \in RP} \sum_{i=1}^{|rpath|} \log_2 f(res_i) \tag{2.43}$$

where the function $f()$ returns the normalized frequency of i_{th} resource in *rpath*. It evaluates whether a resource r available in the suggested paths is too frequent or not.

If diversity is to measure the degree of difference between different resources, then this metric is to measure the quality of a sequence from the frequency of occurrence of the same resource.

6. Serendipity metric measures the number of resources recommended meanwhile these resources are attractive and unexpected. In some cases, if the RS has no serendipity at all, some very useful resource will never be recommended to the user. This metric is based on the *spath*, which is also a path generated under the same background as *rpath*. The only difference is that *spath* includes very strong characteristics, such as it could be a sequence that which all the resources are the most popular. The proposed metric can be presented

as follow:

$$serendipity = \frac{1}{|RP|} \sum_{rpath \in RP} \frac{f(path, rpath - spath)}{min(|path|, |rpath|)} \tag{2.44}$$

where *hit* is defined as in Equation 2.40, $rpath - spath$ are the resources that appeared in the *rpath* but not in *spath*, the function $f()$ computes all the resources that matched the original path and not in the most popular path. This metric evaluates the recommended resources without very strong characteristics. The results of serendipity should be less or equal to the precision. The difference between precision and serendipity represents the percentage of obvious resources suggested correctly.

7. The confidence metric reflects how much the system trusts its own suggestions:

$$confidence = \frac{1}{|RP| \times |rpath|} \sum_{rpath \in RP} \sum_{res_i \in rpath} Pr(res_i | res_{i-1}, res_{i-2}, ...) \tag{2.45}$$

where the function $Pr(a|b)$ is conditional probability of a, knowing b. For a given recommended path *rpath*, this metric represents the average the probability that each resource appears with the resources before it. The bigger $confidence$ result is, the more the recommended resource should appear in *path*. This metric evaluates whether the recommended path is the system trusted or not. But the user's preference for this sequence is not necessarily the same.

8. The perplexity metric is a exponential in base on the cross-entropy of a RS model:

$$perplexity = 2^{\frac{-1}{\sum_{rpath \in RP} |rpath|} \sum_{rpath} \in RP \sum_{i=0}^{|rpath|} \log_2 Pr(res_i | res_{i-1}, res_{i-2}, ...)} \tag{2.46}$$

This metric is based on the cross-entropy which gives the information perplexity of a path, so the higher of the *perplexity*, the worse. Note that this metric is widely used in the evaluation of natural language processing. The perplexity is a traditional metric to evaluate a generated phrase in the natural language processing domain. As a phrase is generated from one seed word, it could be treated as similar to a path recommendation. So M. Diego et al. proposed to use this metric in the path evaluation domain. It is useful for cross-entropy loss function based models, the perplexity can be used as an indicator of model convergence. But for the other sequential RS, it may be useless.

These measures, proposed by Monti et al. [40], are all updates of measures previously proposed in other work, they make adjustments to accommodate sequential characteristics of a path. Furthermore, these 8 metrics consider path measurement from various perspectives and can be applied in various domains, such as natural language processing, e-commerce, etc. To the best of our knowledge, this is the first comprehensive paper on how to measure the path recommendation problem.

With the development of path RSs, traditional evaluation methods are also becoming increasingly unsuitable for specific situations [47, 40]. During the same time, the path RS evaluation is being studied by researchers. We believe that the existing evaluation methods are inadequate and need to continue to come up with some new methods.

For all these metrics, whether traditional or innovative, they all have the main drawback that most of them do consider the resources of the recommended path as independent of each other, and their order in the estimated path is not considered. Yet, it is a fundamental dimension in path recommendations.

2.3.3 Evaluation of Learning Path Recommender Systems

Online Evaluation of LP Recommendations

Recall that online evaluation aims at quantifying the actual effect of the recommendation on the users, here of the recommended LP on the learners' learning experience or state. In the literature, such an evaluation exploits the learners' level of knowledge through A/B testing of two groups of learners [25]. For this kind of online evaluation, the literature generally chose to focus on the level of knowledge to represent the learning experience, because this is one of the most attributes in the learning process [13, 29, 18]. In our point of view, it is easier to evaluate if a learner has improved his/her knowledge for an LP or not through the level of knowledge. And the evaluation of the level of knowledge mainly exploits results at exams, tests, quizzes, etc., which can evaluate the impact of the recommended/adopted LP through time such as we presented in Section 2.1.3 for tradition education RS. Note that commitment, motivation, etc. are rarely considered [13]. Besides, questionnaires investigating learners' satisfaction with the recommended LPs also exist [26]. Due to the scarcity of online evaluation, we only give a few examples below in learner's progress degree and learner's satisfaction degree.

A.H. Nabizadeh et al. [25] also adopted online A/B testing to evaluate the learners' progress degree through recommended LP. Learners are divided into two groups: one group gets personalized LP recommendations, while the other group gets non-personalized LP recommendations. After getting recommendations, the learners in each group take the final exam. The average score in each group is computed, and both scores are compared. The main advantage of this evaluation approach is that it is easy to implement and the evaluation criterion is very simple and easy to collect: the final exam score. Indeed, an exam has to be taken by learners even when no specific evaluation of the recommendations is conducted. In addition, no additional element is required, the evaluation measure is easy to be conducted. However, we still highlight several problems: 1. in online evaluation involving human subjects, deciding the sample size and experiment duration is not that simple; 2. it is difficult to find learners to be divided into two groups and the groups have similar knowledge levels compared with each other; 3. there may exist bias between the two tested groups: a test sample that is not large enough is hard to get a fully accurate result from one test.

Xie et al. [14] went a step further in the way to evaluate the impact of an LP. Not only the learners' performance is compared between groups, but the increase in performance (difference between the performance before and after adopting an LP) is also considered. Concretely, a single test is used, which is taken twice by each learner: a pre-test score is associated with the first time the test is taken and a post-test score is associated with the second time the test is taken (after adopting a personalized LP). The learners are randomly divided into three groups: those who follow a self-decided LP, those who follow a non personalized LP, and those who follow a personalized recommended LP. Self-decided LPs are determined by learners themselves (obtained by questionnaires for example) as proposed in [26], to evaluate learners' satisfaction degree of the recommended LP. The evaluation conducted here is more fine grained than in [25]. Indeed, the impact of the recommended LP is not only evaluated learner by learner, but also compared to the impact of other types of adopted LP (self-decided and non personalized).

Another part of the literature proposed to rely on the evaluation of the distance between the recommended LP and the ground-truth LP. Concretely, the well-known Edit Distance (ED) is the most commonly used measure [26]. Traditionally, ED is used to quantify how dissimilar two LPs are [13]. ED simply counts the minimum number of operations required to transform one of the sequences into the other. In the work of Zhu et al. [26], ground-truth LPs are determined

by asking the learners self-organized LP (through questionnaires), which is the only interaction of learners known by the system. One main limit in this work is that to estimate if the self determined path is correct or not is very difficult.

Although online evaluation is known to be accurate, we can see from these works that conducting such an evaluation is costly in the educational context (as well as in any context), mainly in terms of the time required to perform the evaluation and stakeholders' availability (including learners). We identify some general limits for online evaluation experiments: 1. if the quality of the recommendation model is not high, the recommendations proposed to the learners may negatively impact their learning experience; 2. the online evaluation process cannot guarantee equity between learners, especially when learners are split into groups and get different types of recommendations; 3. the division of the test combination is difficult to guarantee the single-variable method, that is to say, the learning background of each group of learners is almost the same. 4. online experiments are hard to be reproduced.

These limits are not only valid for LP, but for any online experiment in the educational context. These limits are the main reasons why offline evaluation is also a popular evaluation approach.

Offline Evaluation of LP Recommendations

Recall that offline evaluation relies on predefined static datasets. In education, these datasets are mainly made up of traces of learners' activities, i.e. learners' LPs.

At the opposite of online evaluation, no real LP recommendations and no associated test can be actually proposed to learners. So, as in traditional offline evaluation, to evaluate the accuracy of LP recommendations, a ground-truth LP is required. This ground-truth can be either inferred from the data, or obtained from experts, even estimated from external knowledge [141, 116]. Inferring ground-truth from the data is the most simple technique as it requires only data, no expert or any other stakeholder action or knowledge required. Nevertheless, as it relies on data obtained independently of the RS under evaluation, this task and the associated evaluation thus remain a challenging task [142]. The literature assumes that for each target learner, one ground-truth LP exists (also called reference LP), i.e., LP that conducts to the expected goal, for example success at exams [85]. The challenge is thus twofold: (1) identify the ground truth LP, (2) evaluate to what extent the recommended LP is in line with this ground truth. The measures proposed in the literature differ by these two elements.

In the frame of LP recommendations, precision (see section 2.3.1) has also been adapted to fit the complex nature of LPs. It is often assumed that one LP is a series of recommendations of resources, where each resource recommendation is viewed as either a good or a bad recommendation [18]. Given a target learner, the recommended path can be compared to the ground-truth path by identifying the resources in the recommended path that is also part of the reference path: one recommended resource that is part of the reference path is a good resource [105]. Like we talked above, this Some times the ground-truth is not easy to ensure.

Conversely, [143, 18] considered an LP as a whole: an LP is either a good LP or a bad LP. An LP is considered as good if its estimated learning effect on the target learner is greater than the expected learner's learning effect when no recommendation is proposed. Such as in [18], the precision is proposed as:

$$precision(all\ the\ rpath) = \sum_{i=1}^{|RP|} \frac{f(rpath_i) > threshold}{|RP|} \tag{2.47}$$

where the value function $f()$ is the effect function of the i_{th} recommended LP $rpath_i$, $|RP|$ is the number of recommended LPs, the *threshold* is defined by experts to determine a recommended LP is qualified or not. This measure can be viewed similar to the one proposed in [14] for online evaluation, in the sense that it also performs a comparison of learners' knowledge level with and without adopting an LP. The associated precision measure represents the proportion of good LPs.

Since finding enough learners to conduct online experiments is very costly, many researchers conduct to use simulated learners in their offline experiments, such as in [80, 24, 85], etc. Some offline evaluations under such a simulator can also be used in conducting online evaluation [25]. Nabizadeh et al. [80] proposed to simulate learner's learning process including consuming time and evaluation score through 2PL and 3PL IRT theories. The recommended LP is formed by maximizing its estimated score under consuming time constraints. Given such a recommended LP, its performance is evaluated by using the traditional MAE metric through comparing the real time and score with the estimated time and score. Here they consider the self-decided LP as the ground truth. The learner simulating method does overcome the shortcomings of offline evaluations: in lots of cases, there is not a standard ground-truth LP to compare with. But accuracy of this simulation is also an approximate behavior. Besides, the MEA metric does not consider any sequence characteristics of LPs. In this work, the evaluation of the effect relies on both learner and resource descriptions.The main disadvantage for them to use the MEA metric lies in the difficulty of estimation that requires much information and the estimation of several elements: each resource degree of difficulty, each learner's learning ability, the effect of a resource on a learner, and the effect of a learning path on a learner, which is rarely all available, especially in an offline setting.

Liu et al. [24] proposed to evaluate the accuracy of recommended LPs with two measures: promotion and logicality. Promotion represents the increase in the knowledge level of learners. This measure is close to the one presented in [14, 25], mentioned in the online evaluation section. They define an evaluation metric that evaluates the progress *pro* after a target learner adopted a recommended LP:

$$pro = \frac{E_{end} - E_{start}}{full\_score - E_{start}} \tag{2.48}$$

where the E_{start} is the exam result in before starting the learning, E_{end} is the exam result after learning the recommended LP. This metric could be used in both online and offline evaluation. Here, it is used in an offline setting where learners and their exam results are simulated by the IRT: for each question in the exam, the answer could be simulated by the IRT. The computation of *pro* is really a good enough metric. But the use of the IRT to simulate the score should be under the help of an expert which is a heavy workload.

In this section, we identified many measures proposed for traditional RSs. Evaluation measures dedicated to paths are rare, especially for LP. We can still identify two types of measures: those commonly used in many other domains and adapted to LP recommendations, and those specifically designed for LP recommendations.

As we said in the previous section, most of these metrics share one main drawback: Nor do they take into account the associativity of resources within an LP.

2.4 Markovian Algorithms in Recommender Systems

Recently, some researchers have started to consider to solve recommendation tasks as decision making problems [144]. Among the approaches proposed, Markovian algorithms are particularly

widely used [65, 66, 87, 86, 85]. On the one hand, Markovian algorithms can be treated as a kernel of a hybrid recommender model because of their sequential properties and their good interpretability; on the other hand, as a decision making algorithm, Markovian algorithm can directly change a recommendation problem to a decision making problem with the feature that it does not require a lot of detailed information [79]. As we already presented the Markovian chain in a previous section, here we will mainly focus on two Markovian algorithms that are popular in the literature: Markov Decision Process(MDP) and Partially Observable Markov Decision Process(POMDP). These algorithms are used both in general contexts and in the e-learning context as well.

MDPs and POMDP have been proven to be useful in solving a variety of sequential planning problems. They identify the optimal actions to be taken by an agent (user, robot, etc.) to reach a given goal. More than a decade ago, MDP have been studied in RSs, for a short-term goal [63]. Recently, Huang et al. [145] proposed to view long-term goal recommendation task as a MDP. On the premise of considering the sequential characteristics, in education domain, POMDP further considers some observable characteristics of learners, such as the work of Rafferty et al. [86]. Indeed, MDP and POMDP can manage the long-term effect of each recommendation, as well as the expected value of a recommendation.

2.4.1 Markovian Algorithms

We will start by presenting the background of the MDP and POMDP, as presented in [146].

In the Markovian decision process domain, the term "agent" is often used to describe the virtual actor or decision maker [146] (a robot or a user in reality). We will use the term *agent* to represent an online/offline learner.

As defined in the first chapter of [146], the output of Markovian decision process is a policy (noted π), that is a way of defining the agent's action a selection with respect to the changes in the environment related state s. At time point t, when the policy is deterministic, the policy is noted as $\pi(s_t) \rightarrow (a_t)$; when the policy is stochastic, the decided action a_t is a distribution $\pi(s_t, a_t) \rightarrow [0, 1]$.

Markov Decision Process

"Markov decision process (MDP) relies on the concept of states that describe the agent's current situation, the actions (or decisions), the dynamics that influence the process, and the rewards that observe the transitions between each state. MDP presents the probability of triggering a transition to state s' and receiving a certain reward r when taking decision a in state s" [146].

A MDP can be defined as a tuple: (S, A, p, r, γ), where:

- S is the state space.
- A is the action space.
- $p()$ is a transition probability function.
- $r()$ is the reward function.
- γ is the discount factor.

In robotic, a state may include the position [147], in online shopping, the purchasing power [148], in e-learning the level of knowledge [86], etc. A reward is a gain or loss that the agent receives by doing an action leading to the next state from the current state. An action, or

one step policy, guides the agent on what to do next. The transition probability function $p()$ indicates which states are likely to appear after the current state. For a given action a and a state s, $p(s'|a,s)$ denotes the transition probability that the agent transits to state s' after taking action a in state s. Note that the transition probability is a probability distribution, so the $p()$ should meet the following condition: $\forall s, a, \sum_{s'} p(s'|s,a) = 1$. Traditionally speaking, the transition function $p()$ is viewed as a $|S| \times |S|$ matrix. For a stochastic MDP, the action a is a probability distribution of possible actions. Unlike S and A, once set, the transition and reward functions can not change across time. It is called a stationary decision process. When been used in an off-line environment, the MDP and POMDP must be stationary decision processes.

Solving a MDP problem implies finding a policy π in a given set of $(S, A, R, p(), \gamma)$ where the optimal *criterion* for the MDP under consideration is optimized

Criterion is a digital measurement standard, whose purpose is to describe the characteristics of the policy that will provide the best sequence of rewards and used to select the best policy. In the criterion, the reward is based on the expected cumulative sum of rewards along a learning trajectory. The most used reward criterion is discounted criterion, following the expected cumulative sum with a discount factor γ:

$$\begin{aligned} R &= E[r_0 + \gamma r_1 + \gamma^2 r_2 + ... + \gamma^n r_n | s_0] \\ &= E[\sum_{i=0}^{n} \gamma^i r_i | s_0] \end{aligned} \tag{2.49}$$

where R is a general criterion, each step of reward r is computed by the reward function $r(s,a)$:

$$r(s,a) = \sum_{s'} p(s'|s,a) r(s,a,s') \tag{2.50}$$

All the explanations above allow a definition of the value function of MDP, which is used to evaluate a given policy π from a start state s_0 to a computed value (could be either loss or gain). And this computed value is the expected gain of the whole policy. The value function V computes a value related to the predefined criterion. Given an initial state s, it can be presented as: $\forall \pi V^\pi : S \to \mathbb{R}$. And the value function can be presented as:

$$\forall s \in S, V^\pi(s) = E^\pi[\sum_{t=0}^{n} r_t | s_0 = s] \tag{2.51}$$

where E^π is the expectation of all the possible states' outcomes computed by the predefined reward function in the agent's trajectory.

If an optimal π^* is the optimal policy in the policy space Π, i.e., it leads to the maximum reward computed by the value function V for the agent. If such an optimal policy exists, it should meet the following condition:

$$\forall \pi \in \Pi, \forall s \in S, \pi^* \in Argmax_{\pi \in \Pi} V^\pi, V^\pi(s) \leq V^{\pi^*}(s) \tag{2.52}$$

The MDP solver aims at finding such an optimal policy π^*, if such a policy exists, with respect to a given criterion mentioned before.

Partially Observable Markov Decision Process

When an agent does not know its real state, i.e., the agent only observes a part of information or condition partially which cannot contribute to decide exactly its current state, we say that

the MDP is changed to a POMDP: a Partially Observable Markov Decision Process (POMDP). POMSP generalizes MDP, and manages uncertainty in decision processes. POMDP are highly popular in robotics to determine the sequence of actions a robot has to perform to reach a predefined goal. In recent years, POMDP has been studied to solve education-related problems.

As MDP, POMDP is defined as a tuple $(S, A, T, p, r, \gamma, \Omega, O, b_0)$ as noted in [79], where:

- S is state space where for each state $s \in S$.
- A is action space where each possible action $a \in A$.
- $p() : S \times A \times S \rightarrow [0, 1]$ is a transition probability function between states.
- $r()$: $S \times A \times S \rightarrow \mathbb{R}$ is reward function .
- Ω is the observation space where each possible observation $o \in \Omega$.
- $O : S \times A \times \Omega \rightarrow [0, 1]$ is a probabilistic observation function.
- b_0 is an initial probability distribution over start state s_0. The belief state b is updated from b_0, i.e. after taking each action, it should be updated.
- $\gamma \in (0, 1]$ is a discount factor.

Since the agent in POMDP does not know its real stqte, the control part of this uncertainty is based on three new parameters (that are not part of MDP): initial probability distribution over states b_0, observation space Ω and observation function $O()$.

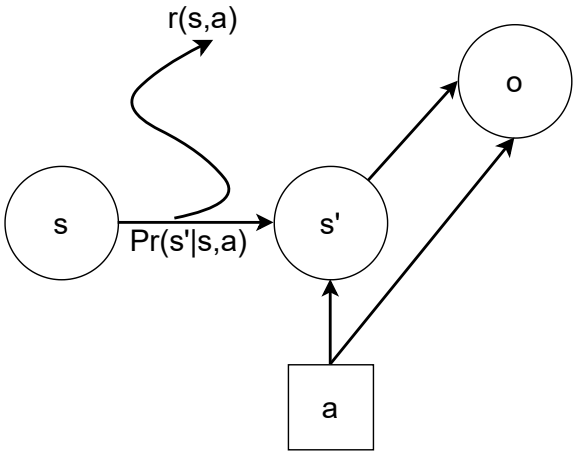


Figure 2.3: Overview of a POMDP.

As illustrated in Figure 2.3, at a given time point, the agent is in a state s. In the next time point, given an action a, the state changes to s' under the transition function $Pr(s'|s, a)$, and the related observation under the given a, s is o. The probability of s' is acquired from belief state b, so the belief state should be updated after each action taken. This observation o is given by the observation function O based on the related state s' and action a. With the

observation o and the action a, then the agent can update the information, the state changes to a next state s'. Then the agent gets the observation o(related to s') from observation function O. Recall that in MDP, we have the probability of observation as $Pr(o|s,a) = O(o|s,a)$. In POMDP, Equation 2.53 stands.

$$\forall t, \forall s, \sum_{o} Pr(o|s) = 1 \tag{2.53}$$

which means that one observation can map to several different states and all the observations of a state sum up to 1.

Recall that in the very beginning of a POMDP, the initial belief state is noted b_0 and it is a distribution of the possible start state based on the initial information that the agent got or set by the expert (an example in Appendix A.3). It is under the control of the preceding belief state, observation, and action that will be taken by the agent. During the same time, the belief state should also be updated after taking each action. If b is the current belief state for the current state, after taking action a, getting observation o, it becomes b_a^o, the current state changes to s', according to the Bayesian formula, it is derived as shown in 2.54 bellows:

$$b_a^o(s') = \frac{Pr(o|s',a)\sum_{s\in S} Pr(s'|s,a)b(s)}{\sum_{s''\in S}\sum_{s\in S} Pr(o|s'',a)Pr(s''|s,a)b(s)} \tag{2.54}$$

The details of belief derivation is shown in Appendix A.2.

In POMDP, rewards, transition probability for belief and the update of observation function should be computed with b. From such a given belief b, if the agent takes an action a, it can get a conditional observation that can be defined as:

$$\begin{aligned} O_c(b,a,o) &= Pr(o|b,a) \\ &= \sum s \in S \sum_{s''\in S} Pr(o|s'',a)Pr(s''|s,a)b(s) \end{aligned} \tag{2.55}$$

Based on the conditional observation and the transition probability from b to next b' given an action a can be defined as $\phi(b,a,b')$:

$$\begin{aligned} \phi(b,a,b') &= Pr(b'|b,a) \\ &= \sum_{o\in\omega} O_c(b,a,o) \end{aligned} \tag{2.56}$$

And the reward for this belief b state is a sum of all the states related to b:

$$\rho(b,a) = \sum_{s\in S} r(s,a)b(s) \tag{2.57}$$

The policy of a POMDP is noted as $\pi : S \to A$, is got from a value function V^π under the state distribution b and action a:

$$\begin{aligned} V_b^\pi &= \arg\max_a [r(b,a) + \gamma \sum_{o\in\Omega} Pr(o|b,a)V(b_a^o)] \\ &= \arg\max_a [\sum_{s\in S} r(s,a)Pr(s|b) + \gamma \sum_{s\in S}\sum_{s'\in S}\sum_{o\in\omega} Pr(s|b)Pr(s'|s,a)Pr(o|s')V_b^\pi] \end{aligned} \tag{2.58}$$

where the b_a^o is the next belief state under the observation o, taking action a. The optimal policy can be solved by getting the max value function of POMDP.

The complexity of POMDP is the main limit, which is highlighted by lots of researchers such as Young et al. [149], Chen et al.[150]. With a well defined model, the time and memory consumption are related to both the number of states and the number of actions. And there exist lots of solvers to solve an approximated policy of a POMDP decision making problem. To minimize complexity, we present our solution in the next Chapter.

2.4.2 Markovian Algorithms in Education

Markovian algorithms are becoming more and more popular in RS for the educational domain, mainly due to the fact that prerequisites (and the associated sequential nature) are very important in education and that Markovian algorithms inherently take into account the confidence (explainable) problem [40] thanks to its probability-based nature. Before presenting the associated related work, we elaborate that for all the Markov bases algorithms in education, a state represents the learner's knowledge state, i.e. the learner's knowledge level associated to the concepts managed [86, 134, 79].

The definition of a state in education usually contains a knowledge level and the related learning resource. In almost all the Markovian models in e-learning, such as in [79, 134], the learner's knowledge level (called concept or other name in different literature) is the most important part of the model. The level of knowledge is generally inferred by the learner participating in actions with feedback such as quiz, exam, etc. As an important part of the state, this knowledge level is used in the computing of reward function, updating of belief state, observation, everywhere of the POMDP model.
The action is usually defined as leading from current learning resource to another one [134, 87, 85], or using other teaching actions, such as giving prompts [79, 86].

F. Mi et al. [87] focus on the problem of frequent changes of learners' interests and preferences, with a RS that needs to be adapted to these changes. Their application is the recommendation of forum threads in MOOCs, where threads are highly dynamic. They proposed to apply a context tree structure to this sequential recommendation task and this context tree is based on a variable-order Markov model (VMM). They used pattern mining to find the suffix for the candidate set and this suffix relationship is used in the Markovian model to compute the probability of the next resource. In addition, they also used an expert based model with the VMM context tree to increase the accuracy. In this approach, as exploring all options for all contexts is not feasible, the probability of the next possible recommendation is approximated.

Liu et al. [24] used a MDP based deep knowledge tracing model to recommend learners learning resources. The current state is defined as the current knowledge level and the learning resource. The knowledge level is computed by deep knowledge tracing, based on the learner's learning history including notes. To represent the learning process, they used a one-hot encoding:

$$T^j = \begin{cases} 1 & if\ res_j\ in\ the\ learning\ process, \\ 0 & else \end{cases} \tag{2.59}$$

Given N learning resources in the graph, the learning target can be defined as $T = \{0, 1\}^N$ where 1 represents the fact that the resource is in the target. The reward is as described in section 2.4.1. In the education domain, the reward usually represents the increase of the knowledge level. Besides, they mine the inherent semantic structure among learning resources by using a prerequisite graph. The training process of the model is based on this prerequisite graph. To get a better performance, they used a hybrid model that combines MDP and knowledge tracing. Although the performance improves, this combination raises some problems: the training of the

neural network requires a large amount of data, a deep learning based knowledge tracing requires a large number of learners' scores, etc.

Hunag et al. [151] also used a MDP based deep reinforcement learning method to recommend learning resources. Their work presents a three-fold contribution:

- They proposed a trade-off between review and exploration: to control a skill through learning, learners should review to master the knowledge that has been cognitive, but also should explore the knowledge that has not been mastered;
- They defined a method to smoothly enhance the level of difficulty of the learning resources: easy-to-difficult learning process is more acceptable;
- They defined different engagement level for different learners, i.e. recommends difficult resources for advanced learners and easy resources for low level learners.

Based on these three contributions, Huang et al. modeled the reward function into three sub-reward functions. Traditional educational RSs generally focus on a single target: learners who do not have a good grasp of a given knowledge point are recommended resources covering that specific knowledge point, so that they focused on this point. Different from traditional educational RSs, this work is based on the combination of the three sub-reward functions, more accurate recommendations of learning resources is proposed, considering multiple perspectives.

POMDP also considers the previously mentioned learner knowledge, domain structure and goal (see chapter 1). These attributes are represented in the state definition, transition and observation modules [86, 85, 134]. A more detailed setting of state s and a series of functions such as transition $p() : S \times A \times S \leftarrow [0, 1]$, reward R, etc., changes according to the state. These problems are all essentially caused by the partial observation of the state. Hence, the definition of a state in POMDP is even more important.

A. Rafferty et al. [86] proposed the first work that formulated teaching as a decision making problem and they solved it through POMDP models. Given an objective (a learner knowledge), the POMDP selects optimally personalized teaching actions for individual learners. The learner's real state at a given time is partially observed and corresponds to the state of the POMDP model.

In their work, an action is defined as an education action space and includes two main parts: take a quiz or learn. The next (knowledge) state depends only on the current (knowledge) state and the pedagogical activity (the resource studied by the learner, the action). Three models are proposed in this work.

First is the memory-less model. The memory-less model considers the learner's knowledge state at the level of a single knowledge concept, i.e. in their work, a state does not contain any previously learned information. If the action is "pass a quiz for the acquired concept", which assessed the learner's current knowledge, the state will not be changed, and this is the only case that the state does not change; else, the state will transit to another state with the reward of the action This model has a low complexity since it has only one element in the state, a discrete model that memorizes the last actions, and a continuous model where a state is a probability distribution over the concepts related to learning resources.

Second, the authors proposed a memory-based POMDP model that maintains a fixed length M of a learner's learning history and the past M history's knowledge in the state. This memory can help the model to more efficiently determine if the action is consistent with the current state or not. From this history, the model contains information including not only the hidden guess at the true situation of the learner, but also the fully observed history of the past M actions. Thus,

the model can make a decision with more information than if a learner shifts to a situation, so it not only uses the current evaluated evidence but also all evidence from the M-step history, which is intended to be a more accurate representation of the learner state.
At last, the continuous model is a more complex model based on the memory-based model. In this model, each learner maintains a probability distribution overall concept. For a given learner, the concepts that she/he has not ruled out should all have a non-zero probability. The state is defined as a $|C|$-dimensional, continuous valued vector that sums to 1, where C is the set of possible concepts. For this model, the system must maintain a belief state over the infinite number of possible knowledge states. The difference is that this model used an infinite state space to consider all the possible situations. This continuous model includes much more information than the memory-based model and its complexity is much higher than the previous two models.

In their experiments, they confirmed that the continuous model and memory-based model are much faster to help a learner control a predefined learning goal. At the same time, there are very few differences between the memory-based model and the continuous model. Based on that, they demonstrated that very complex models do not necessarily make learners acquire knowledge faster. To enhance the performance, we could make this state more complex such as the model proposed in the work of A. Ramachandran et al. [79]. Besides the simple state definition, there is another drawback for these three models: their model requires detailed annotation for each knowledge concept. This is difficult to do for some learning platforms.

There are some researchers who consider the knowledge level at a more complex level. Reddy et al. [85] proposed to use POMDP in deep reinforcement learning, associated with spaced repetition. The model differs from [86] by two main elements. First, no information about the content of the resources is exploited. Second, it manages repetition to increase knowledge level retention (spaced repetition). The state here is defined by resources difficulty, the time delay between two learning actions for the same resource, and the memory strength of the learner. The repetition follows by a policy π from a model-freed recurrent neural network (RNN). To the best of our knowledge, this work is the first that uses a POMDP based on deep reinforcement learning and the first work that combined POMDP with memory (and repetition). The main limit of this work lies in the volume of data required to train the model, as it relies on deep reinforcement learning, which makes its application limited, especially in education, where the number of learners or traces is often limited. Besides, as the computational complexity of RNN is quite high, the state definition is very complex, which exponentially increases the complexity of the model.

Ramachandran et al. [79] proposed to use POMDP on robot tutors to provide suitable teaching actions to help learners. The background of this work did on mathematics teaching for primary school students: each *one* math exercise is treated as an independent POMDP problem. A state consists of knowledge levels, engagement levels, and math problem attempt numbers. From low to high, the knowledge level takes one value among four; the engagement level is divided into two levels and the max number of attempts for the problem is set as 3. Since ONE exercise is treated as an POMDP model, the size of the state space can thus be defined as $|S| = 4 \times 2 \times 4 = 32$. The action space consists of six tutoring actions, and the observation is the accuracy of the attempt (correct or incorrect) and the speed (slow, medium, or fast) at which the learner answers the question. In their modeling, the most noteworthy side is the proposed transition model: $T(s, a, s') = Pr(s'|a, s)$ can be derived by examining the likelihood of change in the attempt s_a, engagement s_e, and knowledge s_k dimensions of a single state s, and these three parts on probabilistic are independent of each other (the independence is implicit, if not,

the transition will be more complex):

$$Pr(s', a, s) = Pr(s'_a, a, s)Pr(s'_e, a, s)Pr(s'_k, a, s) \tag{2.60}$$

The POMDP model proposed is designed for a naive education problem. Indeed, they define the state in a very small scale POMDP problem. The drawback is that for each math problem, they should re-build a targeted model. If we need to apply this model in a relatively complex environment with more data and more learning resources, the main limitation of this work lies in the complexity of this model i.e. states number will be exponentially increased.

To summarize, in the educational RS literature, a state is generally made up of at least two main components [86, 85, 134]:

- The history of resources the learner interacted with, including the current learning resource.
- The learner's knowledge level. When resources are associated with concepts, the knowledge level is related to these concepts [86]. The estimation of a learner's knowledge level is of interest for a large number of works, but is not the focus of this thesis.

3

Managing Learner's Memory in POMDP for Learning Path Recommender Systems

In this chapter, we introduce our contribution related to the design of a learning path recommendation system (LPRS). Two main elements have guided our reflection. First, an LP recommended to a target learner should be in line with this **target learner's learning behavior**. Second, a recommended LP should **improve the target learner's level of knowledge**. Besides, our models do not need any resources content just like Zhao et al. did in [101]. Note that in classic e-learning environment, parts of cases do not have any resources content and rare researchers focus on this kind of RSs.

The literature presented in the previous chapter has highlighted that several approaches have been proposed to perform path recommendations, including in the educational frame for LP recommendation. Especially, Markov-based algorithms have shown to be good at dealing with sequential problems [60, 105]. Markov-based algorithms were traditionally used in the field of robotics and automation, they have started to be studied in the field of RSs these last years [79], [152]. Besides, path RS has recently become a popular research direction as discussed in section 2.2.1, where Markov-based algorithms is an important part of the researched conducted.

In the context of education, where path RSs are called learning path recommender systems (LPRS). In this domain, Markov-based algorithms have not been much studied but recent works have confirmed the relevance of MDP and POMDP [86, 26]. This is the main reason why we propose to study POMDP for LPRS.

Recall that in the literature, the educational recommendation models generally exploit the learners' past interactions with the resources of the learning management system to determine the LP to recommend to a target learner. In addition to these traces, most of the works proposed to exploit the description of the resources, mainly in terms of concepts [86]. These works have shown the added value of such data, but their use is limited to cases where pedagogical resources are accurately indexed. During the same time, not all contexts are associated to such data. We thus aim at designing an LPRS that can be used when no metadata or concepts can be associated to resources.

Besides, even when such additional information (about learners or resources) is available, the state of each learner cannot be fully known. It is said to be partially observable. For example, in robotics, where the agent is a robot, the partial observation is mainly about the current robot's coordinates. In education, the partial observation is generally about the learner's learning state

(knowledge, concentration, motivation, even memory). POMDP models are designed to manage such partial observation, whatever is its nature and the application context. This is the reason why we, and many researchers, propose to use POMDP in the educational frame.

This chapter is divided into three parts: the first section presents how to formulate LP recommendation as a POMDP; the second section describes two POMDP models that consider learners' memory strength for personalized LPRS.

The main notations used throughout this chapter are summarized in Table 3.1.

Notation	Description
$\mathbf{D}$	The whole dataset
tl	target learner
tr	target resource
res	a resource
RES	set of all resources
s, s', s''	states in a POMDP model. s' is the next target state and s'' is the general next state
S	S is a state space that includes all the possible state s
$KL()$	knowledge level of a target learner on a target resource
LP	a Learning Path
$history(tl, LP)$	history of the target learner in a given learning path
$NLT(tr)$	number of learning times of a resource tr in a learning path LP

Table 3.1: List of the main notations used throughout the chapter

3.1 Formulating LP Recommendation as a POMDP

3.1.1 Definition of A Basic POMDP Model in Education

In this section, we define a basic POMDP model under the constraints of the education domain. As described in section 2.4.1, a POMDP based model is a tuple: $\{A, S, \Omega, p, O, R, b_0, \gamma\}$. The POMDP-based recommendation model that we designed includes these elements, where some of them, especially the state space, are designed according to the characteristics of our educational context and the objective of recommending an LP. Here we will describe how we propose to manage the elements of the tuple in a POMDP.

This basic model will be used as a baseline in our experiments.

State As explained in the previous section, the state in a POMDP used for education generally represents the learner's knowledge (level) and it is related to the learner's previous LP. Let us recall that the literature generally used these two attributes to define a state of a POMDP for a learner: the history of this learner and his/her level of knowledge. The model we proposed is in line with this definition [86, 134, 79]. In this baseline model, a state contains both the LP history (noted $LP(tl)$) and the knowledge level for a target learner (noted $KL()$). Each of these two attributes is a component of a state s, they are noted s_{LP}, and s_{KL}:

- s_{LP} represents the learner's history LP, i.e. the last N resources he/she studied/accessed, including the current one. The history LP is represented in the form of an array $s_{LP}[N]$. This choice is in line with [79]. The higher N, the higher the number of states, hence the higher the complexity.

- s_{KL} represents the estimated knowledge level of a learner, as in [86]. As in our context resources are not indexed and are the atomic component in data, we propose to represent the learner's knowledge for each resource in s_{LP}. To limit the complexity of the model, we discretize the knowledge level, as proposed in [79]. So, each value in s_{KL} ranges between 0 (totally unknown) and K (mastered resource). The following section will introduce how we estimate this knowledge level for a learner and a resource.

Note that it is possible improve the POMDP model by adding one or more state dimensions.

Action The data is mainly made up of LPs, where an LP is composed of a sequence of learning resources (id). Traditionally in the education domain, the action space contains all the possible learning resources that a target learner can access, such as described in [85, 153, 153, 134], etc. We also define the action space A as the complete set of learning resources.

Observation The observation we choose here is stochastic. The observation function evaluates the distribution among the possible knowledge levels $0 \leq KL \leq K$: given that action a is taken in state s, the observation function $O = p(o|s, a)$ indicates the probability distribution of observing o. Our definition of observation is consistent with [79].

Transition Function Under the definition of the state in this basic POMDP, the transition function $p()$ represents the likelihood that a learner moves from one state with a knowledge level and other information to another state. The traditional transition function is defined as $p(s', s, a)$ which is under the control of state and action. In our definition, we propose that each transition probability is associated with probabilities that are learnt from the learners' traces of interactions and another important information: knowledge level. Based on the traditional definition, we propose the definition of the transition probability as:

$$\begin{aligned} p(s', a, s) &= Pr(s'|s, a) \\ &= Pr(s'_{LP}|s, a) \cdot Pr(s'_{KL}|s, a) \end{aligned} \tag{3.1}$$

where s'_{LP}, s'_{KL} are two components that formed the state s, and the related likelihood are based on these two components.

Since in our transition function, there exist not only resource to resource transition, but also knowledge level to knowledge level transition, so our transition definition must contain the above two components. At the same time, the transition probability between resources is only based on the probability from one resource to another and the transition of knowledge level computes the probabilities of knowledge levels that the learner will get when learning the next resource, so we define the transition function as presented in Equation 3.1.

Reward Function Recall that a one-step reward is a gain or loss computed by the reward function $r(s, a)$. It indicates that an agent in state s after taking action a the reward it will receive. In the educational context, a reward is mainly getting through the evaluations taken by the learner, for instance in [24]. Mathematically, the reward function relies on the state and the action. It is a gain or loss mainly through the change of knowledge level and change from one resource to another. As a state includes knowledge level s_{KL}, after taking an action s_{KL} should be updated. Note that the knowledge level is estimated in the state, in a more complex model, we could use other information to help the computation of getting a more detailed reward. For instance [151] used changes in resource difficulty in LP and changes in learners' engagement with

different resources as supplementary information. When more information exists, the reward function could be defined with several information dimensions. For the reward function, in each step there is a cost for taking an action, here we defined this cost as a unit for both the cost and reward. Under this definition, we propose the following reward function:

$$r(s,a) = r(s_{LP},a) + r(s_{KL},a) \tag{3.2}$$

where the $r(s_{LP},a)$ is a sub reward function for dimension s_{LP} based on cost and general prerequisite from the change of resource, related with the transition probability, (presented in Equation 3.3 and $r(s_{KL},a)$ is a sub reward function for dimension s_{KL} based on the change of knowledge level after taking a resource pointed by action a. They are defined separately in Equation 3.3 and Equation 3.4.

$$r(s_{LP},a) = \phi + ur \times Pr(s'_{LP}|s_{LP},a) \tag{3.3}$$

where ϕ is a general cost, ur is a unit reward and $p(s'_{LP}|s_{LP},a)$ represents the prerequisite probability learned from the dataset. Since the prerequisite relationship is very important in the educational context, we decide to add it into the reward function as presented in Equation 3.2.

$$r(s_{KL},a) = \begin{cases} ur, if\ the\ KL\ increased\ by\ 1\ level; \\ 2 \times ur, if\ the\ KL\ increased\ by\ more\ than\ 1; \\ 0, else. \end{cases} \tag{3.4}$$

where the ur is the unit reward as presented in Equation 3.3. The increment here is related with the KL increment. Normally speaking, the KL would not increase too much, so if the KL increases more than 1, the reward is defined as $2 \times ur$. The details of the definition of different reward parts with dimension should be adjusted according to the model.

Observation The observation for a state is stochastic. It is a distribution of probabilities on the next resource's possible knowledge levels. The knowledge level in a state is the most possible one, the observation indicates the probability: given that an action a is taken in state s, the observation model $O(o|s,a)$ indicates the probability of observing o, as in any POMDP [79].

3.1.2 Estimating The Knowledge Level of a Resource

Many works in the literature exploit (and estimate) the knowledge level in different ways. In addition, some works mentioned concepts, others simply knowledge. This attribute is very important in the state since we can not get the knowledge directly, here we collectively refer to the *knowledge level*, which is defined as follows:

Definition 5 *The knowledge level of a learner is the learner's mastery of knowledge at the current time. It encompasses two elements learner's behavior, and learner's academic performance.*

Even though POMDP are known to be complex, in the literature dedicated to education their complexity is not often mentioned. From our view, the main limit of the works proposed comes from the number of states. Especially, the representation of the knowledge levels, even when it is associated to each concept, makes this number explode. To limit the complexity of the model, we propose to discretize the knowledge level, in line with [79]. The K knowledge levels have to correspond to the diversity of knowledge of all the learners. Before computing the knowledge level, we choose to use the similarity of LP and final exam to divide all the learners

into K groups. The division of knowledge levels is based on the fact that a group of learners shared the same learning habits, learning ability, etc. This idea is inspired by [18].

A knowledge level function $KL()$, that represents the knowledge level of a target learner tl on a target resource tr is computed based on this target resource that the target learner took. Rafferty et al. [86] used a table to store concepts related knowledge levels, i.e., they have a description for each distinguished knowledge point described as *concepts* in their work. Since knowledge level is only related with one concept, if through a test that proved the concept control situation changed, the related knowledge level changed. Here we used the same method. The only difference is that we have no description of the resource content, our knowledge level is at the resource level.

Generally speaking, there are two cases for computing the knowledge levels.

First, given a target learner and a target resource, if the action is an evaluation resource (for example quiz, exam), the knowledge level for this current evaluation resource er in a given LP $KL(LP, er)$ can be directly estimated from the grade obtained by the learner on this evaluation resource ($eval(LP, er)$). We propose to evaluate it as presented in Equation 3.5, so that the knowledge level ranges from 0 to K.

$$KL(LP, er) = \lfloor \frac{eval(LP, er)}{(eval_{max} - eval_{min})} * K) \rfloor \tag{3.5}$$

where $eval_{max}$ and $eval_{min}$ are respectively the maximal and minimal evaluation values. $\lfloor\ \rfloor$ represents the round value in parameter to the nearest integer.

Second, if this action is not an evaluation resource, the knowledge level associated to this resource cannot be obtained directly, it has to be estimated. In most learning scenarios, a learner can not take an evaluation after each resource accessed, which leads to a lack of information on learners' knowledge level. So we propose to estimate the knowledge level of a target learner tl on a target resource tr by exploiting the nearest evaluation after tl has learned. It is based on a hypothesis that before an evaluation (quiz or exam), the knowledge levels of resources are all related to this evaluation, in the limit of the preceding evaluation, and the closer the resources are to the quiz she/he is taken, the closer the knowledge level is of knowledge level of the evaluation resource.

So, the knowledge level of the current target resource tr can be estimated from the evaluation resource that follows tr. Based on the assumption that the more resources are studied by learners, the higher their knowledge level, we propose to apply a discount factor (λ) to represent the fact that the more distant tr is from er, the lower the knowledge for tr. Equation 3.6 presents the way we estimate the knowledge level on tr.

$$KL(LP, tr) = round(\lambda^{dist(LP,tr,n\_eval(LP,tr))} eval(LP, n\_eval(LP, tr))) \tag{3.6}$$

where $n\_eval(LP, tr)$ evaluation resource that follows tr in LP and $dist(LP, tr, n\_eval(LP, tr))$ is the distance (in number of resources) between tr and this following evaluation resource.

To explain this, Figure 3.1 presents an example of the estimation of the knowledge level of some resources, based a learner's LP.

The resources in red cube are evaluation resources and in green cubes are non-evaluation resources. A learner took an evaluation resource r_y (with grade $eval(r_y)$), then adopted the learning path $LP =< r_c, r_b, r_d, r_d, r_f >$ and finally took evaluation r_x (with grade $eval(r_x)$). For each resource that the learner accessed, there should be an associated knowledge level $KL()$. Since the lack of real evaluation, we propose to estimate resources' knowledge levels through the real evaluation of $eval(r_x)$. The estimated evaluation of one resource decreases as the distance

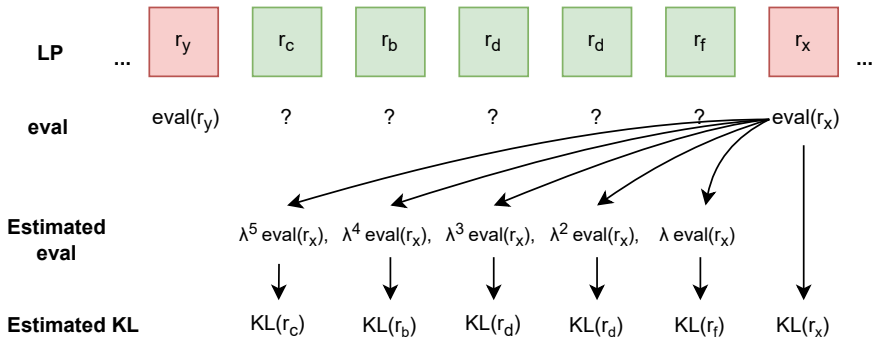


Figure 3.1: Example of a learner's learning path, and estimated levels of knowledge

from the next real evaluation increases. Then the estimated knowledge level can be calculated by this estimated evaluation, as presented in Equation 3.5.

In the following sections, we will focus on the contribution of the way we propose to represent and manage learners' memory in a POMDP-based LPRS, with the goal to foster the repeated recommendation of old resources. We propose two ways to manage learners' memory under the definition of the basic POMDP model.

3.2 A POMDP-based RS that Exploits Learners' Memory

As explained in the introduction, memory strength is important in education. We feel this factor needs to be carefully managed in the education RSs, but few works consider to use it in a POMDP based RS model. This is the goal of this section. To achieve managing the memory strength in a POMDP model, all of our contribution is mainly in the definition of a state s and a reward function $r()$. We aim to build RS models that tries to model the learner's memory in POMDP in a simple way. As far as we know, this is the third human memory built-in POMDP model.

The literature highlights that each time a resource is viewed (or reviewed), the associated knowledge is more firmly and even longer memorized by the learner. To distinguish the difference between human memory and mechanical memory, we write them into *learner's memory* and *memory* separately. Managing learner's memory generally achieved by using the spaced repetition technique, that fosters the repetition of already studied resources, to reinforce the associated knowledge level of the learners.

The first work that used POMDP and spaced repetition for educational RS is [85]. This work has shown the relevance of such an approach with the limitation that the amount of interaction data required by their model is large. The main challenge that we highlight is to limit the amount of data required by the model, to make it usable in the greatest number of cases.

To conquer this limitation, we propose to model the learners' memory ability by changing the state definition of the general POMDP model introduced in the previous section. It is named M-POMDP, that stands for Memory-POMDP. But how to represent learner's memory in a POMDP? We propose to define an additional dimension in the state s, that represents the learners' memory. This new dimension will require an adjustment of the transition function $p(s', a, s)$, the reward function$r(s, a)$ and also observation function.

Inspired by [86], [79] and [85], to limit the complexity of the POMDP based model, we

propose simplify the representation proposed by [79]. Exploiting the number of times a resource has been studied in our model is inspired by [33].

In a learner's memory based spaced repetition model, there are generally four parameters: learning times, learning decay, difficulty of each learning resource and the memory strength of the learner. Below we explain our selection and operation of these attributes one by one:
1. Learning times:
In our associated model, an attribute is thus added to the state definition, that corresponds to the number of times each resource has been studied. This attribute is called "number of learning times" (NLT). s_{NLT} is thus also an array, where each cell represents the number of time a resource has been seen.
2.Delay of time:
Here the time decay is continuous which is not possible to be added in the POMDP due to the fact that it will make the number of states infinite. To reduce the complexity, our model does not contain the time delay, which may limit the accuracy of the representation.
3.Difficulty:
The difficulty of a resource and learner's memory strength could be obtained through the dataset. Our calculation of the difficulty of a resource is based on the proportion of a resource in the overall learning interaction. In the transition function Equation 3.1, $p(s'_{LP}|s,a)$ is based on the frequency for all the resources in the dataset. It is implicitly used as difficulty in our POMDP model.
4. Memory strength:
The learner's memory strength in our model is implicitly computed through the changing of knowledge level in the learners' state.

The advantages of M-POMDP are three-fold: 1. The number of states is not exponentially increased; 2. the additional dimension s_{NLT} will allow to improve the reward function; 3. Based on s_{NLT}, spaced repetition in the recommended learning path can be managed.

Given this way to manage learners' memory, a state s is thus defined with three attributes s_{LP}, s_{KL} and s_{NLT}. s_{NLT} represents a learner's last N resources he/she studied/accessed, including the current one ($s_{NLT}[N]$).

NLT is deterministically incremented each time the learner interacts with a resource.

To limit the complexity of the model, we set a maximal value for NLT: MAX_{nlt}. As in Figure. 3.2, when a learner has viewed a resource $MAX_{nlt} = n$ times, this value is no more modified, even if the learner reviews it again. The number of states is thus limited. This is in line with the traditional learner's memory model that represents that once a resource has been reviewed several times, a long-term memorization is reached and the resource should be less recommended even not recommended at all.

Besides, the rate at which a resource is forgotten decreases as the number of times it is learned rises [33]. By managing NLT, the POMDP explicitly models learners reviewing habits in the reward function. s_{NLT} thus plays an important role in the reward function, as a supplement to the estimated knowledge level :

$$r(s,a) = r(s_{NLT},a) + r(s_{LP},a) + r(s_{KL},a) \tag{3.7}$$

where $r(s_{NLT},a)$ is the reward function that computes the reward of NLT, $r(s_{KL},a)$ is the reward function that computes KL the estimated knowledge level, and the $r(s_{LP},a)$ is as we presented in Equation 3.3.

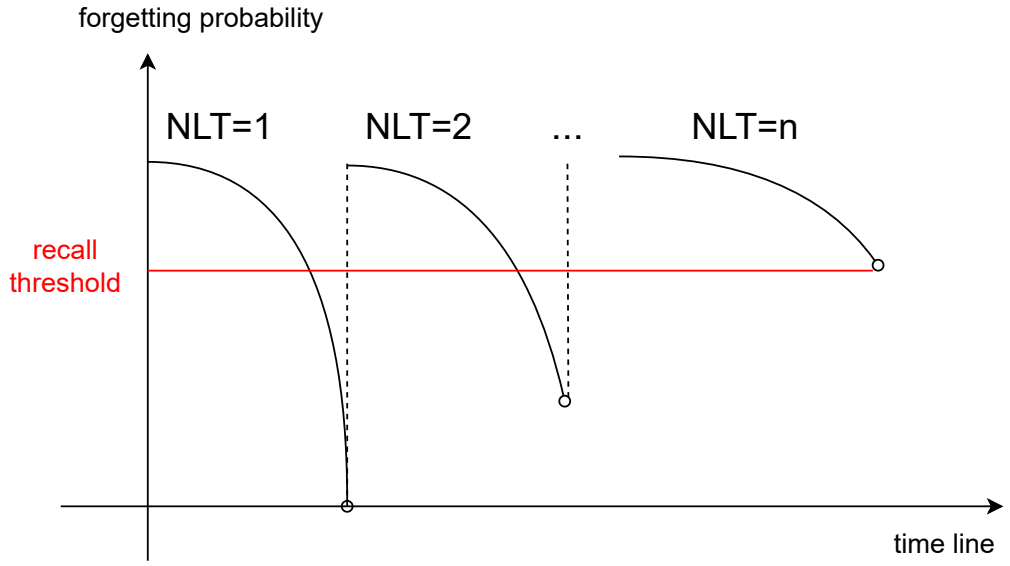


Figure 3.2: The forgetting curve of a target learner learns one knowledge concept several times.

The proposed $r(s_{NLT}, a)$ is defined as follows:

$$r(s_{NLT}, a) = \begin{cases} ur, if\ NLT\ increased; \\ 0, else. \end{cases} \tag{3.8}$$

where the ur is a unit of reward as introduced in Equation 3.3.

The transition function $p(s, a, s') = Pr(s'|a, s)$ is not impacted by the NLT. The transition function evaluated by three independent functions:

$$p(s, a, s') = Pr(s'_{NLT}|a, s) \cdot Pr(s'_{KL}|a, s) \cdot Pr(s'_{LP}|s, a) \tag{3.9}$$

where these functions represent the number of learning times NLT, knowledge KL and action a. Since the NLT is deterministic, the $p(s'_{NLT}|a, s) = 1$, so the equation remains unchanged compared to Equation 3.1.

To easily understand, the LP can be treated as generated in a directed graph which contains the prerequisites of resources. As presented in Figure 3.3, we considered that the model generates an LP based on the prerequisite graph. Each yellow cycle is a resource. The state in this graph is partially hidden: we only show the current resource in a state, we hided the NLT, KL and the history of an LP. The one direction of an arrow represents the next available resource for learning after the current resource, i.e., a arrow with one determined direction is a possible action. On each arrow with a determined direction, it should have a probability. The *distance* can be represent as $dist(LP, res_i, res_j)$, where res_i and res_j are two resources in the same LP. For example, the shortest distance between resource 1 and resource 4 is also 2, and the shortest distance between resource 2 and resource 3 is 2. Note that there are cycles in the graph. When

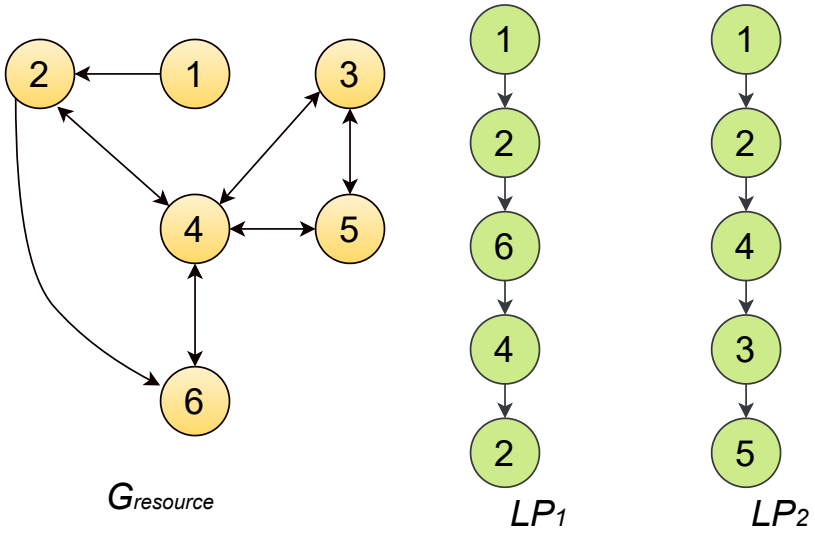


Figure 3.3: Graph of resources

the circle appears in an LP, it means that the learner has carried out a review activity, more specifically, it is a spaced repetition review activity.

The right part of Figure 3.3 includes two LPs that wrer generated from the left part. LP_1 includes a review for resource 2 and LP_2 does not include any review action.

This simple model faces some obvious limitations: the number of times a resource has been studied is stored in a simple way, it is directly and naively noted in the state, and in fact, the duration of time spent during each learning action is not considered in this indicator. Besides, we did not implement the time decay of a resource in our model.

3.3 Unique POMDP RS and Repeated Unique POMDP

We have highlighted that M-POMDP has limitations related to the simple way that the learner's memory is modeled. However, using a more evolved way to manage memory would lead to a significant increase in the complexity. This section is dedicated to overcome these two scientific difficulties. In Chapter 2 we highlighted that many researchers use a combination of two or more models together for personalized recommendation with the goal to decrease the complexity of the recommendation problem. We adopt such a strategy to achieve our objective.

As we present above in this chapter, the M-POMDP does manage the NLT.

In the literature review (section 2.1.4) we saw that the memory of learners (whether it is evaluated through the forgetting curve, a number of learning times, etc.) is generally managed within the learning path construction step. The choice made in M-POMDP is in line with this.

In the model introduced in this section, we adopt the opposite point of view. The learners' memory is not represented within the state of the POMDP, but is managed in an external part of the model, that supplements the LP construction process (the POMDP model). The goal is twofold. The decrease in the complexity of M-POMDP and a more accurate management of the learner's memory. Forming an LP in two steps is quite original proposition.

The first step is dedicated to the construction a skeleton of an LP. In this step, no learner's memory management is performed. The second step acts as a post-processing and is purely

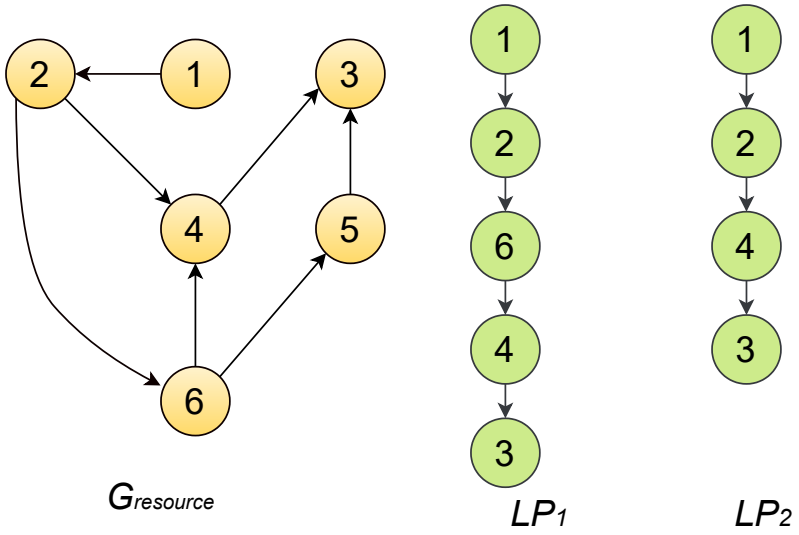


Figure 3.4: Acyclic-Graph of resources

dedicated to memory management.

Concretely, the LP formed in the first step, all resources appear only once. This LP is called a Unique LP (*U-LP*). This step is performed by a POMDP model, designed to force the construction of an LP without any repetition. The associated model is called the Unique POMDP (U-POMDP).
The second step takes *U-LP* as input and is designed to insert repetitions in *U-LP*, i.e., already recommended resources. These insertions are chosen to force the review of the associated resources, i.e. the repetition of resources. This second part of the model fully focuses on handling the learner's memory to constitute a spaced repetition based LP, called *U-LP*. The associated model is the Repeated Unique POMDP (RU-POMDP).

To the best of our knowledge, this model is the first one that manages repetition in this way in an LP-RS: constraining the construction of an initial LP and managing repetition in a post-processing step.
Let us now focus in details on both steps.

3.3.1 The U-POMDP Model

U-POMDP is a POMDP-based LP construction model, dedicated to forming an LP made up of resources that occur only once in the path. To force the unicity of the resources that occur in the LP, we make small changes in the definition of the POMDP. To understand the way U-POMDP works, we create a directed acyclic graph for the *U-LP* as shown in Figure 3.4, and based on the previously introduced Figure 3.3,

Let us first focus on the left yellow part of Figure 3.4. The graph *Gresource* is the same as the one in left side of Figure 3.3, except that it contains no cycle. The path generated from *Gresource* is a *U-LP*.

Note that the graph is built based on the transition function as we presented in Equation 3.1. According to this definition, the LPs generated in this acyclic graph do not contain any repetition. This will prevent from recommending resources that are already in *U-LP*. Two examples of such *U-LP*s formed from *Gresource* are shown in the right part of Figure 3.3: LP_1 and LP_2.

Regarding the U-POMDP model, the state definition is in line with the one of M-POMDP: the same two attributes s_{LP} and s_{KL} are used. However, the s_{NLT} is no more a part of the state definition. About the action space, it remains unchanged, it is made up of the set of resources that can be accessed by a learner. The reward function is same as the reward function as presented in M-POMDP: the reward function focuses on the knowledge level and resource's prerequisite relationship. The transition function can be described as in Equation 3.1. The observation is also about the target learner's knowledge level and it is the same as in Section 3.1.1.

Based on the definition of U-POMDP, we suppose that the LP formed has a reduced length. Besides, the number of possible actions in a given state is also reduced. Finally, the execution time will thus be reduced as well. Indeed, as the consuming time of solving a POMDP based problem is exponential to the length of the path, such a decrease in the length of the path formed will have a significant impact.

3.3.2 The RU-POMDP Model

The second step of the model is dedicated to the definition of a resource insertion strategy in *U-LP* to foster reviewing, i.e., spaced repetition. Three questions are raised here: 1) which resources should be repeated, 2) to what extent should they be repeated, 3) where should they be inserted in *U-LP*?

These questions will be answered by using a learner's memory model. As each learner has his/her own memory strength [36], we propose to use a personalized memory model (for each learner). This personalized model will allow identify which resources should be inserted in *U-LP* and where to insert them.

To this end, we are inspired by the well-known exponential forgetting curve [38], that defines the probability that a learner tl recalls a target resource tr $Pr_{recall}(tr, tl)$, presented in Equation 3.10.

$$Pr_{recall}(tr, tl) = -e^{\frac{-\theta(tr)t(tr,tl)}{ms(tl)}} \tag{3.10}$$

where $ms(tl) \in \mathbb{R}^+$ is the memory strength of the target learner tl, $\theta(tr) \in \mathbb{R}^+$ is the difficulty of the resource tr and $t(tr, tl) \in \mathbb{R}^+$ is the time since the learner tl has reviewed the resource tr. To model the memory curve, we have to evaluate the learner's memory strength and the difficulty of each resource in the dataset.

We propose to learn the memory strength of a learner ($ms(tl)$) from the data by considering his/her repetition behavior, from the traces of activities in the data. The details are as follows:

$$ms(tl) = \mu - \frac{\theta(tr)t(tr, tl)}{\ln(P_{recall}(tr, tl))} \tag{3.11}$$

where the $P_{recall}(tr, tl)$ is a threshold value, μ is a basic memory strength. Note that this equation is a simple transformation of Equation 3.10. The time slot of a resource ($t(tr, tl)$) is estimated from the dataset. The difficulty of a resource ($\theta(tr)$) is also learnt from the data: Equation 3.12 describes the computation of the difficulty: difficulty is expressed as the number of times the target resource appears in the dataset divided by the number of all interactions, i.e., we consider that the more a resource is repeated, the more difficult it is. The $\mathbf{D}$ represents the whole dataset.

$$\theta(tr) = \frac{\#\ of\ interactions\ of\ tr\ in\ \mathbf{D}}{\#\ of\ interactions\ in\ \mathbf{D}} \tag{3.12}$$

In a dataset without any content description, the frequency of resources in dataset is the only information that can be used to estimate the difficulty. This method is a traditional memory curve based method as presented in [38].

The literature considers that the forgetting curve is influenced by the number of times the resource has been already studied [93]: The more times you review, the slower you forget. Our model is in line with this work and we propose to consider this information to evaluate a more accurate probability of recall. The traditional memory curve only computes the probability that a resource will be memorized after the first learning. Given a target resource in a long-term learning distribution with multiple repetitions, as the number of learning times increases, the need for review this resource will decrease [36]. Note that this kind of review strategy is called spaced repetition. In our work, we use the memory curve and the Poisson distribution to realise the spaced repetition strategy. The choice of Poisson distribution can be explained by the following reasons:

- Most of the proposed models for realising the spaced repetition strategies are based on complex algorithms, this is inconsistent with our research goal.
- Spaced repetition has been confirmed by many literature that the need to review a knowledge concept decreases exponentially as the number of reviews increases.
- Among simple algorithms, according to the characteristics of Poisson distribution, it is very suitable to be used in spaced repetition : it is exponential; it includes both the NLT and the time slots; it is specifically used to model the distribution of the probability of repetition of events.

Since the probability that a resource re-appears decays gradually, the decay rate is calculated for each learning resource with different NLT. This decay rate can be modeled from the dataset as follows with the Poisson distribution:

$$P(NLT, tr, tl) = \frac{(\lambda t(tl, tr))^{NLT}}{NLT!} e^{-\lambda t(tl,tr)} \tag{3.13}$$

where NLT is the number of learning times of the target resource, $P(NLT, tr, tl)$ is the probability that a target learning resource is learnt NLT times (the number of NLT increases, the probability decreases, in the definition of the Poisson distribution, it is called decay) by the target learner, $t(tl, tr)$ is the temporal gap between current timestamp and the first learning action for the same target resource. λ is called arrival rate, for any target resource, we should find an appropriate λ from the data through the Equation 3.13.

The Poisson process uses the number of learning times of the target resource tr as parameter so that it decreases at an adequate rate. Note that NLT is a vector computed based on the learning history (noted $his(tl)$) and $U\text{-}LP$ of tl (line 3 Algorithm 1). In the generating of the review part, given a target learner with a target resource with a related NLT, the decrease of review need $decrease(NLT, tr, tl)$ can be presented as follow:

$$decrease(NLT, tr, tl) = \frac{1}{P(NLT, tr, tl)} \tag{3.14}$$

$$decrease(tr, NLT) \cdot Pr_{recall}(tr, tl) \geq \sigma \tag{3.15}$$

In our model, we define a criterion for judging that a resource needs to be reviewed in Equation 3.15. We multiply the decrease of need $decrease()$ and the recall probability $Pr_{recall}()$,

then compare it with a threshold value σ. When the right part in Equation 3.15 is bigger than σ, we think that the target learner could remind this target resource.

Next, we introduce in detail how our proposed spaced repetition is implemented through memory curve and Poisson distribution. As we talked about, *U-LP* is generated by the U-POMDP. Based on this *U-LP*, the algorithm 1 presents how *RU-POMDP* implements the spaced repetition. It adopts a greedy strategy to perform insertions. It iterates over each candidate target resource tr^* (from the set of resources in *U-LP*) , till the maximum length $MAX_{RU\text{-}LP}$ is reached. The $MAX_{RU\text{-}LP}$ is predefined before the RU-POMDP as proposed in the work of Liu et al. [24].

The inputs of algorithm 1 include the recall threshold σ, the length of the recommended LP $MAX_{RU\text{-}LP}$, the history LP of the target learner $his(tl)$ and the transition matrix TM (Algorithm 1, line 1).

First *U-LP* is formed by the function *U-POMDP*() (algorithm 1, line 2).

Recall that in our second part of RU-POMDP, we have a same NLT as we defined in M-POMDP model: it records the learning times of a target resource. It will be used in Equation 3.15 to help determine if a target resource should be reviewed or not. The NLT is computed by $NLTcal(U-LP, his(tl))$ function (Algorithm 1, line 3). And the recommended LP is initialized by this *RU-LP* (Algorithm 1, line 4).

With these information, the algorithm decides how to form a spaced repetition LP in a *while* loop. Before the recommended LP satisfies the length condition, for each layer of the loop, the should-be-repeated target resource tr^* is firstly decided by the $EarliestTime(tr, tl, nlt(tr), \sigma)$ function. This function also returns the earliest time slot et that the tr^* will be forgotten by the target learner (see Algorithm 1, line 6). Note that the Poisson distribution and memory forgetting curve we presented above are used in this function. More specifically, as the $NLT = nlt(tr)$, this function combines Equation 3.10 and Equation 3.14. It gets et through comparing the value of $decrease(tr, NLT)Pr_{recall}(tr, NLT)$ and the threshold σ.

Then the position where the tr^* will be added in the *U-LP* is computed by the function $ind(tr^*, et)$ (Algorithm 1, line 7). The position is noted as pos.

Before adding the target resource tr^*, a check is made at the position pos to guarantee that the resource to be inserted tr^* is coherent with its neighbor resources in *RU-LP*. β_{tl} and β will be used to decide take such an insertion action or not. Under this condition, we firstly calculate the transition probabilities of the to-be-inserted tr^* with the two neighbor resources in the original *U-PL*, then we return the average value β_{tr} of these two transition probabilities (Algorithm 1, line 8).

Next, the algorithm computes the original transition probability β at position for the resources in the *U-LP* (Algorithm 1, line 9).

If $\beta_{tr} > \beta$, the insertion will be operated by another function $Insert(RU\text{-}LP, tr^*, nlt(tr^*), pos)$ (Algorithm 1, line 11). This function also updates *RU-LP* and $nlt(tr^*)$ if the insertion can be done. Even tr^* is possible to be added, there exist two cases:

1. if $pos > |RU\text{-}LP|$), tr^* will be inserted at the end of *RU-LP*, and $nlt(tr^*)$ is increased by 1.

2. if $pos < |RU\text{-}LP|$, tr^* should be inserted in position pos. The resources at this position and at its right are shifted by 1.

The Algorithm 1 repeats the above operation til form a recommended LP to meet the length requirement.

Algorithm 1 RU-POMDP Algorithm

procedure RU-POMDP
Input: σ, $MAX_{RU\text{-}LP}$, $his(tl)$, TM
Output: *RU-LP*
  $U\text{-}LP \leftarrow U\text{-}POMDP()$ // Use the unique POMDP
  $NLT \leftarrow NLT_{cal}(U\text{-}LP, his(tl))$ // compute personalised NLT vector
  $RU\text{-}LP \leftarrow U\text{-}LP$
  while $|RU\text{-}LP| < MAX_{RU\text{-}LP}$ **do**
    $tr^*, et \leftarrow argmin\ EarliestTime(tr, tl, nlt(tr), \sigma)$ // tr^* with minimal earliest time of no recall
    $pos \leftarrow ind(tr^*, et)$// the index of inserting tr^*
    $\beta_{tr} \leftarrow Trainsition_{tr}(tr^*, pos, TM, U\text{-}LP)$ // the avg transition probability of tr^* in position with left and right resource
    $\beta \leftarrow Trainsition(pos, TM)$// transition probability the of the *pos*
    if $\beta_{tr} > \beta$ **then**
      $Insert(RU\text{-}LP, tr^*, nlt(tr^*), pos)$

Figure 3.5 shows an example of resource insertion in *U-LP* (obtained from the first step of the algorithm) in the second step of RU-POMDP model. The given length $MAX_{RU\text{-}LP}$ is 6. While the length of *RU-LP* is lower than MAX_{RU-LP}, the algorithm traverses the entire *U-LP*. In the first loop, *argminEarliestTime* found that in *U-LP*, the candidate resource that should be added is b with $et = 2, pos = 4$. Then function $Insert(RU - LP, tr^*, nlt(tr^*), pos)$ is evaluated with the parameters $Insert(RU - LP, b, nlt(b) = 1, 4)$. The resource b should be added in the yellow box, after the resource f (case 2 of the previous paragraph). The $Insert()$ function updates $RU - LP$.

In the second loop, *argminEarliestTime* found that after the insertion of the candidate resource b, f should be added with $et = 3, pos = 6$, $Insert()$ function adds the candidate resource f in the same way. Here, it is inserted at the end of the path. After these two loop, the $RU - LP$ is satisfied with the $MAX_{RU-LP} = 6$, the algorithm operation is completed.

Compared with M-POMDP, the way the memory is managed is finer: we not only used the temporal gap, but also used a NLT to judge whether multiple review action should be performed. As this is managed in a post processing step, its impact on the complexity of the model remains limited. In addition, we can note that the number of possible candidate resources for the second step is reduced, which limits even more the complexity of this second step.

In this chapter, we proposed three POMDP base LPRSs.

First we proposed a basic POMDP LPRS. In this RS, we mainly presented how to define some concepts of POMDP in the field of education, among them the most important of which is how KL is defined in the state of a POMDP. Then we proposed a LPRS named M-POMDP, in this model we simply built the spaced repetition method into a POMDP. Due to the simple application of the spaced repetition method, this model still has some shortcomings. Based on this model, we propose a third LPRS model RU-POMDP. The contribution of the RU-POMDP is that we first built a model that combined the POMDP and learner's memory model together.

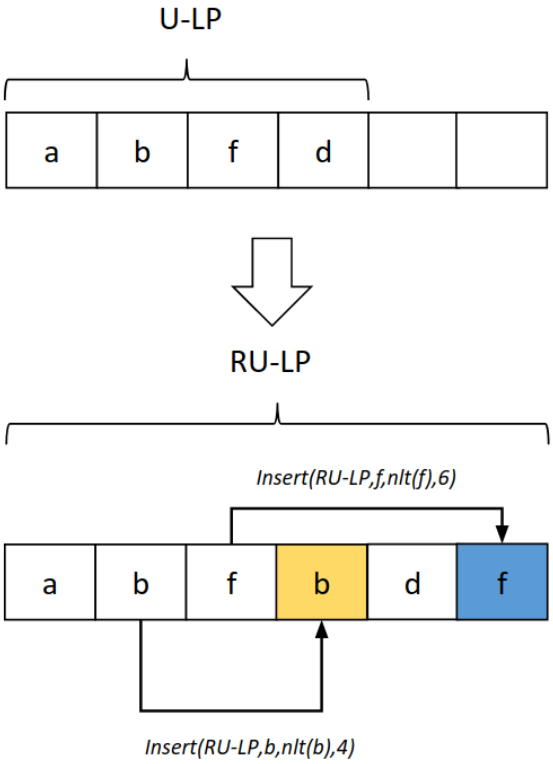


Figure 3.5: An example of insertion of resources by *RU-POMDP*

4

New Measures for Offline Evaluation of Learning Path Recommendations

In this chapter we aim at answering the second Research Question introduced in section 1.1: "How to define offline LPRS evaluation measures, i.e. measures that can evaluate the effectiveness of a recommended LP and that can contribute to a more general adoption of LPRS?"

The previous chapter has highlighted that the literature lacks well-adopted offline evaluation measures for LP recommendations and most of existing measures require additional information. Most literature evaluated their experiments differently, except for literature using traditional methods. Indeed, they require either expert knowledge [87], or information about learning resources: content, difficulty level, structure [18], or data/knowledge about learners such as learning ability, level of knowledge, etc., [24]. They are thus complex, even highly complex to implement and may not be adapted to many contexts, especially due to the requirement of additional data which may not be available. Besides, many of the measures proposed are inspired from traditional measures used for evaluating next step recommendations [143], [18], [18]. This limits their expressivity as they do not consider the sequential characteristics of path or learning recommendation. In addition, they do not consider the educational specificities. These limitations may partially explain the lack of an general LP offline evaluation measures.

Beyond these limits, in our point of view, the educational context faces a lack of universally applicable evaluation methods due to a lack of methodology and the lack of attention of researchers in the evaluation methods in education.

As a consequence based on the comprehensive analysis of the research characteristic of offline evaluation measures in education, in this chapter we aim to propose new measures that cope with the identified limitations. The proposed measures are designed to be easily implemented and thus widely adopted, whatever are the contexts, datasets, even the goals. Concretely, these measures are designed to rely only on learners' traces of activities (their LP) and their performance (results at exams or quiz for example), which are a subset of the data required by the measures used in the literature.
Particularly, no content information or difficulty level about learning resources or about learners is required for these evaluation measures, which is a choice in line with those made in the previous chapter. Indeed, many datasets do not have adequate descriptions of learning materials and descriptions of learners.

In the design of evaluation measures, we propose to focus on two elements already highlighted for the design of a LPRS.
First, a recommended LP should be **in line with the target learner's learning behavior**.

Second, a recommended LP should **improve the learner's level of knowledge**. The evaluation measures that we propose are designed for considering these elements.

Table 4.1 lists the main notations used throughout this chapter.

Notation	description
$\mathbf{D}$	the whole dataset
tl	target learner
$GT(tl)$	ground-truth learning path of the target learner
L	set of all learners
TL	set of all target learners
$level(tl)$	the performance level of the target learner
$LP_{start}(tl)$	starting learning path of a target learner tl i.e. history
$LP_{rec}(tl)$	recommended learning path of target learner tl
$LP_{end}(tl)$	learning path actually adopted by tl (after having performed $LP_{start}(tl)$)
$dist(a, b)$	a distance function between two paths a and b
$M(tl)$	set of mentor learners of the target learner tl
$m(tl)$	a mentor learner of a target learner, among the set of mentor learners, $m(tl) \in M(tl)$

Table 4.1: Notations used in this chapter

4.1 Offline Learning Path Evaluation and Problem Definition

In the frame of LP recommendation, an evaluation measure aims to estimate the accuracy of a LP recommended to a target learner.

The traditional offline evaluation process is shown in Figure 4.1. This process requires a dataset $\mathbf{D}$. $\mathbf{D}$ is split into two sets: training and test sets. Here, we choose to split the dataset at the learner level. A set of learners $L_{train} \in L$ is set as the training set, which contains the data of these learners. Concretely D_{train} is made up of these learners' complete LP $D_{train} = \cup_{l \in L_{train}} LP(l)$. The same for the test set $D_{test} = \cup_{l \in L_{test}} LP(l)$. With $L_{train} \cup L_{test} = L$.

The training set is used to train the LPRS. In our work, it is used to learn the policy of the POMDP. In the experiments conducted in chapter 5, we will also present how the training set is used for non MDP-based models.

The test set is used to evaluate the recommendation model. For each target learner $tl \in L_{test}$, $LP(tl)$ is split into two LPs: the starting LP ($LP_{start(tl)}$) and the ending LP ($LP_{end(tl)}$) with $LP(tl) = LP_{start(tl)} \cdot LP_{end(tl)}$. The profile of the target learner is evaluated on $LP_{start(tl)}$, that represents the learning behavior of tl. In our work, $LP_{start(tl)}$ is used to determine the initial state of the POMDP. The model recommends tl a LP ($LP_{rec}(tl)$). Evaluation measures are used to evaluate the accuracy of $LP_{rec}(tl)$, i.e. to what extent the recommended LP fits a ground truth LP ($GTLP(tl)$).

Recall that the evaluation for e-learning is different from other application domains. Indeed, in traditional evaluation settings (i.e. not in education), the evaluation compares the recommended path and the actually adopted path, as the system considers that what the user does is correct. For example, in e-commerce what a customer has decided to buy is considered as an adequate product for him [15]. In music, the playlist (sequence of songs) listened by the user is considered as an adequate and liked playlist [97]. However, in education the problem is different. Indeed, what the learner does (the LP adopted) is not always the good path for him/her. This is obvious for learners who get low grades at exams. This characteristics should be considered

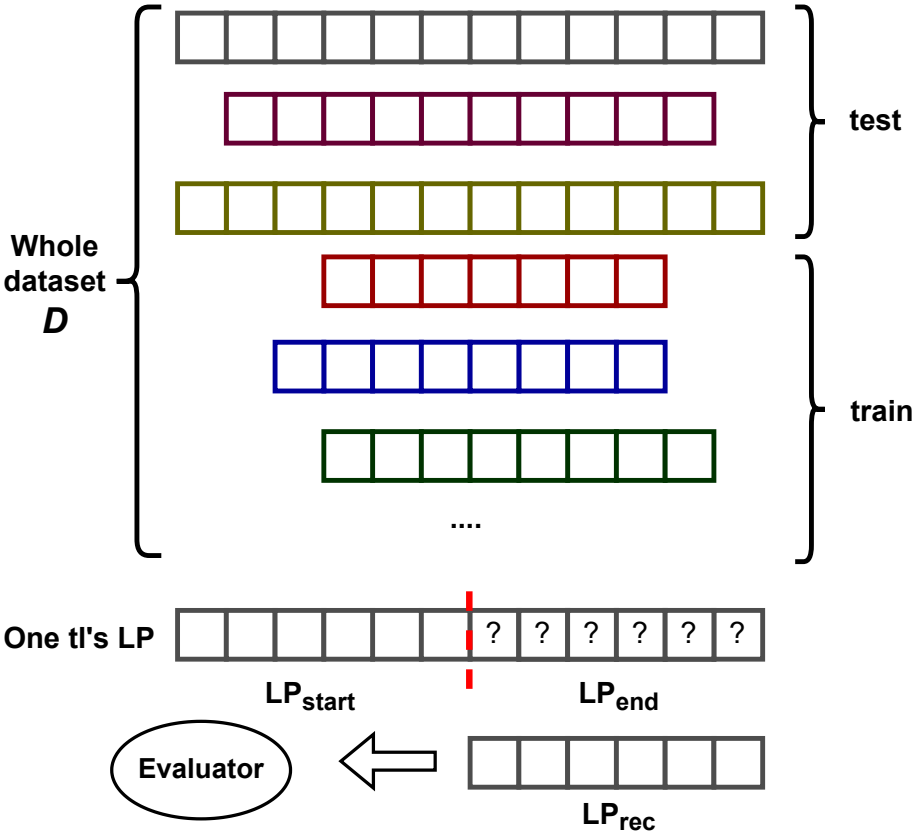


Figure 4.1: Evaluation procedure for one LP_{rec}.

when proposing new evaluation measures. Nevertheless, the literature does not consider this dimension.

We thus propose to define the LP evaluation problem as follows. A target learner tl is defined by two elements: (1) this sequence of resources (s)he has consulted so far, referred to as his/her starting LP ($LP_{start}(tl)$), (2) his/her estimated level of performance (knowledge level, recall that knowledge level is used for a resource or concept and level of performance is a single value for a learner), from the grades obtained for the evaluations he took so far ($level(tl)$).

The problem is thus twofold: how to determine the ground-truth LP $GTLP(tl)$ from the dataset ? and how to compare it to $LP_{rec}(tl)$? Below, we will introduce several evaluation measures that differ from each other in several ways: they rely on different definitions of the $GTLP(tl)$ and rely on different ways of comparing $GTLP(tl)$ and recommended $LP_{rec}(tl)$.

Given a target learner tl with his/her LP_{start} and LP_{rec}, we choose to identify $GTLP(tl)$ as the path adopted by a learner in the dataset. This learner acts as a mentor learner. In the field of offline evaluation, the concept of **mentor** is proposed for the first time.

4.2 New Offline Evaluation Measures

In this section we introduce four evaluation measures. They differ by the way the mentor learner(s) is/are chosen and the way the recommended LP is compared to the one of the mentor

learner(s), i.e., the ground truth LP.

The first two measures rely on the hypothesis that learners who have a high level of performance (learners who get top grades at exams, quizzes, etc.) have an adequate learning behavior, i.e. they adopt an adequate LP. These learners are thus considered as mentor learners and their LP is considered as the ground truth LP.

The third measure considers that top learners may not be adequate mentor learners of any learner, and considers that learners with a level slightly higher than the one of a target learner can be viewed as mentor learners.

The last measure is in line with the third one, it also considers learners with a higher level as mentors. It also consider top level learners (as in the two first measures), but not as mentor learners, rather as learners who adopt a LP that may not be recommended as it is too specific to be adopted.

Let us now focus in details on the measures we propose. Each of these measures is expressed for a given target learner tl, and can be also evaluated for a set of target learners by evaluating the average measure.

4.2.1 The Top learners Learning Path based Measure

The Top learners Learning Path based measure (TLLP) defines a learner m as a mentor learner if he/she has a high level of knowledge ($level(m)$) (i.e. he/she has top grades, he/she is a top learner), without considering any other aspects. As previously highlighted, the LPs of top learners explicitly or implicitly include good learning habits, or contain a sequence of resources that seem to contribute to achieve good grades. So, TLLP considers that the paths adopted by top learners lead to a high level of knowledge. TLLP evaluates to what extent the recommended LPs are in line with the paths adopted by top learners.

TLLP relies on a predefined set of possible mentor learners M, selected according to their grades. The set of mentors M is common to any target learner tl. This way to define M is opposed to the way the following other measures define mentor learners, that form personalized sets of mentor learners. In TLLP, $\forall tl_1, tl_2 \in TL, M(tl_1) = M(tl_2)$.

Based on this definition, and given the possibly large set of mentor learners, TLLP evaluates to which extent the recommended LP ($LP_{rec}(tl)$) reflects top learners' learning habits. Concretely, TLLP compares $LP_{end(m)}$ for all $m \in M$ with $LP_{rec}(tl)$. If the recommended LP is close to a LP adopted by a top learner, we can say that this recommended path leads to a high level of performance, or at least to an increase in the target learner's level of performance. In this case, the recommendation is accurate.

A subsequent question arises: how to choose the most adequate mentor learner(s) that best fit the target learner? As a simple evaluation measure, TLLP does not make any choice between mentor learners, but considers all of them. This is in line with the way the mentors are chosen: based on their grades only, not their similarity with target learners.

However, we do not exclude that some of the top learners have a high level background knowledge and that their LP does not reflect good learning habits, or at least adequate for any other learner. Therefore, TLLP does not consider that the recommended LP should fit all the LPs adopted by mentor learners, but that the recommended LP should fit some of them.

In details, TLLP compares the recommended LP of a target learner ($LP_{rec}(tl)$) with the ending LP of each mentor learner ($LP_{end}(m), m \in M$) and evaluates the ratio of mentor learners who have an ending LP close to the path recommended. TLLP is presented in Equation 4.1.

$$TLLP(tl) = \frac{\sum_{m=1}^{|M|} |dist(LP_{rec}(tl), LP_{end}(m)) < c|}{|M|} \tag{4.1}$$

where m is a mentor learner, $dist(L_1, L_2)$ is a distance between paths L_1 and L_2, $|M|$ is the total number of mentor learners, c is a maximal distance threshold and $dist(LP_{rec}(tl), LP_{end}(m)) < c$ equals to 1 if the condition holds, else 0. The larger TLLP, the more accurate the RS.

The distance measure is not specified here and any measure can be used. It can for example simply rely on the number of resources shared by both LP (as in traditional metrics [25]), or on any Edit Distance [26], as proposed in the literature.

In light of TLLP simplicity, we can highlight several obvious limitations. First, TLLP does not consider if the recommended LP matches the background knowledge of the target learner, i.e. the personalization degree is not measured. In addition, as for the traditional precision measure, the distance value between paths is not explicitly taken into account.

$TLLP$ may be not adequate for learners with a very low $level(tl)$ value. Indeed, it may not be pertinent to consider that the LP adopted by top learners is an adequate LP for other learners. Indeed, the large difference in their levels of knowledge and their learning ability, the mismatch between both learner profiles may be too large. Thus, the values of TLLP for learners with a low $level(tl)$ value may not make sense.

4.2.2 The Similar Learners Learning Path based Measure

The Similar Learners Learning Path based measure (SLLP) is designed to improve TLLP, by refining the definition of a mentor learner. In SLLP, a mentor learner m is defined by two criteria. First of all, a learner who adopts a starting LP ($LP_{start}(m)$) close to the one adopted by the target learner ($LP_{start}(tl)$) is a good candidate to be a mentor. Second, such a learner should be a successful learner and this is a requirement as candidate for a TLLP tutor.

At the opposite of TLLP, by exploiting the starting LP of learners (target and mentors), an emphasis is put on the learning behavior of the learners.

In SLLP, the set of mentor learners is personalized, i.e. each target learner tl has his/her own set of mentor learners $m(tl)$, which are defined as the learners who have a starting LP close to the one of the target learner, while being top learners.

Considering the way mentor learners are used, as for TLLP, SLLP uses the distance value between the recommended LP of the target learner ($LP_{rec}(tl)$) and the actual path adopted by his/her mentor learners ($LP_{end}(m)$). All mentor learners ($m(tl)$) are used to evaluate SLLP, to avoid the mismatch caused by a single mentor that could be not suitable for the target learner, and the distances are averaged.

$SLLP$ is presented in Equation 4.2.

$$SLLP(tl) = \frac{\sum_{m=1}^{|M(tl)|} dist(LP_{rec}(tl), LP_{end}(m))}{|m(tl)|} \tag{4.2}$$

Here again, any distance measure can be used: edit distance, cosine similarity (adapted to fit distance properties), etc. The smaller SLLP, the better the recommender system. At the opposite of TLLP, SLLP can be used for any learner, ranging from low-level to high-level.

4.2.3 Performance in Learning Path based Measure

The point of interest of PLP (Performance in Learning Path) is different from the ones of TLLP and SLLP. PLP aims at evaluating to what extent the recommended LP contributes to improve the level of knowledge of the target learner. The main focus element of PLP is thus the increase in the level of knowledge of learners. PLP does not consider at all the learning behavior of learners, at the opposite of SLLP.

As for the two first measures, PLP considers the ending LP of a mentor learner ($LP_{end}(m)$) as the ground-truth. However, PLP differs from these measures by the choice of the mentor learner. PLP requires that at least two temporally distant evaluations have been taken by learners, noted E_1 and E_2 (E_1 being taken before E_2). The distance between E_1 and E_2 should be long enough to contain a recommended LP. Given a target learner tl, PLP relies on the following assumptions:

- if there exists a learner m with a level of performance associated to E_1 ($level(m, E_1)$) similar to the one of tl ($level(tl, E_1)$)
- this learner should have a level of performance associated to E_2 ($level(m, E_2)$) higher than the one of tl ($level(tl, E_2)$), i.e., $level(m, E_1) = level(tl, E_1) \wedge level(m, E_2) > level(tl, E_2)$
- during the same time the LP of this learner and the LP of the target learner are similar

Indeed, $m(tl)$ is chosen as the learner that meets a trade off between a minimum distance in the levels of performance for E_1 and a possibly high distance in the levels of knowledge for E_2. This way to select a mentor is significantly different from the previous measures. A coherent assumption is made about the ending LP for both tl and m. In addition, PLP considers a unique mentor learner m, who is chosen as the learner with the highest increase in performance. Indeed, this means that the path adopted by m is more adequate than the one adopted by tl, as it leads to a higher increase in performance. If such a learner m exist, he/she can be considered as a mentor learner of the target learner tl.

About the comparison of both LPs, here again a distance is used and the closer the recommended path to tl ($LP_{rec}(tl)$) to the ending of m ($LP_{end}(m)$), the better the recommendation.

Notice that the choice of m, and the associated ground truth LP, is totally realistic as m corresponds to a learner who starts with a similar performance than tl and who reaches better performance, in the limit of an increase that has actually been performed. PLP is defined by Equation 4.3.

$$PLP(tl) = dist(LP_{rec}(tl), LP_{end}(m)) \tag{4.3}$$

The lower PLP, the better. At the opposite of TLLP, PLP can be used for any learner level, ranging from low-level to high-level learners. Notice that PLP can be adapted to manage a set of mentor learners, not a single learner. In this case, the measure can be expressed as in Equation 4.4.

$$PLP(tl) = \frac{\sum_{m=1}^{m=|M(tl)|} dist(LP_{rec}(tl), LP_{end}(m))}{|M(tl)|} \tag{4.4}$$

Comparing PLP and SLLP, they both use the distance between target learners and their mentor learner(s). They differ in the way their mentor learner is selected: SLLP focuses on the learning behavior of learners, and PLP focuses on their level of knowledge.

4.2.4 Difference in Learning Path based Measure

The main focus element of Difference in Learning Path based measure (DLP) is twofold: the increase in the level of learners and their learning behavior.

The DLP measure addresses the limitation highlighted in TLLP: it may not adequate to consider that the recommended LP of a very low level learner (who tends to be unable to complete assignments) should be close to the one adopted by high level learners. Intuitively, the

difference (or the distance, if we consider the distance measures previously mentioned) between both paths should be high.

DLP is in line with this intuition: for an under performing target learner, DLP evaluates to what extent the recommended LP of this learner is different from the ones adopted by top learners. Obviously, if we only compare a target learner with the top learner, this criterion cannot be used alone. So, to avoid that a non-sense recommended LP is considered as an adequate recommended LP (as it is only highly different from any path adopted by good learners), the recommended LP is also compared to the one of learners with a higher level than the one of the target learner, both the top learners and the higher level learners are considered as mentor learners.

For better understanding, here we give an example. Assume that all learners are divided into three levels by their performance level, i.e. learners are divided into three groups: top learners, average learners and promising learners. The recommended LP for promising learner should be different compared with the LPs adopted by top learners. Moreover, the difference between promising learner and average learner should less than the difference between the promising learner and the good learner.

Through this kind of comparison, DLP measure verifies the personalized recommendation effectiveness of the recommended path.

Given the above definition, in DLP we should choose two different kind of mentors. The learner $m_{top}(tl)$ is a learner with top grades (as in TLLP) who has a starting LP the closest to the one of tl. The learner $m(tl)$ is the learner with a level higher than the one of tl and with the closest starting LP to the one adopted by tl.

Equation 4.5 presents the exact way DLP is evaluated.

$$DLP(tl) = dist(LP_{end}(m_{top}(tl)), LP_{rec}(tl)) - dist(LP_{end}(m(tl)), LP_{rec}(tl)) \tag{4.5}$$

The higher DLP, the better. As for other measures, the distance used can be any distance measure, such as the Edit Distance.

DLP compensates for the limitations of TLLP and is specifically dedicated to low level learners.

4.3 Grouping Learners

To limit the complexity of the previously introduced measures, and to consider the uncertainty about the levels of learners, we propose to automatically assign each learner $l \in L$ a label that represents his/her level of performance $level(l)$. This level implicitly represents the effectiveness of his/her LP. The learners are divided into n levels, with n being determined by either the size of the data set or by an expert. Level 1 represents the learners with the highest learning performance (the top learners), and level n represents the learners with the lowest performance. For any level $k \leq n$, the level $k-1$ is higher than k.

Learners are thus grouped according to these levels.

We propose to refine the definition of the four measures according to these levels. Considering different levels for the learners and adapting the approach accordingly is not new. For example, [154] cluster learners according to their level of performance, participation and engagement with the course. Their division is not common. They need additional information such as for frequency of mouse clicks by learners while studying for engagement level.

For TLLP, we propose to define the set of mentor learners M as the set of learners in the group of learners of level 1.

For SSLP, given a target learner with level k, the mentor learners are chosen in the set of learners with a slightly higher level, concretely from group $k-1$. The exact set of mentor learners is then refined to keep only mentor learners with a starting path that is close to the one of the target learner.

For PLP, given a target learner of level k_1 for evaluation E_1, and level k_2 for evaluation E_2, the mentor learner m is chosen among the set of learners with $level(m, E_1) = k_1$ and $level(m, E_2) < k_2$. m is chosen as the learner with the highest $level(m, E_2)$.

For DLP, the set of top learners (top mentors) is taken from the group of learners with level 1 and the set of mentor learners is taken from the group of level $k-1$, considering that the level of the target learner is k.

4.4 Conclusion

The four measures introduced in this chapter are designed to be general and easy to implement measures, so that they can be used in most of the contexts. Indeed, what needs to be done is only to find the adequate mentors and apply the equation that mainly rely on a distance between paths. Even simpler, TLLP does not need any selecting of specific mentors, it relies on a general set of mentors. Besides, these measures only require traces of activities of learners and grades on evaluation resources (exams, quiz, etc.), which is a subset of the data required by many measures, especially in the educational domain. Indeed, they require neither specific descriptions of resources nor other descriptions are required. Only learners' history and a few quizzes or exams are needed.

A downside of the simplicity of some of these measures is that they do not fit all the learners' level, hence the need of using the measures in conjunction with each other to compensate for each other's shortcomings. The conjunction makes the aspects of these measures more comprehensive. For example, combining these measures can be a way to evaluate both the ability of a recommender to form a LP that is in line with the learning behavior of the learners and the increase in the knowledge level.

Besides, some of these measures do either use a single or a set of mentors. This choice is arbitrary and each of these measures can be simply updated to be in line with the other choice.

In the following chapter, we will evaluate the contributions introduced in this chapter and in the previous one.

5

Experiment and Analysis

This chapter is dedicated to the evaluation of the contributions introduced in Chapter 3 and Chapter 4. We conduct experiments to evaluate both the evaluation measures proposed and the recommendation models (M-POMDP and RU-POMDP) designed.

Considering the evaluation measures we designed, the experiments we conduct aim to achieve the following goals:

1. Compare the different evaluation measures we propose and analyze the advantages and disadvantages.
2. Combine the different evaluation measures to jointly analyze the efficiency of a recommended LP.

Considering the evaluation of the introduced LPRS (M-POMDP and RU-POMDP), the experiments are designed to answer the following questions:

1. Can a simple memory model (M-POMDP) help to recommend accurate LP?
2. Does RU-POMDP, which has reduced time and space complexity, performs better than M-POMDP?

Prior to conducting these experiments, I will present the datasets used and the experimental setup.

5.1 Datasets

To conduct a thorough analysis and evaluation of the two main contributions of this thesis, we choose to conduct the experiments on two real-world datasets, namely EOLE and EdNet that have different characteristics, and are presented below.

5.1.1 EOLE Dataset

The first dataset used is made up of the traces of interactions of learners with learning resources on their LMS. The learners are first-year university students who are enrolled in a Mathematics and Computer Science Bachelor program. This dataset has been collected in the frame of the PIA2 DUNE EOLE (Engagement to Open Education / Engagement pour Ouvrir L'Education) project, 2017-2021. This dataset will be named the EOLE dataset. The traces represent the interactions for one specific course: "algorithms and programming" from the Fall Semester in

2018, which is a core course of this bachelor program. The learners' LPs are made up of the temporal sequence of accessed resource IDs.

Diverse learning resources are available in this course: slides, exercises for lab sessions, quizzes, and final exams.
Each learning record represents a learner activity, and includes (1) learner ID, (2) timestamp of the interaction, (3) type of access (view, submit), (4) resource ID, (5) score (for quizzes and exam only).

Before the course starts, each learner's level of knowledge is assessed through an initial quiz. At the end of the semester, each learner takes the final exam. Besides, an optional mid-term quiz is proposed to learners when half of the course has been taught. Some of the learners do not take the initial or mid-term quiz, but each of them takes the final exam.

In this dataset, we identify two periods by time line:
The course period, where learners access resources after they are uploaded on the platform by the teacher. Some activities are imposed by the teacher (expert), other activities are optional and are mainly adopted on the learners' own initiative.
The review period, that stands between the last lecture/lab session and the exam, where all learning resources are available and learners review on their own initiative for the final exam.

The EOLE dataset will not only be used for assessing both the evaluation measures proposed, and recommendation models proposed.

To perform a detailed evaluations of our contributions, we propose to divide learners into $n = 3$ groups (as introduced in Chapter 4, based on their estimated level of knowledge. We propose to estimate this level of knowledge by computing a weighted average of the quizzes and exam scores. The three groups are called Good Learners (GL), Average Learners (AL) and Promising Learners (PL). The split criterion between groups has been determined by the teacher of the course, on the basis of her personal experience. As a result, Good Learners (GL) represent the top 30% learners, Average Learners (AL) represent the 30% following learners, and the remaining learners (30%) are Promising Learners (PL).

We removed the last 10% of learners because their learning paths were severely missing, i.e. too short to do any operation. As the number of learners in each group is quite the same, we can consider that the dataset is well-balanced and that learners' level will not bias the models trained and the evaluation.

Table 5.1 presents an overview of the EOLE dataset. First of all, we would like to highlight that this dataset is not a large dataset (104 students, 39 resources, about 4,500 records), which will be considered in the experiments.
We can also notice that learners' LPs are longer during the course period than the LPs during the review period, about twice more for each group. This can be due to the duration of the periods: the course period is 1.5 times longer than the review period. We will use the data from the review period for evaluating the recommended LP for two reasons:
1. During the review period, all resources are open to learners, learners are free to learn whereas during the course period, even if learners can review, learners' LPs are still consistent with the availability moment of resources.
2. The review period is close to the exam, and the engagement of the learners is stronger than the course period, which means that it is easier for us to find a learning path that suits a learner's learning characteristics.

The general goal of an LP is to improve the knowledge level of learners. Intuitively, the only thing that can verify the improvement of learners' knowledge level is the improvement of exam/quiz scores.

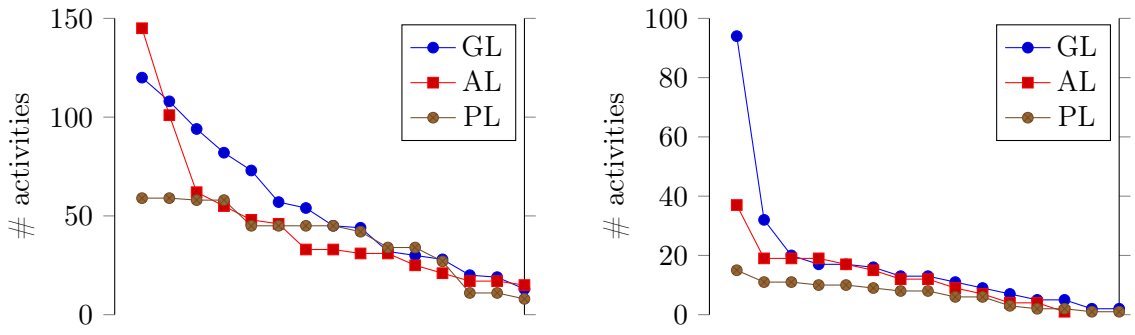
In addition, good learners tend to have a larger amount of activity than other learners.

Table 5.1: Overview of EOLE dataset

Indicators		Values
	Number of learners	104
	Number of resources	39
Course period	Number of learning records	3,279
	Median number of learners' learning records	46
	GL average number of learning records	60
	AL average number of learning records	49
	PL average number of learning records	35
Review period	Number of learning records	1386
	Median number of learners' learning records	14
	GL average number of learning records	17
	AL average number of learning records	13
	PL average number of learning records	11
	repetition rate	0.30

However, w.r.t. their initial amount of activity, they do not tend to review more, than promising learners. To refine this analysis, Figure 5.1, displays the number of activities (learning records) for each of the three groups of learners. Learners are sorted in descending order of their number of activities. This figure confirms that the number of activities of learners differs between groups. Although we can see that the learners who work more during the course period are good learners, we cannot say that learners that have a low activity are promising learners. GL and AL tend to have a similar number of activities during the review period. Except for a small number of learners who have really high activity.

The average repetition rate in EOLE is 30% ($GL, AL, PL = 0.42, 0.31, 0.18$), which means that when a learner studies 3 resources, one of them is a repetition in both review and course period.



(a) learners' activities during the course period

(b) learners' activities during the review period

Figure 5.1: EOLE dataset: learners' activities by group during both course and review periods

5.1.2 EdNet Dataset

In the online education domain, since users' data privacy is protected, it is difficult to find two completely similar public datasets. In this situation the choice of the second dataset is limited.

In summary, we try to choose a dataset that can be cut out to meet the requirements. The second dataset we chose to use is a part of the EdNet dataset [155], which is a large-scale educational dataset of over 131 million interactions collected from Santa[6], which prepares South Korean learners for the Test of English for International Communication (TOEIC). Each learners' LP is made up of the sequence of id of all accessed resources, as in the EOLE dataset.

In EdNet, each learning record also includes a learner's ID, the consumed learning resources (quizzes or lectures), the responses, and timestamp in milliseconds between this learner interaction and the first event completion from that learner. Similar to the EOLE dataset, a learning record represents a learner activity and includes (1) learner ID, (2) timestamp of the interaction (3) type of access (quiz, lecture), (4) resource ID (5) score (for quizzes only).

Considering the complexity of POMDP, the number of resources directly affects the number of states, so the number of resources should be within a controllable range. If the number of resources is exactly the same as in the EOLE dataset, the LPs in the database will be too short. Here, under the premise of forming an effective LP, we have selected the number of resources that better fit our model. Under this background, the dataset we use in our experiments is a subset of EdNet-KT1, where we consider the 100 most popular lectures interactions of 5,000 learners and the related quizzes. Note that the quizzes in EdNet are not considered as a part of learning resources in our dataset here since the rear learners redo the same quizzes, but in the EOLE dataset, the quizzes are similar to a resource, i.e. there are a lot of learners going to re-study their quizzes in EOLE dataset.

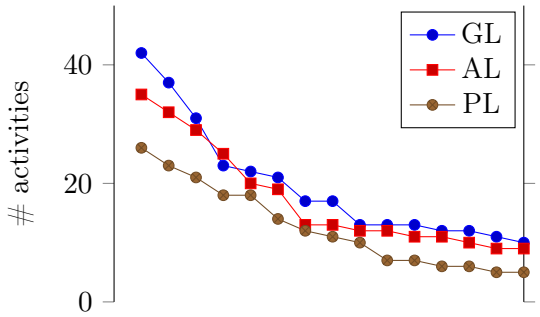
The Table 5.2 contains statistics of the EdNet dataset: number of learners, resources, records, etc. We also added some statistics of the EOLE data set to compare them together.

The EdNet dataset is quite bigger than EOLE dataset, due to the larger number of learners. If we take both the course and review period of EOLE into consideration, in the EdNet dataset the median of the LP (when discarding quizzes) is three times smaller than in the course period of EOLE. If we only take the review period of EOLE into consideration, the two median lengths are almost the same. In addition, the repetition rate in EdNet is a little bit smaller than in EOLE (22%, with $GL, AL, PL = 0.3, 0.23, 0.16$ respectively). We would like to precise that in EdNet, we aim to recommend only lectures, not quizzes.

[6] https://aitutorsanta.com/

Table 5.2: Overview of EdNet dataset

Statistics	EdNet dataset
Number of learners	5,000
Number of resources (lecture only)	100
Number of evaluation resources	319
Number of learning records	1,285,578
Median length of all LP	15
Repetition rate	0.22
median Number of learning records	14
GL average LP length	18/17
AL average LP length	14/13
PL average LP length	13/11
repetition rate	0.22



(a) learners' activities during the review period

Figure 5.2: learners' activities by group in EdNet dataset

For the EdNet dataset, even if it has more learners we still divide learners into $n = 3$ groups as we did in the EOLE dataset: GL, AL, PL, which is for the convenience of comparison with the experimental results of the EOLE dataset. Figure 5.2 represents the LP for the EdNet dataset. For consistency with Figure 5.1, each point in Figure 5.2 represents 100 learners here. We can find that the LP distribution here is almost the same as the one of the review period in Figure 5.2.

5.2 Experiments around New Evaluation Measures

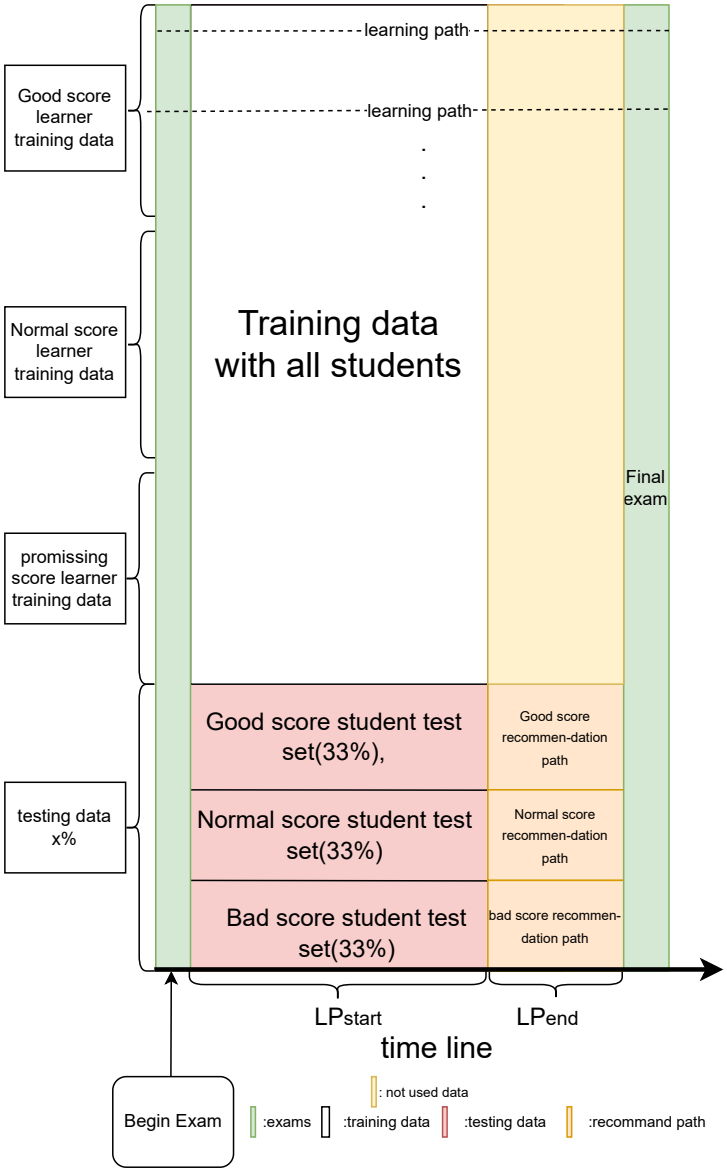
5.2.1 Experimental Setup

For the experiments, the dataset is split into training and test sets, as shown in Figure 5.3.

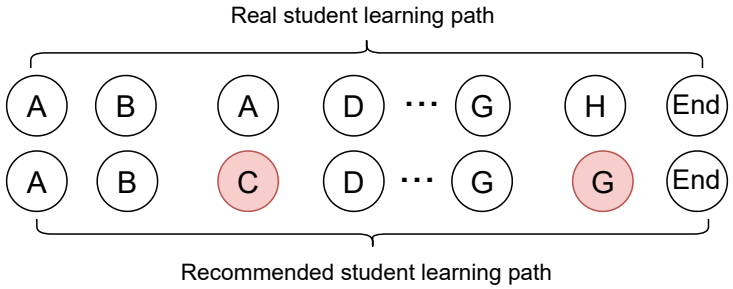
For the experimental setup, we start by explaining the way we propose to arrange the dataset into training and test sets.

Data Arrangement

To learn the recommendation model, we propose to perform cross-validation, which is a commonly used protocol [40]. It is a statistically practical method that divides a given dataset into smaller sub datasets. The principle of cross-validation is to use part of data which has not been



a) Data arrangement in evaluation



b) Comparation between GT and recomandation

Figure 5.3: Evaluation Setup

used to train the model to evaluate the performance of the model. The model is then trained on a subset or several subsets (the training data in Figure 5.3), while the remaining subsets are used for subsequent validation or evaluation of the model trained (the test data in Figure 5.3). The process is run several times till each element belongs to the test set once.

There are a number of validation methods, among which the famous following ones:
1. The exhaustive validation randomly divides the dataset into two partitions. For example, a cross-validation that uses 80% data as the training set and 20% as the test set[156], [40].
2. The k-fold cross-validation divides the dataset into k sets of similarly sized mutually exclusive subsets, and one of the k copies is taken as the test set, while the remaining $k-1$ sets are used as the training set to ensure the consistency of data distribution in each subset. A k iterations are performed so that each of the k sets is used as the test set in turn.
3. The leaving one out cross-validation is a special case in k-fold validation, where k is equal to the number of instances in the dataset. The test set is made up of one instance and the training is made up of all the instances except the one used for the test. leaving one out is used when the dataset is not large.

Since the EOLE dataset is not that big in terms of the number of learners, the number of resources, and the number of activities (we say it is a medium dataset), we propose to adopt the leaving one out cross-validation method [157]: One learner (and his/her associated activities, whether the course of review period) is used as a test (called the target learner) and the remaining learners form the training set. The traces of activities of this target learner are split into two, according to the course and review periods. The traces from the course period will be used as historical data by the recommendation model and computed as the recommended learning path. In the testing of our experiments, the first 50% form the elements which help to determine the initial state of the model. The last 50% will be covered as the path to be recommended. In Figure 5.3, the $x\% = 1$ learner for each validation experiment.

Considering the EdNet dataset, which is big enough, we decided to use a k-fold cross-validation: the dataset is divided into $k = 5$ subsets. For each fold, the training set contains 80% of the dataset (4 subsets) and the test set contains 20% (1 subset.

In order to better compare the effects of new evaluation measures and POMDP based RSs on learners in different performance groups, the subsets are randomly selected, premised on that the proportions for the three different groups are the same.

Training Model

For each fold or each run, the recommendation model is trained on the training dataset.

As mentioned in the literature [24], the length of the recommended LP is required by MDP and POMDP methods and the length should be set before forming a recommended LP. In accordance with the literature [24], it will be set as the median length of the learners' learning paths in the review period. The length of the recommended LP is thus fixed to 7 (see Table 5.1) for the EOLE dataset and 8 for the EdNet dataset, separately.

To form recommendations, we rely on two well-adopted LP recommendation models in the literature: Sequential Pattern Mining (SPM) [104, 158] and Markov Chains (MC) [159], presented in chapter 2. Similarly to [158], the LP of the target learner is formed by exploiting the patterns and probabilities from the two models separately, that fit the target learner's traces during the course period. "Fit" is here equality of the traces.

Recall that the goal of the evaluation conducted here is not to identify the best recommendation algorithm, but to study the evaluation measures. That is the reason why we did not choose the most evolved recommendation algorithms, but simple algorithms that we can thoroughly

understand.

For both of our two algorithms, Since the data rules in the course period and the review period are quite different and both the MC and SPM are easy models, here we only used the data in the review period to train the model. In order to ensure the training set is large enough and to use a data group that is as similar as possible to train the model, for the target learners of GL and AL, we use GL and AL for training; for the target learners of the PL group, we use the learner data of AL and PL for training.

As introduced in Chapter 2, the MC algorithm considers the recommendation as a decision making problem, and forms the recommended LP by exploiting Markovian properties [160]. In the following experiments, we propose that the recommended LPs are formed by a naive optimal policy calculated by MC [159]. Note that here we used a 1-*order* MC, i.e. the length of history that we used is 1.

For the SPM we used PrefixSpan SPM algorithm to form a recommendation sets as said in [17]. For a target learner, the generation of LP of the PM algorithm is based on the previous learning resources and the algorithm adds the most probable resource in the next. This cycle is repeated until the length of the recommended LP reaches the requirement.

As mainly proposed by the literature [26], the distance measure we use is Edit Distance.

5.2.2 Experimental Results and Analysis

As previously mentioned, to perform a detailed analysis, the set of target learners is split into the three previously introduced groups (GL, AL, PL) and the four measures introduced in Chapter 4 are evaluated for each group of learners. Table 5.3 and Table 5.4 respectively present the associated values for each of the four measures on the EOLE dataset and EdNet dataset.

Recall that both MC and SPM recommend LPs with the explicit purpose of reaching success in exams, so we expect that the recommended LP is very similar to the ones adopted by higher-level learners. This is especially true for TLLP, SLLP, and PLP measures.

Evaluation Measure	Learner Group	Recommendation Algorithm	
		MC	SPM
TLLP	PL	0.5	0.4
TLLP	AL	0.6	0.5
TLLP	GL	**0.7**	**0.6**
SLLP	PL	**6.9**	6.8
SLLP	AL	7.3	**5.4**
SLLP	GL	7.9	5.8
PLP	PL	**5.4**	**5.7**
PLP	AL	5.6	6.3
PLP	GL	6.8	7.9
DLP	PL	3.5	1.4

Table 5.3: Evaluation measures for each recommendation algorithm and each group of learners on EOLE dataset

Evaluation Measure	Learner Group	Recommendation Algorithm	
		MC	SPM
TLLP	PL	0.2	0.2
TLLP	AL	0.3	0.2
TLLP	GL	**0.4**	**0.4**
SLLP	PL	**7.2**	7.5
SLLP	AL	7.5	**7.0**
SLLP	GL	7.6	7.1
PLP	PL	**6.3**	**6.9**
PLP	AL	6.5	7.1
PLP	GL	6.8	7.2
DLP	PL	4.0	3.2

Table 5.4: Evaluation measures for each recommendation algorithm and each group of learners on EdNet dataset

TLLP

Recall that TLLP is designed to measure how many recommended LPs are close to the LP adopted by GL learners, i.e. all GL group learners are regarded as mentoring learners. TLLP does not perform any complex selection of mentor learners, the proportion of recommended LPs with similar tutors is a result of TLLP.

In the experiments, the threshold c in TLLP is set to 30%. For both EOLE and EdNet datasets, it is expected that the values for AL and GL are higher than for PL, for both recommendation algorithms. First of all, we can see that it is actually the case: for AL and GL, both algorithms tend to recommend LPs adopted by at least one top (good) learner. Indeed, for EOLE the highest TLLP value is obtained for the GL group (0.7 for MC and 0.6 for SPM), which means that from 60% to 70% of good learners receive an LP that is an LP close to the one adopted by good learners in the dataset. In EdNet (Table 5.4), even if GL also has the highest TLLP values, they are lower: 40% of good learners receive similar LPs of the mentor learners. This can be explained by the fact that the EdNet dataset's optional resources are 2.5 times that of the EOLE dataset, which leads to a decrease in the accuracy of recommended paths. We expected that this value would be higher. The fact that the TLLP for GL is not that high may be due to the fact the training set is made up of traces of all learners' levels and that both MC and SPM do not consider the learners' level during training. So, AL and PL influence the recommended LP. After this, we performed a simple additional experiment conducted on GL only (for train and test). This experiment shows a small improvement (TLLP$>$ 0.8 for EOLE and TLLP$>$0.6 for EdNet), which confirms the negative impact of promising learners. Notice that the values for PL are the lowest ones. As mentioned in section 4.2, this value may have no sense as recommending a promising learner a path adopted by good learners may not make sense. This may tend to highlight the fact that MC or SPM probably considers this fact and recommend more adequate paths. However, the value of TLLP associated with PL is still relatively high, even higher than expected, when comparing it to the values of AL and GL. We were expecting a significantly lower value for PL. It may be that our RSs recommend recommended LPs that are

closer to AL when learners in the PL group are the target learners. This is especially noticeable for the EdNet dataset.

Through comparing SPM and MC, we can see that for the EOLE dataset SPM has a performance lower than MC, whatever is the group of learners considered; for EdNet the results of SPM are also always less or equal to the MC, but compared to the EOLE dataset, the gap is not that big. This was expected as SPM may suffer from over-fitting in case of lack of data, which is the case of the dataset used. As in EdNet, we have enough data, the problem of lack of data has been improved, so the results between SPM and MC became closer.

This first set of experiments tends to confirm that TLLP is an interesting measure. It highlights differences between both algorithms and shows that MC tends to recommend LPs close to those adopted by good learners, especially for good learners and average learners. The experiments also validated that the TLLP has an outstanding performance for the GL target learners.

SLLP

Let us now focus on the SLLP measure that considers learners' learning behavior and evaluates to what extent the recommended LP of a target learner is close to the one adopted by the learners with a similar learning behavior and who also succeed in their exam (in EOLE the score of the final exam, and in the EdNet dataset the average score of related quizzes), i.e. learners from the GL and AL group. Note that the AL group learners could also be chosen as mentors for PL group learners. SLLP is evaluated as the average distance of the recommended LP and the one adopted by the mentor learners. So, the lower SLLP, the better the recommended LP.

Before using the SLLP measure, we have certain expectations for the results of the experiment:
First, the results on the EOLE dataset will be better than the EdNet dataset;
Second, for both of these two datasets, the results of the AL group should be better than the GL and maybe better than the PL group target learners.

Through our experiments, we have the following validations.

First, from comparing the results of Tables 5.3 and 5.4 we can see that the results of the EOLE dataset do better than the results of the EdNet dataset.

Further, SPM recommends LPs that better fit the learners' learning behavior for both EOLE and EdNet datasets (except for the PL group for the EdNet dataset). The above results can be analyzed from different aspects of the two datasets: First, the EOLE dataset is collected from university courses, which is more coherent, and the college students admitted as screened, in the three groups Their performances are not very different; secondly, for the EdNet dataset, which is collected from a language learning platform, the distribution of learners is very wide, and language learning will not have as strong coherence as university courses.

Secondly, for both the results of the EOLE and EdNet datasets, the best results appear in the target learners of the AL group and the PL group. This is in line with our expectations because for the target learners of the PL group and the AL group, their training sets use better learners, so more suitable mentors can be found, while the training set of the GL group contains more than the target learners. of poor learners.

About the PL group in the EdNet dataset, who get a worse result, it may also be explained by the specificity of the dataset: EOLE is for university students whereas EdNet is for everyone. The students in the university have been screened to a certain extent, and the level is relatively

close whereas PL learners in EdNet learn more infrequently and irregularly. This leads to the difficulty of finding a suitable mentor for a PL learner in the EdNet dataset.

Finally, we analyze the performance of MC and SPM algorithms. The LP recommended by MC is based on conditional probability, the improvement of the learner's LP is smoother, so the recommended path is more consistent with the LP of the selected mentor. Second, since the learning of promising learners is badly performed, SPM has difficulty finding the right recommended LP. During the same time, recommended LPs of SPM are more suitable for higher-level learners as we analyzed before.

Other than that, the designed SLLP is validated that the experiment results are biased by the datasets used. From the SPM for EdNet dataset as shown in Table 5.4 we can see that SLLP does not decrease as the learner level raises. This is in line with our expectations, and its phenomenon can be explained as follows: The length for the recommended LP is fixed. Meanwhile, for both the EOLE and EdNet datasets in Figure 5.1 and Figure 5.2 we can see that as target learner's performance increases, the LPs become longer.

Through these experiments, we confirm that SLLP is an interesting measure, that shows the ability of the algorithms to recommend LP that are in accordance with learners' learning path and conduct to a success. Here also, a difference between both algorithms is highlighted.

PLP

The PLP measure evaluates to what extent a recommendation algorithm contributes to the increase in the level of knowledge of learners while respecting their learning behavior. Recall that, the lower PLP, the better the recommended LP. As for SLLP, this information is not explicitly considered by any of the two recommendation algorithms used.

Before the experiments, we expect that the increase in level should be higher for low-level learners (i.e. $PLP_{PL} < PLP_{GL}$ and $< PLP_{AL} < PLP_{GL}$), as it is difficult to improve the level of an already high performing learner. For both the MC and SPM algorithms on both datasets, the experiment results are exactly as we expected. This experimental result validated the usability of the PLP method for PL target learners.

Through comparing the results in Table 5.3 and 5.4, we can see that, at the opposite of SLLP, MC is the most adequate recommendation algorithm with regard to PLP: the LPs recommended by MC are closer to the LP_{end} adopted by similar learners who increase more their level. For both algorithms, promising learners tend to get recommendations that actually correspond to an increase in their level, which can be justified by the fact that the training set is mainly made up of traces of activity of learners that have better grades. Compared with AL and PL, the values associated with good learners are lower, which was expected. Good learners tend to not get recommendations that increase their level.

Compared with the previous two measures, the results of PLP for the EdNet dataset are better. Since the choice of mentors for PLP is more strict, this may be explained by the fact that in EdNet, there are more candidate mentor learners for a target learner.

We conducted an additional thorough study, that has confirmed that promising learners negatively influence the recommendations made to good learners for both MC and SPM. This limit will be tackled for POMDP-based models, which will be presented in the following section.

DLP

DLP is radically different from the other three measures proposed, it is dedicated to promising learners only. DLP identifies to what extent the recommendation algorithms tend to recommend promising learners, some LPs that are more different from the ones adopted by GL group learners than the ones adopted by average learners. In other terms to what extent the recommendations are closer to the path adopted by average learners than by GL group learners. As previously mentioned, all groups are almost equally represented in the training dataset, which guarantees that there is no bias due to the higher influence of a group in the model.

For the DLP measure, if the result is positive, it means that the recommended LP for the target learners in the PL group is closer to the AL, otherwise, it is closer to the GL. We expect the resulting values to be all positive. For both algorithms on both datasets, the DLP value is positive, it confirms that the paths recommended to PL are closer to those adopted by AL, which is an interesting finding. Among the results, the recommended LPs of MC are all higher than SPM, which means that the LP recommended by MC is smoothly in the improvement of level. This is also confirmed by the results of SLLP. For both datasets, the results of EdNet are larger, which is also in line with our expectations: the EdNet has a larger set of resources (about 2.5 times larger than EOLE), and it is expected that the recommended paths are more different.

As mentioned in section 4.2, promising learners should not receive learning paths that are too similar to those adopted by GL group learners. In addition, MC has the largest value, which shows that the recommendations proposed by MC to promising learners are highly different from the ones adopted by GL group learners, so they may fit the target learners better.

5.2.3 Conclusion about Evaluation Measures

Throughout these experiments, we confirm the usefulness of the four measures proposed. These measures contribute to highlighting some specificities of each of the recommendation algorithms studied. As mentioned previously, the recommendation algorithms used are simple algorithms (they are not part of the most performing algorithms) and are not dedicated to the educational context, but we can analyze their recommendations and evaluate the relevance of the measures. Based on the analysis of the measures proposed, we can say that MC tends to recommend paths that are close to the ones adopted by average learners, and that MC recommends LPs that increase the level of knowledge of learners. As for SPM, we can conclude that it is more adequate to recommend LPs that purely fit learners' learning behavior.
Let us notice that if only one recommendation algorithm is used, the values of the measures, especially SLLP and PLP can be compared to a baseline that represents the average distance between the recommended LP and the actual ending LP of the target learners. And the TLLP and DLP have an outstanding evaluation of the GL target learners and PL target learners respectively.

5.3 Evaluation of POMDP-based LP-RS

5.3.1 Experimental Setup and Implementation Details

In this section, we also use both the EOLE and EdNet datasets for our experiments. The experimental protocol used to evaluate POMDP-based recommendation algorithms is the same as the one presented in Figure 5.3. As in the previous section, we also propose to adopt the leaving one out cross-validation method on the EOLE dataset and the k-fold cross-validation for the EdNet dataset.

For both of datasets, learners are divided into three groups, based on their level of knowledge. The three groups are also referred to as GL group, AL group, and PL group. In the experiments, the performance of each group will be studied separately.

Similar to the parameter settings of [24], the average length of the recommended LPs is set to half of the median length of all the learning paths: 7 for EOLE and 8 for the EdNet dataset.

About the parameters of POMDP, the discount factor γ is set to $\gamma = 0.9$. Following [79], MAX_{NLT} is set to 3 and the possible knowledge level is set to $K = 3$. As described in [33], the threshold used for recall is set to $\sigma = 50\%$. The SARSOP solver [161], which is one of the fastest POMDP solvers as far as we know, is used for all the models studied.

An example of a policy that recommends a length 3 recommended LP calculated by SARSOP is shown in Figure 5.4.

The generation of a POMDP policy in this figure is exactly followed the Figure 2.3 in Chapter2.

The Figure 5.4 includes 5 layers. Each state in each layer includes the last visited resource list, the knowledge level list and the NLT list. From the start state, the action 30 is taken (resource 30). The observation of knowledge level 2 is got with probability 1. Then it goes to another state and the only possible action points to learn resource 3. For the third layer, the next most probable state is $s(7)$ where the biggest next state contains knowledge level $KL = 2$ with probability 0.308 and the transition possibility 0.88 to resource 7. For the fourth layer, the most probable state is $s(27)$ under the action $A(31)$. The knowledge level is still 2. Then it goes to the final layer with the possible resource is 0, and a length 4 LP is formed. The LP, including start and end resource is $< r_{30}, r_3, r_{31}, r_0, r_0 >$.

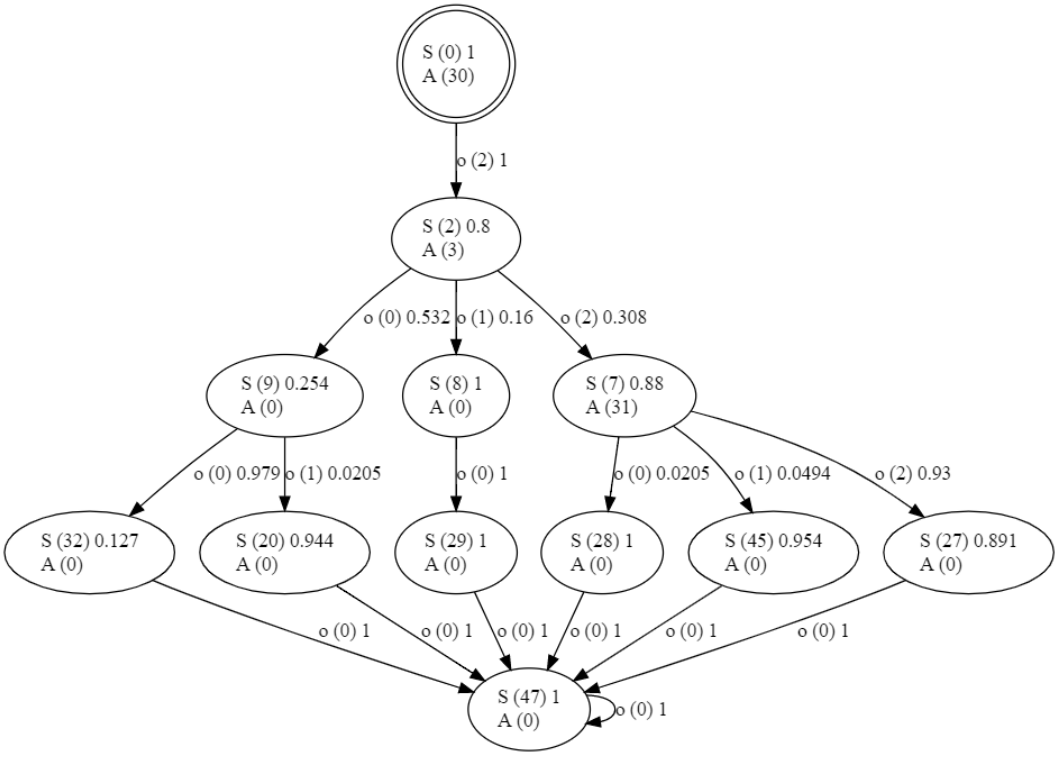


Figure 5.4: The recommended LP of SARSOP

5.3.2 Evaluation Metrics

We propose to conduct an offline evaluation of the accuracy of the recommended LPs based on the EOLE dataset and the EdNet dataset. We divide the evaluation metrics into two parts. We firstly use two traditional evaluation methods, precision and recall to evaluate our experiment results. Note that we modified these two measures to fit the sequential experimental context. Then, we use the new offline evaluation measures as we proposed in Chapter 4 to evaluate our experiment results.

Precision and recall are two of the most highly popular measures used in RS domains. In our experiment, one main question that is that under the context that traditional methods can only evaluate individual recommendations, how to modify the traditional metrics? Our adaptation relies on the identification of the Longest Common Sub-sequence LCS, adapted to evaluate sequential data. Our customized precision and recall are defined in Equations 5.1 and 5.2 as follows:

$$Precision = \frac{|LCS(LP_{rec}, GT)|}{|RLP|}, \tag{5.1}$$

$$Recall = \frac{|LCS(LP_{rec}, GT)|}{|GT|}, \tag{5.2}$$

where LP_{rec} is the recommended LP and GT is the ground truth LP. Note that here $|\ |$ is the length of the sequence rather than the cardinality. When evaluating with precision and recall, for the group of AL and GL, we choose the target learners as their own mentors. For PL, since their original LP may be not good, we chose themselves and learners with the most similar LP_{start} in group AL as the mentors.

The following sections are dedicated to the evaluation and analysis of the precision, recall and new measures of our LP recommendation models that we have introduced in Chapter 3. To thoroughly evaluate the impact of the post-hoc repetition process in RU-POMDP, U-POMDP is also analyzed. The baseline model, simply called basic POMDP, is a model close to the one proposed by [86].

5.3.3 Evaluating Precision and Recall for POMDP based RS

Let us first consider the experiment results of the EOLE dataset and the EdNet dataset with precision and recall measures, as presented in Table 5.5 and Table 5.6.

Recall that M-POMDP manages learners' memory strength the state of POMDP. The learners' memory strength is mainly presented as a simple count of the number of times a resource has been accessed.

First of all, we can see that M-POMDP improves the POMDP baseline in terms of precision and recall, for all the learner groups and both datasets.

For GL on the EOLE dataset, the improvement of our M-POMDP model is 44% for precision and 15% for recall, which is high and significant. For GL on the EdNet dataset, precision gets a huge boost of 93% and for the recall is also a high increase of 25%. For AL on EOLE dataset, this increase is lower for the precision: 19%, but the equivalent for recall: 17%. For AL on EdNet dataset, the increase of precision and recall are both 23%.

These values are in line with what we expected and confirm that M-POMDP has the capacity to recommend a more suitable LP to target learners.

The differences of these improvements with different datasets and different groups of learners can be explained by several elements:

Measures	Method	GL (self)	AL (self)	PL (self)	PL (hi-lvl)
Precision	POMDP	0.41	0.36	0.24	0.45
	M-POMDP	0.59	0.43	0.30	0.66
	U-POMDP	0.67	0.78	0.33	0.60
	RU-POMDP	0.61	0.50	0.35	0.53
Recall	POMDP	0.39	0.40	0.30	0.33
	M-POMDP	0.45	0.47	0.35	0.49
	U-POMDP	0.22	0.24	0.29	0.36
	RU-POMDP	0.54	0.49	0.48	0.55

Table 5.5: Precision and recall for EOLE dataset

Measures	Method	GL (self)	AL (self)	PL (self)	PL (hi-lvl)
Precision	POMDP	0.14	0.31	0.17	0.22
	M-POMDP	0.27	0.38	0.22	0.30
	U-POMDP	0.28	0.38	0.30	0.29
	RU-POMDP	0.38	0.40	0.33	0.35
Recall	POMDP	0.28	0.26	0.15	0.19
	M-POMDP	0.35	0.33	0.19	0.26
	U-POMDP	0.17	0.18	0.15	0.17
	RU-POMDP	0.40	0.35	0.20	0.30

Table 5.6: Precision and recall for EdNet dataset

First, recall that the length of recommended LP has to be fixed ahead of the recommendation process. This leads to that the average length of the recommended LP is not all equal to the one of GT for target learner. Generally speaking, GL has a longer LP and the length of the AL also fluctuates around the length of the recommended LP. This may be the reason why the increase of AL in the recall is larger than that of GL. This is also the reason why the increase in GL in precision is greater than AL.
Second, the reason comes from the choice of the mentor learners. Recall that the mentors are target learners themselves. Indeed, as for all the learners, the recommended path contributes to a high reward within their performance level. So AL tends to adopt a path that corresponds to a medium grade and GL tends to adopt a high-grade LP. This leads to the results that the recommended LP is closer to the LP of GL then the corresponding GL results will be better.
Third, the increase for the EdNet dataset is higher than the EOLE dataset. The reason is that the number of resources that can be recommended in the EdNet dataset is 2.5 times that of EOLE, which leads to low accuracy of the POMDP recommendation results.

Notice that the running time of M-POMDP is almost the same compared to POMDP, whereas the cost of storage memory increased about 5%.
It is worth noting that the experimental results obtained by our M-POMDP are surprisingly improved on both datasets by comparing with POMDP. This tends to confirm that the M-POMDP model has certain general applicability. We can thus first conclude that managing the learners' memory as a simple count in the state allows for a higher recommended LP accuracy, with no impact on the complexity.

Let us now focus on the RU-POMDP RS model. In the following, we compare the experimental results of three models: M-POMDP, U-POMDP, and RU-POMDP. For *U-LP*, the path formed by U-POMDP is made up of an ordered list of unique resources. From our experiments, we get that the average length of *U-LP* is 4. Given that the average length of the two datasets is 7 and 8 respectively, both paths cannot be compared directly. From the results of precision, we find that the U-POMDP is almost the best for both the EOLE and EdNet datasets. This demonstrates the effectiveness of our U-POMDP model. This is especially true for the EOLE dataset. When looking at the recall of the *U-LP*, results are worse than the precision, this further proves that the recommended content is included by the mentor LP.

As described in Chapter 3, under the length gap between *RU-LP* and *U-LP*, the RU-POMDP is designed to identify which resources need to be reviewed and inserted in the recommended LP.

Based on the definition of precision and recall, when resources are inserted in *U-LP* to form *RU-LP*, the precision is expected to slightly increase if each of the added resources is part of GT and is at a good place. Precision is expected to decrease if some of the resources inserted are not at a good place or are not part of GT. The results of recall are also increased significantly for both of our datasets.

We first present the experimental results details of the EOLE dataset by comparing the U-POMDP and the RU-POMDP. For GL learners, the RU-POMDP precision result slightly decreases (about 9%), which means that most of the resources inserted in *U-LP* are adequately chosen and placed. The decrease is higher for AL, about 36%. Even though, if we compare the RU-POMDP results with the basic POMDP and the M-POMDP, the results of RU-POMDP for AL group learners are still the highest.

Then we compared the detailed experimental results between the U-POMDP and the RU-POMDP. On the EdNet dataset, for GL learners, the precision increase is 35.6% and the recall increase is 135%. For the AL learners, precision and recall increase is respectively 5.3% and 94%. The improvement of the results is better than that of the EOLE database. This is to say the accuracy of the resource recommended for review is more accurate than in the EOLE dataset.

The experimental results on the EOLE and EdNet datasets are still both improved. The average increase on EdNet is even larger, which may lead by the fact that after training with more data, the RS model is more accurate.

The increase of the precision and recall confirm the previous findings: it increases significantly (more than twice) for GL and increases less for AL. This confirms that the idea of managing learners' memory as a post processing is an adequate way to manage learners' memory.

Now, we would like to compare M-POMDP and RU-POMDP. Both models recommend LP by managing learners' memory. They mainly differ in the fact that M-POMDP allows repetitions in the recommended LP, without explicitly managing it, whereas RU-POMDP explicitly manages repetition. For the EOLE dataset, a significant improvement of the precision and of the recall has been obtained for GL (3% and 20% respectively). The improvements are different for AL, where the precision is increased by 16% and the recall by 2%. For the EdNet dataset, the precision for GL and AL increase is respectively 41% and 5%. The recall for GL and AL increase is respectively 14% and 6%. We notice that the increase of precision for GL is always huge and the recall increase is also large enough. For a RS with POMDP as the cornerstone, the most important thing we value is the reward of an LP. The LP of GL starts with a very good LP_{start}, and the recommended LP can just achieve the maximum reward meanwhile this LP fit the profile of the target learner. So it is normal that the recommended LP for GL is always better than the other groups of learners.

This shows that compared with M-POMDP, the content recommended by RU-POMDP is

more accurate. For recall, it is normal that the improvement is small because generally, the LP length of GL and AL is greater than or equal to the length of the recommended path. At the same time, under the premise that the results of M-POMDP are good enough, the improvement of RU-POMDP in results is difficult.

Highly important, the execution time of RU-POMDP is more than significantly decreased compared with the M-POMDP. Indeed, it is 90% lower (9.95 seconds vs 110.20 seconds). This decrease has a twofold justification. First, the average length of the *U-LP* is decreased by about 30%. As the complexity of POMDP is exponential to the value of the horizon, this justifies this high decrease in running time. Second, the learning path completion step has a low complexity, which does not increase the running time.

During the same time, the used storage memory is also decreased significantly. As presented in Figure 5.4, we can see that the POMDP-based model needs to store all possible branches LPs, which means that the larger the length, the larger the memory space requirement. As the length of LP decreases, the average storage space of RU-POMDP is 759 Kb and the average storage space of M-POMDP is 62Kb, shrunk by 92%.

These figures confirm the relevance of the construction of recommended LP in two steps: first building a unique learning path, and second managing learners' memory by inserting repetitions in a post-processing step, which contributes to the decrease in the running time. In addition, the way we propose to identify the resources that are repeated and the place where to insert them in *U-LP* seem also to be adequate as it contributes to increasing the accuracy (precision and recall) of the recommendations.

In the last part of this section, we specifically focus on the PL group of learners. Considering the case where PL are self mentors, we can see that precision and recall are lower than the ones of GL and AL, whatever is the model. These low values were expected and are in line with the preceding section: PL group includes learners who tend to fail, and POMDP-related models tend to recommend LP that conduct to a high reward (success). There is an expected mismatch between both paths (recommended LP and GT). For the PL group, considering the column where mentors are learners with a higher level, the results are significantly increased. For the EOLE dataset, compared with self-mentor, the average increase for precision is 100% and the average increase for recall is 22.3%. For EdNet dataset, the increase is 13.7% and 33.3% for precision and recall respectively. These results are as we expected: LP recommendations for promising learners are expected to gradually increase the learner's learning performance by specifying the resources to be learned. There is another reason why the recall on the EOLE dataset increases less than on the EdNet dataset: since the EOLE is not a very big dataset, the high-level GT should be longer than the self GT for promising learners while the EdNet is a big enough dataset that the length for all the three groups is relatively stable.

Another point worth noting is that both recall and precision for M-POMDP and RU-POMDP have a certain degree of improvement for the self mentor configuration, for both EOLE and EdNet datasets. That means M-POMDP and RU-POMDP models are both able to recommend more suitable LP for target learners in PL groups than the pure POMDP model. If we focus on the high-level mentor results (last column of Tables 5.5 and 5.6, we can find that this improvement is larger than the self mentor on both the two datasets.

From this fact, our conclusion can be described into two parts. For the PL group learners, our RU-POMDP model recommends more suitable LP. In our point of view, the M-POMDP also has good enough performance. The learning habits that our recommended LP conforms to are between the target learner and the higher learner.

5.3.4 New Evaluation Measures for POMDP Based RS

In this section, since the LP_{start} of the PL learner group is not very regular and as we mentioned in Chapter 4, the results of LP group learners are of little significance for experimental results other than DLP, so we mainly focus on the learners of the GL, AL group learners. The PL is set as a secondary focus.

Comparing MC and SPM with POMDP

In this section, we first compare two naive recommendation algorithms MC and SPM used at the beginning of this chapter with the basic POMDP model to show the extent to which POMDP is suitable for LP recommendations.

For the EOLE dataset, compared with MC and SPM in Table 5.3 and 5.4, the basic POMDP slightly increases the accuracy of the recommendations: for the TLLP measure, the POMDP results of GL and AL are increased by 1.4% and 16.7% respectively compared with the MC, and there is a medium increase compared with SPM (16.7%, 42.1% for GL and AL).

This is also true for SLLP and PLP measures, where the recommended LP of POMDP is always better than MC and SPM on GL and AL. For the SLLP measure, compared with MC, the results of POMDP are increased on average by 25.0%. Compared with SPM, the POMDP performances are almost the same. For PLP measure, the POMDP results are also higher than MC and SPM. The improvement in POMDP results compared to MC is not as much improvement compared to SPM.

For the EdNet dataset, we also compared the baseline POMDP with the MC and the SPM on TLLP, SLLP, and PLP separately. First we list some basic numerical information for comparison: for TLLP on MC, POMDP improves the performance results of GL and AL groups by 25% and 50%, respectively; while on SPM, POMDP improves the performance of the same two groups by 25% and 200%, respectively. For the SLLP, the POMDP increased slightly for both AL and GL group for the MC but decreased for SPM in the AL group and little change in the GL group. These results positive are in line with what was expected as MC is a basic Markovian algorithm, less complex than POMDP based algorithms. For the negative results, a possible reason is the different choices of learners' mentors: our mentor selection may still be flawed if the difference between the recommendation LP and the mentor is too big. For example, the chosen mentor and the target learner are too similar, and the recommended LP improves a certain level. In this case, the SPM algorithm provides better recommendations.

For the comparison of the recommended LPs of GL and AL, we get the same results as we expected: the basic POMDP recommends betters LPs for target learners than neither MC nor SPM. It also proved that the POMDP does have good enough performance in LP recommendation.

In addition to this conclusion, we found an interesting phenomenon. Here we can see that the results obtained by the above three evaluation methods are that the LPs recommended by POMDP improves the GL group less than the AL group. This is the opposite of the results obtained by the MC and SPM recommended paths. Again, the recommended LP for the GL group is more difficult because they do not have much room for improvement; whereas it is relatively easy to improve the target learners in the AL group. Therefore, in our opinion, the AL group is more likely to get the recommended LPs with good performance. For this fact, we conclude that our POMDP recommendation model better considers personalized recommendations for target learners.

Evaluate POMDP Based LPRSs with Proposed New Evaluation Measures

In this section, we evaluate the performance of our POMSPD based LPRSs with our proposed new offline evaluation measures. Overall speaking, POMDP based models have different degrees of improvement in the two datasets among the three groups of learners.

Measures	Method	GL (mentor)	AL (mentor)	PL (mentor)	PL (hi-lvl)
TLLP	POMDP	0.71	0.71	0.40	0.80
	M-POMDP	0.86	0.71	0.50	0.75
	U-POMDP	1.0	1.0	1.0	1.0
	RU-POMDP	1.0	1.0	0.80	1.0

Table 5.7: TLLP for EOLE dataset

Measures	Method	GL	AL	PL
TLLP	POMDP	0.5	0.6	0.3
	M-POMDP	0.5	0.6	0.4
	U-POMDP	0.4	0.5	0.3
	RU-POMDP	0.7	0.7	0.4

Table 5.8: TLLP for EdNet dataset

Firstly we focus on the comparison of M-POMDP and our baseline basic POMDP.

While comparing the baseline POMDP and M-POMDP for the TLLP measure for both EOLE and EdNet dataset in Table 5.7, the results improved in almost the three learners groups. Among them, the M-POMDP results of GL and PL is improved very large. It is especially true for GL, the data is improved to 0.86. Under the premise that the basic POMDP model is good enough, the relatively little improvement is understandable. But for the EdNet dataset, from the point of view of TLLP, the results have not been improved. It is more difficult to improve the LP in the EdNet, there are several reasons: 1. same as in the EOLE dataset, the result of the recommended LP of basic POMDPD reaches a certain level; 2. compared with the EOLE dataset, the recommended LP length in the EdNet dataset is longer and there are more optional resources to be chose to form an LP, so the recommendation is more difficult; 3. for the EdNet dataset in Table 5.8, since the correlation between the resources is not as strong as that of EOLE, this result is also in line with our expectation. Comparing the TLLP results of these two datasets, we conclude that at least M-POMDP can improve a recommended LP through the operation of learner memory, and make this LP closer to the LP of learners with better learning habits.

Then we compare the results of SLLP between M-POMDP and POMDP in Table 5.9 and Table 5.10. In Table 5.10 we can see that the distance between the mentor and the target learner of LPs recommended by M-POMDP is also closer than the distance of the basic POMDP. This is especially true for the GL and AL groups, the distance was improved by 0.6 lengths.. If we compare the results of TLLP and SLLP, we get the conclusion that although the recommended LPs are not close a lot to all the top learner's paths, they have quite an improvement for similar learner's LPs (similar but with better performance). And in the EdNet, the SLLP for GL decreased a little bit, for AL increased. This can be explained by the same fact that there are more optional learning resources in the EdNet dataset, which increases the difficulty of the recommended LPs. Especially for a dataset with a weaker correlation than EOLE, GL learners may go to learn some resources they are interested in.

Measures	Method	GL	AL	PL
SLLP	POMDP	5.6	5.8	6.3
	M-POMDP	4.8	5.2	6.2
	U-POMDP	5.2	5.5	6.0
	RU-POMDP	4.9	5.8	6.2

Table 5.9: SLLP for EOLE dataset

Measures	Method	GL	AL	PL
SLLP	POMDP	7.2	7.1	7.1
	M-POMDP	7.3	6.8	6.9
	U-POMDP	6.7	6.3	7.0
	RU-POMDP	6.7	6.8	6.9

Table 5.10: SLLP for EdNet dataset

For the comparison of POMDP and M-POMDP in PLP, we also start with the EOLE dataset in Table 5.12. The AL group improved the most since the most then is the GL and PL (GL, AL, PL increased 6.9%, 17.5%, and 3.2% respectively). The improvement in results reflects the improved scores on the quizzes/exams that the recommended LPs help learners achieve. Furthermore, this reflects the improvement of learners' knowledge level. The results show that AL has the highest promotion, which is somewhat unexpected, because PL is easier to promote, followed by AL, and GL is the most difficult. But given that PL learner's LP_{start} is not so regular, it is understandable that this part of the improvement is limited. Then for the EdNet dataset in Table 5.12, the increase of GL is smaller than AL which is very much in line with our expectations. Through the cooperation of the two tables, it is proved that the recommended LP we obtained through M-POMDP can improve the knowledge level of learners.

Measures	Method	GL	AL	PL
PLP	POMDP	5.8	5.7	6.3
	M-POMDP	5.4	4.7	6.1
	U-POMDP	5.7	4.4	5.9
	RU-POMDP	5.3	5.0	6.1

Table 5.11: PLP for EOLE dataset

Measures	Method	GL	AL	PL
PLP	POMDP	6.1	6.3	5.7
	M-POMDP	5.9	5.9	5.7
	U-POMDP	6.7	6.6	6.0
	RU-POMDP	5.9	6.2	5.5

Table 5.12: PLP for EdNet dataset

From the above analysing of comparing results between POMDP and M-POMDP, we solved a question that a simple memory model, i.e. M-POMDP do help recommended accurate LPs for target learners.

After the analysis for our M-POMDP RS, we then analyze RU-POMDP.

For the RU-POMDP, we directly compare it with the M-POMDP. We firstly analyze the results of the TLLP measure in Table 5.9 and Table 5.10. Before the experiment, we expected that the RU-POMDP should outperform or at least equal to the performance of the M-POMDP

Measures	Method	AL	PL
DLP	POMDP	0.1	0.2
	M-POMDP	0.5	0.6
	U-POMDP	0.3	0.7
	RU-POMDP	0.3	1.1

Table 5.13: DLP for EOLE dataset

Measures	Method	AL	PL
DLP	POMDP	0.2	0.41
	M-POMDP	0.5	0.59
	U-POMDP	0.3	0.67
	RU-POMDP	0.4	0.61

Table 5.14: DLP for EdNet dataset

model. For both the EOLE and EdNet datasets among all the three groups of learners, the RU-POMDP did perform better than the M-POMDP. This comparison preliminary confirms the effectiveness of our RU-POMDP. Then we focus on the U-POMDP measured by TLLP evaluation measure. It is worth noting that all the U-POMDP paths in the EOLE dataset yield good results; the U-POMDP paths in the EdNet dataset the performance is second only to RU-POMDP. These prove that our U-POMDP recommendation effect conforms to the learning habits of the GL group.

Second, we compare SLLP metrics in Table 5.9 and Table 5.10. In the EOLE dataset, the results of RU-POMDP were not better than M-POMDP, but the difference was not significant: Among them, the results of learners with only AG results of RU-POMDP are quite lower than M-POMDP, whereas the results of RU-POMDP for GL groups learners are only slightly inferior to M, and the results of PL groups are completely consistent. But for the EdNet dataset, its results for GL are significantly improved, for AL and PL group learners are completely consistent. For the EOLE dataset, the results may be explained by the fact that for a target learner, when the recommendation context is complex and the data set is small, his corresponding mentor is difficult to find. As the dataset grows, the target learner can find more suitable mentors, so the results in the EdNet dataset are in line with our expectations.

Thirdly, for the PLP results in Table 5.11 and Table 5.12, the results are similar to the results of the SLLP measure: the results of RU-POMDP for AL group learners in both EOLE and EdNet datasets are always worse than the M-POMDP whereas the RU-POMDP for GL groups gets a little bit of progress compared with M-POMDP for both EdNet and EOLE dataset.

This result is beyond our expectations. Go back to the precision and recall results of comparing the precision and recall in and Table 5.5 Table 5.6, we find that for the AL group learners' results of RU-POMDP is always better than the M-POMDP. But the increase is less than the GL and PL group learners. This shows that our RU-POMDP is still well performed. At least the comprehensive results are not weaker than M-POMDP. Considering the time and space complexity, we believe that the performance of RU-POMDP is very meaningful. These results above also confirm that the RU-POMDP has reduced complexity, generally performs equal or better than M-POMDP. This also answers the question we asked at the beginning of this chapter: Compared with the M-POMDP, RU-POMDP has the ability to recommend LPs that are more suitable for target learners under the premise of reducing time and space complexity.

At last of the analysis of our POMDP based RS models, we focus on the DLP. For both of the two datasets, whether it is POMDP, M-POMDP, or RU-POMDP, the differences in DLP

show that for PL group learners the results are always greater than the AL group.

In the EOLE dataset, the RU-POMDP gets the best result of 1.1 for the PL group. Whereas the others are all less than this result. The second best is the result of U-POMDP, with its peculiarities, it is proved that the personalization level of our U-POMDP is high enough. For the Al group in EOLE dataset, the best result comes from M-POMDP and the second best comes from U-POMDP and RU-POMDP.

For the EdNet dataset, it is still the RU-POMDP for PL group learners have the best result of 0.61, if we excluding U-POMDP because of its peculiarities. The second best is still the M-POMDP for PL group. The differences in the distance indicate that the recommended LP provided by our POMDP algorithms for LP learners are closer to the learning habits of PL learners. The LP recommended by RU-POMDP is more in line with the learning habits of PL learners. This is in line with our expectations since the RU-POMDP considers personalization in both the POMDP model and the target learner's memory strength model.

The results from AL to PL become better proves that our recommendation LPs recommended by both M-POMDP and RU-POMDP can be personalized differently according to the different learning abilities of target learners. Further, for the RU-POMDP, the comparison results from EOLE and EdNet dataset proved that the recommended LPs that learners in the PL group get are always more in line with their knowledge level.

In the end, we analyze and summarize the results' differences between the two datasets.

For all our models, EdNet improvements are always less than EOLE for all M-POMDP, U-POMDP, and RU-POMDP for both new and traditional measures. As we talked about before, our EdNet dataset has more resources, and it is more difficult to recommend accurate recommendation paths. From this point of view alone, it is normal that the improvements of various measures of M-POMDP of EdNet are not as good as that of the EOLE dataset. For the RU-POMDP, the first possible reason is due to the low precision of U-POMDP. The second possible reason is that when focusing in detail on the repetition rate of LP, it is significantly smaller than one of the EOLE datasets. However, both of the M-POMDP and RU-POMDP compared with the baseline including MC and SPM, the accuracy is significantly increased, which confirms the pertinence of our proposed approaches.

5.3.5 POMDP Conclusion

In this section, we were motivated by the LP recommendation challenge. We proposed to tackle this problem with POMDP and manage learners' memory to provide them with repeated resources and increase their retention and learning outcome. The originality of this model lies in the way learners' memory is managed, which is guided by the wish of limiting the complexity of the model.

Two models are proposed. M-POMDP manages memory in the states of the model, in a really simple way. RU-POMDP manages memory in an external step. RU-POMDP is quite original as it not only relies on a unique LP, i.e. a learning path that ensures that no repetition occurs, but also proposes to manage learners' memory in a post-processing step. This way to form an LP is intended to be accurate, and the model less complex.

We have validated our experiments using two datasets with two groups of evaluation measures. Through the comparison with our basic POMDP and MC, SPM, our basic POMDP is validated to be much better than both of the MC and SPM models which validated that the POMDP model used for solving the LPRS is well-performed. In addition, with both precision and recall measures, the U-POMDP model is also validated to be effective in recommending the necessary

components of a unique recommendation LP. Further, the recommendation effectiveness of our spaced repetition module is also verified: the recommendation effectiveness of RU-POMDP is not weaker than that of M-POMDP.

In the second step, we validated our RS models using the new evaluation measures method by comparing the M-POMDP and RU-POMDP with our basic POMDP. The M-POMDP and RU-POMDP are both better than the performance of the basic POMDP RS for all the three groups of learners in both the EOLE and EdNet datasets. This further validated that our two memory-based RS models are effective.

In terms of complexity, the time and space complexity of our M-POMDP are not significantly improved compared to POMDP. Moreover, the time consumption and memory storage consumption of RU-POMDP is much smaller than that of M-POMDP, so our experiments verify that our RU-POMDP is a very successful model.

The experiments conducted, with an offline protocol, confirm that both models significantly increase the accuracy of the baseline model, which does not manage memory. RU-POMDP is even more accurate on some specific learners while reducing drastically the computation time.

6

Conclusion and Perspectives

With the development of technology, the Internet provides more and more convenient services to people. People also face the problem of being overloaded with received data. During the same time, the in-depth research and development of RSs makes everyone's online life more and more convenient. This situation also exists in the field of education. This thesis describes RS models for learning path in the education domain.

In this chapter, we firstly summarize the work conducted in this PhD Thesis. Secondly, we present our planned future research directions, based on the work conducted here.

After a detailed overview of the state of research on RSs and associated evaluation measures, we have focused on the works conducted about LP RSs for the education domain. These works highlight that the accuracy of RSs is significantly improved when the learners memory is taken into account. The learning strategy generally used is the well-known spaced repetition strategy. Based on this review, we designed POMDP based RSs that take learner's memory into account in the recommendation process, through spaced repetition strategy. Besides, we found that there is a temporary vacuum in research on the offline evaluation measures of path recommendations, with few people specialising in this area, and a more shortage of research on the evaluation measures of LP recommendations. We have therefore designed new evaluation measures dedicated to learning paths specifically and for use in education. These evaluation measures take into account the learning levels of learners.

6.1 Summary and Contributions

With the development of e-learning, the learning environment becomes more and more complicated. When recommending a learning path, we increasingly need to consider some partially observable learner's information and what learners need to review as they learn.

Based on the contributions summarized above, we will now answer the questions raised in Chapter 1.

6.1.1 How to Build a LP RS that Considers Educational Characteristics?

Our proposed models are suitable for use in complex e-learning environments. It can be used in the following cases:
1. They can be used for building RSs for MOOC platforms which always have lots of interactions and lots of learners with noted evaluation.
2. In real educational environments such as universities or primary schools, an e-learning system

with our RS models can be used as an auxiliary means to help teachers teach students, such as in the university that collected EOLE dataset.
3. For another example, on some self-taught online learning platforms, such as the TOEIC platform that collected EdNet dataset, it is also suitable to use our RS models to help learners efficiency learning in a learning process of medium to long duration.

As described in Chapter 1, the first scientific question, "how to build a educational LP RS that considers educational characteristics" can be divided into two sub questions.

The first sub-question is related to the uncertainty which can also be viewed as a problem related to partial observation, mainly due to the lack of information about learners. To resolve this sub-question, we used POMDP which is designed to deal with partially observable problems. This algorithm is widely used in robotic domain for calculating an optimal strategy that aims at scheduling to accomplish a predetermined goal. Generally speaking, the strategy of a agent in POMDP is to decide how to accomplish the goal through a *series* of consecutive actions. A LP is a sequence of learning activities performed by a learner. We think that the process of improving the knowledge level of the learner through the learning path is similar to that of the agent in POMDP completing the predetermined goal. In both of the two RS models that we propose, we proposed to manage the main uncertainty of learners, i.e. about their knowledge level. This part is managed into the state definition of the models, in both of the M-POMDP and RU-POMDP model.

The second sub-question is about the trade off between learning new knowledge and reviewing old knowledge since learning is a long-term process. To resolve this sub-question, we relied on the widely used theory called spaced repetition that can deal with this trade-off. We integrated this spaced repetition theory into both POMDP based models proposed. This strategy is based on the famous Ebbinghaus memory curve. Integrating this strategy in POMDP results in modifications of all the core elements (S, T, p, r, Ω, O). The first model M-POMDP embeds important elements of the Ebbinghaus memory curve into the POMDP modelling, in particular, in the definition of state. The second model RU-POMDP considers this curve as a post process. It firstly constructs a recommendation LP without repetition, based on a directed acyclic graph (here the model is called Unique-POMDP), and then fits the learner's spaced repetition strategy into this recommendation LP using an Ebbinghaus memory curve and Poisson distribution. Through experiments conducted on two real-world datasets, we validated that both M-POMDP and RU-POMDP consistently perform better performance than a traditional POMDP. This shows that both models solve to a certain extent the trade-off problem of reviewing and learning new knowledge.

6.1.2 How to offline evaluate the LP recommendations/How to Evaluate Learning Path

Based on the fact that the literature lacks offline evaluation measures for recommendations in the educational frame, especially for learning path recommendations, we proposed four new evaluation measures.

These measures are designed to be simple measures. Recall this is because there are very rare easy-to-use evaluation measures and there are few evaluation methods for LP RSs. First, they rely on a small amount of information: traces of learners' activity and results at exams. They do not require any other additional information about resources (content) or learners, they can thus be used in most of the datasets. Second, they are designed to fit the educational domain, at the opposite of several measures of the literature. Indeed, they are designed to represent elements such as the increase in knowledge level and the fitting to learners' learning behavior. These mea-

sures have been assessed on recommendation algorithms of the literature. These algorithms have been chosen for their simplicity and the exact understanding of the recommendations performed. They are also evaluated on the POMDP-based recommendation algorithms that we introduced.

The quality of these measures is ensured through our experiments, and all of them are universal: TLLP and SLLP evaluate LP through determining whether LP contains good learning habits. The PLP evaluates LP through the increase in the level of knowledge of learners. DLP is radically different from the other three measures proposed, it is dedicated to promising learners only. As an evaluation method with a special angle of consideration, DLP also meets the design requirements.

As described in Chapter 4, our four new evaluation measures are all based on different measures for selecting mentors. While very novel and full of originality, due to its characteristics, this evaluation measures are flawed: they are all mentor-based, especially SLLP, DLP and PLP; once the mentor chooses poorly, it will affect the results of the evaluation.

This also limits the scope of the dataset to which the method can be applied: if the dataset is too small, there may be no suitable mentors.

We also plan to propose additional measures to conduct a thorough analysis of the recommendation algorithms of the literature.

6.2 Future Work

This PhD thesis contributed to the specific problem of learning path recommendation, i.e. the problem of recommending sequences of educational resources. Our contributions can be viewed as a first step to tackle this big challenge, and our contribution did not cover all the directions, and what is covered can still can be optimized. The following perspectives aim to proposed solutions to some of the remaining challenges.

6.2.1 Future Works in Experiment Design

In the details of designing experiments, our work still has the following tasks to be finished:

1. The length of the recommended LP is an important factor that affects the experimental results. We can try multiple lengths. For the length of the recommended LP in future work. Through comparing them, we may find the most suitable path length. Further, on the basis of experiments, we can summarize the length rule of the recommended path. This is very important in LP recommendation. This is very important in LP recommendation. So far we have not found anyone dedicated to this work.

2. We chose Sarsop as the solver in the computation of the POMDP model. Pruning in Sarsop preserves one of two paths with equal payoff. On the one hand, this optimizes the time complexity of the calculation, and at the same time, the negative result is that it will limit the diversity of paths. In future work, we will try some different POMDP solvers and compare their pros and cons.

3. Zhou et al. [18] compared the accuracy of a recommended LP with next step resource recommendations. In this PhD Thesis, we do not conduct a similar comparative experiment. This comparison can obtain an appropriate recommended path length.

6.2.2 Recommendation LP Problem for promising Learners

As we research deeper into RSs in education domain, we discovered a problem that was very rarely been focused on before: in LPRS, these promising-performed learners' need, more than any other learners, personalized and adequate recommendations to overcome their difficulties. Considering learners with learning habits that conduct to poor performance, it is also more difficult to make suitable personalized recommendations.

This will be a focus of our future work, and we focus here on this problem.

Generally speaking, personalized recommendation needs to take into account the background of the learner. For learners with promising learning level, the RS should consider the bad learning habits of poor learners. At the same time, the purpose of RSs aim to recommend an appropriate LP, which should avoid bad learning habits and help learners improve their knowledge. A recommended path to help learners with lower learning ability, needs to balance the relationship between the background material for the target learner and the improvement of the level. For example in a MPD/POMDP based LPRS, the POMDP aims to get the best reward for a recommended LP. In order to get a high-reward LP, MDP/POMPD will prioritize strategies that perform well. As a result, the recommended LP may be quite different from the background of the student. Although this kind of LP will generally get better returns, it may not be so in line with the background of the target user.

As far as we know, this problem is only explicitly considered in one kind of RSs such as Zhu et al. [24] presented, the RSs explicitly takes into account the smoothness of a recommended LP, and the difficulty in such a path is simple and gradually becomes difficult.

6.2.3 Exploration of Different Methods to Recommend Suitable LPs for Learners

In this PhD Thesis we proposed to view the personalized LP recommendation problem as a sequential decision making problem. Other structures of Markovian algorithms can also be used. We especially this about such as Monte Carlo Tree Search(MCTS), deep learning, etc.

As described by [162], MCTS combines a tree search with the generality of random sampling methods. After AlphaGo made a big splash in Go confrontation, MCTS got wider attention. Some researchers used it in RS domain, with a positive feedback [163, 164]. It is worth mentioning that random sampling in traditional MCTS can be replaced by Markovian algorithms, and more and more people start to be interested in this, especially in the field of online recommendation and online decision-making.

In our point of view, even the POMDP is good enough, there are still one big drawbacks: the high time consuming. Whereas this drawback can be solved by MCTS. With a well defined POMDP under a MCTS, we can defined a much more complex state. Defining a more comprehensive POMDP might address the future work that we raised in the section 6.2.2.

As we discussed in Chapter 2, deep learning requires to have a large amount of training data to train the network. At the same time, deep learning requires a high computation time to train a model that performs well. In addition to these shortcomings, algorithms based on deep learning frameworks generally perform better. We will try a route recommendation model based on a deep learning framework in the future.

The above work can be divided into four parts:

- We firstly propose to built a general POMDP RS based deep learning and compare it with our basic POMDP model.

- Secondly we propose to use a RS using POMDP, MCTS in a deep learning structure.
- Thirdly, we propose to use a LPRS only based on POMDP solved by MCTS.
- At last, an online evaluation based on POMDP with MCTS will be conducted. This last online model with experiment requires the participation of real learners. This part should be realized under the premise that the above experiments verify the effectiveness of a MCTS POMDP RS.

Then we could compare our two recommendation models with models based on these two structures in our future works.

6.2.4 Complexity of POMDP

Besides the above future works in section 6.2.3, we also consider some optimizations the POMDP in our future work. First, all the recommendation algorithms should have a well-controlled complexity. In our work, we firstly tried to control the complexity through simplifying the state definition in the M-POMDP. In our experiments we confirmed that the complexity of the model increased exponentially: every time the length of the recommended path increases by one resource, the computation time will be much longer than before. As we talked about, the complexity of POMDP is related to the number of states and actions, as the policy that calculated by POMDP can be described as $\pi B \rightarrow A$, where the B is a belief space represents the state probabilities distribution and the A is an action space. The number of states and actions problem is described in many works of the literature, for example in speech recognition that proposed by Young et al. [149].

As described above, for the future work, our solution will focus on two dimensions: reduce the number of actions or states, called macro-action and abstract state. But both methods are based on an approximate. That is to say, a certain amount of data accuracy will be lost.

Besides, we will also explore a hierarchical POMDP, as presented in figure 6.1. Given a set of resources (from r_1 to r_{14}), with an action set A, there are several possible optional LPs. If we arrange all resources into one state space S, POMDP will be very time-consuming. Under this premise, we can divide these resources into several sub-state spaces according to similarity. Then in each sub-state space, we build a sub-POMDP model. These sub-POMDPs are under the control of a central POMDP. For the case presented in figure 6.1, compare the original POMDP and hierarchical POMDP, the complexity is reduced from $\mathcal{O}(|S|^{14}|A|^{14}|O|^{14})$ to $\mathcal{O}(|S|^{4}|A|^{4}|O|^{4})$ according to a rough estimation as presented in [165].

This approach still has a disadvantage: which resources can be divided into several groups is demanded. Each group should have similar characteristics. This limits the scope of application.

Overall, the PhD Thesis addresses about one topic in the e-learning domain: learning path recommender system. This work combines the POMDP decision-making algorithm and the learner's memory model and validates their outstanding performance. In particular, future work can go further into the direction of using some modules to simplify the time complexity of the algorithm. This PhD Thesis also proposes a unique solution for how to evaluate the recommended learning path. Nevertheless, there is still a lot of work to do in online education.

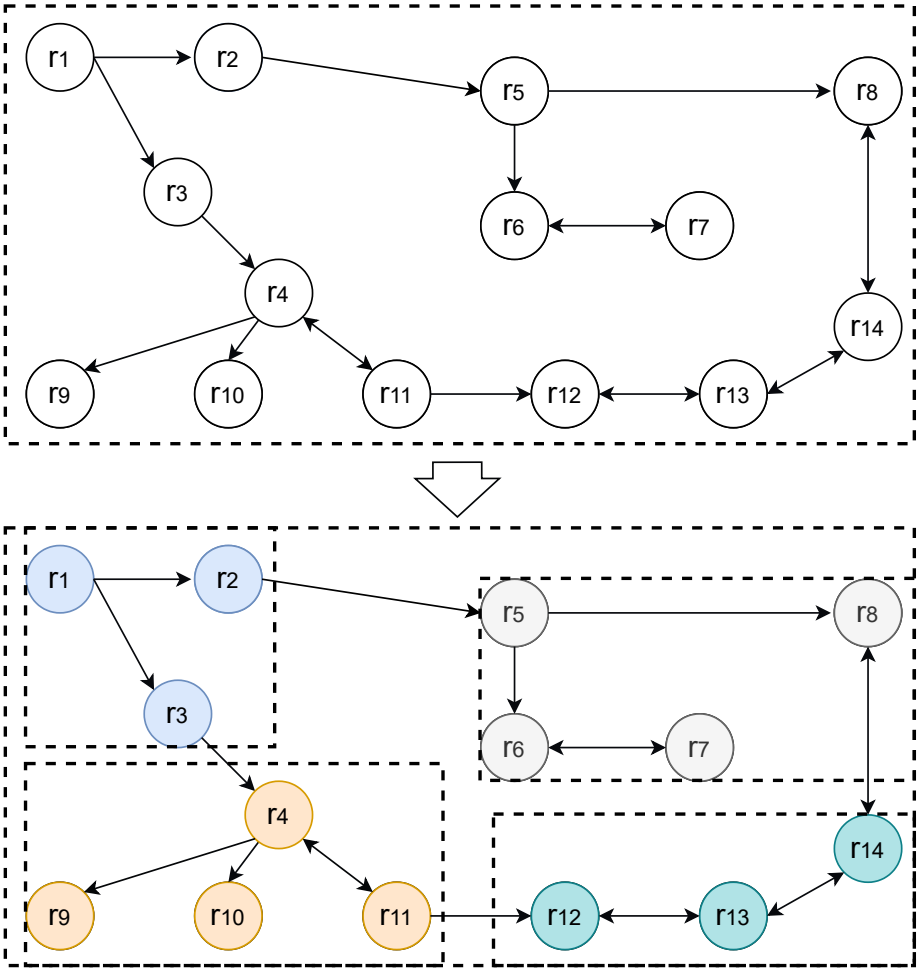


Figure 6.1: The hierarchical POMDP

A

Belief state update of POMDP

A.1 Value iteration of MDP and POMDP

If the value iteration of MDP is a finite horizon, and the horizon is n, the Bellman Equation should be:

$$\begin{aligned}\forall s \in S, V^{\pi}(s) &= E^{\pi}[\sum_{t=0}^{n} r_n | s_0 = s] \\ &= E[r_0 + \gamma r_1 + ... + \gamma^n r_n] \\ &= r(s, \pi(s)) + \gamma p(s', s, a) \overbrace{r(s', a) + ... + \gamma^n}^{V^{\pi} s'} \\ &= r(s, \pi(s)) + \gamma \sum_{s'} p(s', s, a) V^{\pi}(s')\end{aligned} \tag{A.1}$$

A.2 Formula of the belief state update

The computation of belief state in a POMDP based model is very important. In this part we present how to update the belief state.

The distribution of belief state is a conditional probability. In equation A.2, the $b_a^o(s')$ is the belief for one next possible state given an action a and it will get an observation o, the s' is the target next state after tanking action a from the current state s, the o is the related observation

from state s after taking action a and the s'' represents a general possible next state.

$$
\begin{aligned}
b_a^o(s') &= Pr(s'|a,o,b) \\
&= \frac{Pr(s',a,o,b)}{Pr(a,o,b)} \\
&= \frac{Pr(o|s',a,b)Pr(s',a,b)}{Pr(o|b,a)Pr(b,a)} \\
&= \frac{Pr(o|s',a)\sum_{s\in S} Pr(s',s,a,b)}{Pr(o|b,a)Pr(b,a)} \\
&= \frac{Pr(o|s',a)\sum_{s\in S} Pr(s'|s,a,b)Pr(s,a,b)}{Pr(o|b,a)Pr(b,a)} \\
&= \frac{Pr(o|s',a)\sum_{s\in S} Pr(s'|s,a,b)Pr(s|a,b)Pr(b,a)}{Pr(o|b,a)Pr(b,a)} \\
&= \frac{Pr(o|s',a)\sum_{s\in S} Pr(s'|s,a,b)b(s)}{Pr(o|b,a)} \\
&= \frac{Pr(o|s',a)\sum_{s\in S} Pr(s'|s,a,b)b(s)}{\frac{Pr(o,b,a)}{Pr(b,a)}} \\
&= \frac{Pr(o|s',a)\sum_{s\in S} Pr(s'|s,a,b)b(s)}{\frac{\sum_{s\in S} Pr(o,s,b,a)}{Pr(b,a)}} \\
&= \frac{Pr(o|s',a)\sum_{s\in S} Pr(s'|s,a,b)b(s)}{\frac{\sum_{s''\in S}\sum_{s\in S} Pr(o,s'',s,b,a)}{Pr(b,a)}} \\
&= \frac{Pr(o|s',a)\sum_{s\in S} Pr(s'|s,a,b)b(s)}{\frac{\sum_{s''\in S}\sum_{s\in S} Pr(o|s'',s,b,a)Pr(s'',s,b,a)}{Pr(b,a)}} \\
&= \frac{Pr(o|s',a)\sum_{s\in S} Pr(s'|s,a,b)b(s)}{\frac{\sum_{s''\in S}\sum_{s\in S} Pr(o|s'',a)Pr(s''|s,b,a)Pr(s,b,a)P()b,a)}{Pr(b,a)}} \\
&= \frac{Pr(o|s',a)\sum_{s\in S} Pr(s'|s,a)b(s)}{\sum_{s''\in S}\sum_{s\in S} Pr(o|s'',a)Pr(s''|s,a)b(s)}
\end{aligned}
\tag{A.2}
$$

The belief state update is based on the Bayes' theorem and marginal probability, i.e., it follows:

$$Pr(A,B) = \frac{Pr(A|B)}{Pr(B)} \tag{A.3}$$

$$Pr(A) = \sum_i Pr(A)Pr(B_i) \tag{A.4}$$

A.3 Two tiger POMDP example

Here is an example of how to update belief state in a famous POMDP problem: two tigers problem.

The two tigers problem can be described as follows: Set with two doors. Only one door has a tiger behind, i.e. either tiger left(TL) or tiger right(TR). The agent has three possible actions: listen(L), open the left door(OL) and open the right door(OR). Under this condition,

we have a state space $S =< TL, TR >$, an observation state $\Omega =< TL, TR >$, an action space $A = L, OL, OR$.

Given an transition function, the transition function is defined as a matrix as follow:

Table A.1: Transition matrix

action / state	Listen	Open Left	Open Right
Tiger Left	if $s' = s, 1$; else,0	0.5	0.5
Tiger Right	if $s' = s, 1$; else,0	0.5	0.5

Given an observation function, the observation function is also presented as a matrix: Given

Table A.2: Observation matrix

	Tiger Left	Tiger Right
Hear Tiger Left	0.6	0.4
Hear Tiger Right	0.4	0.6

the initial $b_0 = \{s_{TL} = 0.5, s_{TR} = 0.5\}$, action a is listen, the next belief of tiger left can be updated as follow:

$$\begin{aligned} b_a^o(s_{TL}) &= Pr(s_{TL}, o, a, b) \\ &= \frac{Pr(s_{TL}, a, o, b)}{Pr(a, o, b)} \\ &= \frac{\sum_s Pr(s_{TL}, s, o, a, b)}{\sum_s Pr(o, s, a, b)} \\ &= \frac{Pr(o|s_{TL}, a) \sum_s Pr(s_{TL}|s, a, b) Pr(s, a, b)}{\sum_s \sum_{s'} Pr(o|s', a) Pr(s'|s, a, b) Pr(s, a, b)} \\ &= \frac{Pr(o, s_{TL}, a) \sum_s Pr(s_{TL}|s, a) Pr((s_b) Pr(a, b)}{\sum_s \sum_{s'} Pr(o|s', a) Pr(s'|s, a) Pr(s, b) Pr(a, b)} \\ &= \frac{Pr(o|s_{TL}, a) \sum_{s \in S} Pr(s_{TL}|s, a) b(s)}{\sum_{s' \in S} \sum_{s \in S} Pr(o|s', a) Pr(s'|s, a) b(s)} \end{aligned} \tag{A.5}$$

Note that, in equation A.5, the s' represents a general next state.

Followed the givens above, we have:

$$\begin{aligned} b_a^o(s_{TL}) &= \frac{0.6 \times (\overbrace{1 \times 0.5}^{s_{TL}=\text{Tiger Left}} + \overbrace{0 \times 0.5}^{s_{TL}=\text{Tiger Right}})}{\underbrace{0.6 \times (1 \times 0.5 + 0 \times 0.4)}_{s'=\text{Tiger Left}} + \underbrace{0.4 \times (0 \times 0.5 + 1 \times 0.5)}_{s'=\text{Tiger Right}}} \\ &= 0.6 \end{aligned} \tag{A.6}$$

So the next belief state is $b = \{s_{TL} = 0.6, s_{TR} = 0.4\}$.

Bibliography

[1] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," Communications of the ACM, vol. 35, no. 12, pp. 61–70, 1992.

[2] C. Yin, J. Wang, and J. H. Park, "An improved recommendation algorithm for big data cloud service based on the trust in sociology," Neurocomputing, vol. 256, pp. 49–55, 2017.

[3] J. López-Bastida, J. Oliva, F. Antonanzas, A. García-Altés, R. Gisbert, J. Mar, and J. Puig-Junoy, "Spanish recommendations on economic evaluation of health technologies," The European Journal of Health Economics, vol. 11, no. 5, pp. 513–520, 2010.

[4] D. Shasha, "Tuning time series queries in finance: Case studies and recommendations," IEEE Data Eng. Bull., vol. 22, no. 2, pp. 40–46, 1999.

[5] H. Tan, J. Guo, and Y. Li, "E-learning recommendation system," in 2008 International conference on computer science and software engineering, vol. 5. IEEE, 2008, pp. 430–433.

[6] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, "Recommender systems," Physics reports, vol. 519, no. 1, pp. 1–49, 2012.

[7] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," Knowledge-based systems, vol. 46, pp. 109–132, 2013.

[8] Y. Liang, "Recommender system for developing new preferences and goals," in Proceedings of the 13th ACM Conference on Recommender Systems, 2019, pp. 611–615.

[9] A. Nabizadeh, A. Jorge, and J. P. Leal, "Long term goal oriented recommender systems," in Proceedings of the 11th international conference on web information systems and technologies, 2015.

[10] Q. Shambour, "A deep learning based algorithm for multi-criteria recommender systems," Knowledge-Based Systems, vol. 211, p. 106545, 2021.

[11] M. O'connor, D. Cosley, J. A. Konstan, and J. Riedl, "Polylens: A recommender system for groups of users," in ECSCW 2001. Springer, 2001, pp. 199–218.

[12] A. H. Nabizadeh, J. P. Leal, H. N. Rafsanjani, and R. R. Shah, "Learning path personalization and recommendation methods: A survey of the state-of-the-art," Expert Systems with Applications, p. 113596, 2020.

[13] Z. Zhang, A. Brun, and A. Boyer, "New measures for offline evaluation of learning path recommenders," in European Conference on Technology Enhanced Learning. Springer, 2020, pp. 259–273.

[14] H. Xie, D. Zou, F. L. Wang, T.-L. Wong, Y. Rao, and S. H. Wang, "Discover learning path for group users: A profile-based approach," Neurocomputing, vol. 254, pp. 59–70, 2017.

[15] S. Wang, L. Hu, Y. Wang, L. Cao, Q. Z. Sheng, and M. Orgun, "Sequential recommender systems: challenges, progress and prospects," arXiv preprint arXiv:2001.04830, 2019.

[16] F. Yang, N. Liu, S. Wang, and X. Hu, "Towards interpretation of recommender systems with sorted explanation paths," in 2018 IEEE International Conference on Data Mining (ICDM). IEEE, 2018, pp. 667–676.

[17] 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, vol. 19, no. 4, pp. 713–735, 2014.

[18] Y. Zhou, C. Huang, Q. Hu, J. Zhu, and Y. Tang, "Personalized learning full-path recommendation model based on lstm neural networks," Information Sciences, vol. 444, pp. 135–152, 2018.

[19] H. Drachsler, K. Verbert, O. C. Santos, and N. Manouselis, "Panorama of recommender systems to support learning," in Recommender systems handbook. Springer, 2015, pp. 421–451.

[20] R. Venant, K. Sharma, P. Vidal, P. Dillenbourg, and J. Broisin, "Using sequential pattern mining to explore learners' behaviors and evaluate their correlation with performance in inquiry-based learning," in European Conference on Technology Enhanced Learning. Springer, 2017, pp. 286–299.

[21] M. Léonard, Y. Peter, and Y. Secq, "Patterns and loops: Early computational thinking," in European Conference on Technology Enhanced Learning. Springer, 2019, pp. 280–293.

[22] S. H. Kang, "Spaced repetition promotes efficient and effective learning: Policy implications for instruction," Policy Insights from the Behavioral and Brain Sciences, vol. 3, no. 1, pp. 12–19, 2016.

[23] T. Saito and Y. Watanobe, "Learning path recommendation system for programming education based on neural networks," International Journal of Distance Education Technologies (IJDET), vol. 18, no. 1, pp. 36–64, 2020.

[24] Q. Liu, S. Tong, C. Liu, H. Zhao, E. Chen, H. Ma, and S. Wang, "Exploiting cognitive structure for adaptive learning," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 627–635.

[25] A. H. Nabizadeh, D. Gonçalves, S. Gama, J. Jorge, and H. N. Rafsanjani, "Adaptive learning path recommender approach using auxiliary learning objects," Computers & Education, vol. 147, p. 103777, 2020.

[26] H. Zhu, F. Tian, K. Wu, N. Shah, Y. Chen, Y. Ni, X. Zhang, K.-M. Chao, and Q. Zheng, "A multi-constraint learning path recommendation algorithm based on knowledge map," Knowledge-Based Systems, vol. 143, pp. 102–114, 2018.

[27] K. Umemoto, T. Milo, and M. Kitsuregawa, "Toward recommendation for upskilling: Modeling skill improvement and item difficulty in action sequences," in 2020 IEEE 36th ICDE, 2020, pp. 169–180.

[28] C. Mejia, S. Gomez, L. Mancera, and S. Taveneau, "Inclusive learner model for adaptive recommendations in virtual education," in 2017 IEEE 17th International Conference on advanced learning technologies (ICALT). IEEE, 2017, pp. 79–80.

[29] B. Choffin, F. Popineau, Y. Bourda, and J.-J. Vie, "DAS3H: Modeling Student Learning and Forgetting for Optimally Scheduling Distributed Practice of Skills," in Proc. of the 12th EDM, 2019, pp. 29–38.

[30] L.-Z. Cui, F.-L. Guo, and Y.-j. Liang, "Research overview of educational recommender systems," in Proceedings of the 2nd International Conference on Computer Science and Application Engineering, 2018, pp. 1–7.

[31] T. Saito and Y. Watanobe, "Learning path recommender system based on recurrent neural network," in 2018 9th International Conference on Awareness Science and Technology (iCAST). IEEE, 2018, pp. 324–329.

[32] P. Pimsleur, "A memory schedule," The Modern Language Journal, vol. 51, no. 2, pp. 73–75, 1967.

[33] B. Settles and B. Meeder, "A trainable spaced repetition model for language learning," in Proceedings of the 54th annual meeting of the Association for Computational Linguistics, 2016, pp. 1848–1858.

[34] B. Tabibian, U. Upadhyay, A. De, A. Zarezade, B. Schölkopf, and M. Gomez-Rodriguez, "Enhancing human learning via spaced repetition optimization," in Proceedings of the National Academy of Sciences, 01 2019, p. 201815156.

[35] N. Kornell, "Optimising learning using flashcards: Spacing is more effective than cramming," Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition, vol. 23, no. 9, pp. 1297–1317, 2009.

[36] S. Reddy, I. Labutov, S. Banerjee, and T. Joachims, "Unbounded human learning: Optimal scheduling for spaced repetition," in Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 1815–1824.

[37] B. Choffin, "Algorithmes d'espacement adaptatif de l'apprentissage pour l'optimisation de la maîtrise à long terme de composantes de connaissance," Ph.D. dissertation, université Paris-Saclay, 2021.

[38] H. Ebbinghaus, "Memory: A contribution to experimental psychology," Annals of neurosciences, vol. 20, no. 4, p. 155, 2013.

[39] G. H. Teninbaum, "Spaced repetition: A method for learning more law in less time," J. High Tech. L., vol. 17, p. 273, 2016.

[40] D. Monti, E. Palumbo, G. Rizzo, and M. Morisio, "Sequeval: An offline evaluation framework for sequence-based recommender systems," Information, vol. 10, no. 5, p. 174, 2019.

[41] J. A. Konstan and J. Riedl, "Recommender systems: from algorithms to user experience," User modeling and user-adapted interaction, vol. 22, no. 1-2, pp. 101–123, 2012.

[42] P. Resnick and H. R. Varian, "Recommender systems," Communications of the ACM, vol. 40, no. 3, pp. 56–58, 1997.

[43] S. Wang, L. Cao, Y. Wang, Q. Z. Sheng, M. A. Orgun, and D. Lian, "A survey on session-based recommender systems," ACM Computing Surveys (CSUR), vol. 54, no. 7, pp. 1–38, 2021.

[44] A. S. Imran, K. Muhammad, N. Fayyaz, M. Sajjad et al., "A systematic mapping review on mooc recommender systems," IEEE Access, 2021.

[45] D. H. Park, H. K. Kim, I. Y. Choi, and J. K. Kim, "A literature review and classification of recommender systems research," Expert systems with applications, vol. 39, no. 11, pp. 10 059–10 072, 2012.

[46] S. S. Sohail, J. Siddiqui, and R. Ali, "Classifications of recommender systems: A review." Journal of Engineering Science & Technology Review, vol. 10, no. 4, 2017.

[47] D. Sacharidis, C. P. Mukamakuza, and H. Werthner, "Fairness and diversity in social-based recommender systems," in Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization, 2020, pp. 83–88.

[48] M. Sharma and S. Mann, "A survey of recommender systems: approaches and limitations," International Journal of Innovations in Engineering and Technology, vol. 2, no. 2, pp. 8–14, 2013.

[49] D. Kluver, M. D. Ekstrand, and J. A. Konstan, "Rating-based collaborative filtering: algorithms and evaluation," Social Information Access, pp. 344–390, 2018.

[50] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in Proceedings of the 10th international conference on World Wide Web, 2001, pp. 285–295.

[51] A. Bellogín, P. Castells, and I. Cantador, "Neighbor selection and weighting in user-based collaborative filtering: a performance prediction approach," ACM Transactions on the Web (TWEB), vol. 8, no. 2, pp. 1–30, 2014.

[52] J. Bobadilla, S. Alonso, and A. Hernando, "Deep learning architecture for collaborative filtering recommender systems," Applied Sciences, vol. 10, no. 7, p. 2441, 2020.

[53] R. Raziperchikolaei, T. Li, and Y.-j. Chung, "Neural representations in hybrid recommender systems: Prediction versus regularization," in Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 1743–1747.

[54] N. Haubner and T. Setzer, "Hybrid recommender systems for next purchase prediction based on optimal combination weights," 2021.

[55] S. Chen, J. L. Moore, D. Turnbull, and T. Joachims, "Playlist prediction via metric embedding," in Proceedings of the 18th ACM SIGKDD conference, 2012, pp. 714–722.

[56] A. H. Celdrán, M. G. Pérez, F. J. G. Clemente, and G. M. Pérez, "Design of a recommender system based on users' behavior and collaborative location and tracking," Journal of Computational Science, vol. 12, pp. 83–94, 2016.

[57] S. Wang, L. Hu, L. Cao, X. Huang, D. Lian, and W. Liu, "Attention-based transactional context embedding for next-item recommendation," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, 2018.

[58] B. Zhou, S. C. Hui, and K. Chang, "An intelligent recommender system using sequential web access patterns," in IEEE Conference on Cybernetics and Intelligent Systems, 2004., vol. 1. IEEE, 2004, pp. 393–398.

[59] F. Garcin, C. Dimitrakakis, and B. Faltings, "Personalized news recommendation with context trees," in Proceedings of the 7th ACM Conference on Recommender Systems, 2013, pp. 105–112.

[60] R. He and J. McAuley, "Fusing similarity models with markov chains for sparse sequential recommendation," in 2016 IEEE 16th International Conference on Data Mining (ICDM). IEEE, 2016, pp. 191–200.

[61] P. Wang, Y. Fan, L. Xia, W. X. Zhao, S. Niu, and J. Huang, "Kerl: A knowledge-guided reinforcement learning model for sequential recommendation," in Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 209–218.

[62] Y. Wu, C. Macdonald, and I. Ounis, "Partially observable reinforcement learning for dialog-based interactive recommendation," in Fifteenth ACM Conference on Recommender Systems, 2021, pp. 241–251.

[63] G. Shani, D. Heckerman, and R. I. Brafman, "An mdp-based recommender system," Journal of Machine Learning Research, vol. 6, no. Sep, pp. 1265–1295, 2005.

[64] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in Proceedings of the 19th international conference on World wide web, 2010, pp. 811–820.

[65] B. Hidasi and D. Tikk, "General factorization framework for context-aware recommendations," Data Mining and Knowledge Discovery, vol. 30, no. 2, pp. 342–371, 2016.

[66] R. He, W.-C. Kang, and J. J. McAuley, "Translation-based recommendation: A scalable method for modeling sequential behavior." in IJCAI, 2018, pp. 5264–5268.

[67] G. G. Judge and E. R. Swanson, "Markov chains: Basic concepts and suggested uses in agricultural economics," Australian Journal of Agricultural Economics, vol. 6, no. 429-2016-29354, pp. 49–61, 1962.

[68] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," 2015.

[69] B. Hidasi and A. Karatzoglou, "Recurrent neural networks with top-k gains for session-based recommendations," in Proceedings of the 27th ACM international conference on information and knowledge management, 2018, pp. 843–852.

[70] H. Ying, F. Zhuang, F. Zhang, Y. Liu, G. Xu, X. Xie, H. Xiong, and J. Wu, "Sequential recommender system based on hierarchical attention network," in IJCAI International Joint Conference on Artificial Intelligence, 2018.

[71] G. Zhou, N. Mou, Y. Fan, Q. Pi, W. Bian, C. Zhou, X. Zhu, and K. Gai, "Deep interest evolution network for click-through rate prediction," in Proceedings of the AAAI conference on artificial intelligence, vol. 33, no. 01, 2019, pp. 5941–5948.

[72] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in Proceedings of the 28th ACM international conference on information and knowledge management, 2019, pp. 1441–1450.

[73] J. Wang and Y. Zhang, "Opportunity model for e-commerce recommendation: right product; right time," in Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, 2013, pp. 303–312.

[74] M. Braunhofer, F. Ricci, B. Lamche, and W. Wörndl, "A context-aware model for proactive recommender systems in the tourism domain," in Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct, 2015, pp. 1070–1075.

[75] N. Du, Y. Wang, N. He, J. Sun, and L. Song, "Time-sensitive recommendation from recurrent user activities," in NIPS, vol. 15, 2015, pp. 3492–3500.

[76] C. Wang, M. Zhang, W. Ma, Y. Liu, and S. Ma, "Modeling item-specific temporal dynamics of repeat consumption for recommender systems," in The World Wide Web Conference, 2019, pp. 1977–1987.

[77] N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, and R. Koper, "Recommender systems in technology enhanced learning," in Recommender systems handbook. Springer, 2011, pp. 387–415.

[78] C. Obeid, I. Lahoud, H. El Khoury, and P.-A. Champin, "Ontology-based recommender system in higher education," in Companion Proceedings of the The Web Conference 2018, 2018, pp. 1031–1034.

[79] A. Ramachandran, S. S. Sebo, and B. Scassellati, "Personalized robot tutoring using the assistive tutor pomdp (at-pomdp)," in Proceedings of the AAAI Conference, vol. 33, 2019, pp. 8050–8057.

[80] A. H. Nabizadeh, A. M. Jorge, and J. P. Leal, "Estimating time and score uncertainty in generating successful learning paths under time constraints," Expert Systems, vol. 36, no. 2, p. e12351, 2019.

[81] G. Rasch, Probabilistic models for some intelligence and attainment tests. ERIC, 1993.

[82] C.-K. Yeung, "Deep-irt: Make deep learning based knowledge tracing explainable using item response theory," pp. 169–183, 2019.

[83] K. Pliakos, S.-H. Joo, J. Y. Park, F. Cornillie, C. Vens, and W. Van den Noortgate, "Integrating machine learning into item response theory for addressing the cold start problem in adaptive learning systems," Computers & Education, vol. 137, pp. 91–103, 2019.

[84] C. Piech, J. Spencer, J. Huang, S. Ganguli, M. Sahami, L. Guibas, and J. Sohl-Dickstein, "Deep knowledge tracing," arXiv preprint arXiv:1506.05908, 2015.

[85] S. Reddy, S. Levine, and A. Dragan, "Accelerating human learning with deep reinforcement learning," in NIPS workshop: teaching machines, robots, and humans, 2017.

[86] A. N. Rafferty, E. Brunskill, T. L. Griffiths, and P. Shafto, "Faster teaching via pomdp planning," Cognitive science, vol. 40, no. 6, pp. 1290–1332, 2016.

[87] F. Mi and B. Faltings, "Adaptive sequential recommendation for discussion forums on moocs using context trees," in Proceedings of the 10th international conference on educational data mining, no. CONF, 2017.

[88] H. Samin and T. Azim, "Knowledge based recommender system for academia using machine learning: A case study on higher education landscape of pakistan," IEEE Access, vol. 7, pp. 67 081–67 093, 2019.

[89] R. Sikka, A. Dhankhar, and C. Rana, "A survey paper on e-learning recommender system," International Journal of Computer Applications, vol. 47, no. 9, pp. 27–30, 2012.

[90] K. Holstein, Z. Yu, J. Sewall, O. Popescu, B. M. McLaren, and V. Aleven, "Opening up an intelligent tutoring system development environment for extensible student modeling," in International Conference on Artificial Intelligence in Education. Springer, 2018, pp. 169–183.

[91] P. Chen, Y. Lu, V. W. Zheng, and Y. Pian, "Prerequisite-driven deep knowledge tracing," in 2018 IEEE International Conference on Data Mining (ICDM). IEEE, 2018, pp. 39–48.

[92] T. Schodde, K. Bergmann, and S. Kopp, "Adaptive robot language tutoring based on bayesian knowledge tracing and predictive decision-making," in Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction, 2017, pp. 128–136.

[93] B. Tabibian, U. Upadhyay, A. De, A. Zarezade, B. Schölkopf, and M. Gomez-Rodriguez, "Enhancing human learning via spaced repetition optimization," Proceedings of the National Academy of Sciences, vol. 116, p. 201815156, 01 2019.

[94] S. Baleghizadeh and A. Ashoori, "The impact of two instructional techniques on efl learners' vocabulary knowledge: Flash cards versus word lists," Mextesol journal, vol. 35, no. 2, pp. 1–9, 2011.

[95] A. Burashnikova, M. Clausel, M.-R. Amini, Y. Maximov, and N. Dante, "Recommender systems: when memory matters," arXiv preprint arXiv:2112.02242, 2021.

[96] Y. He, Y. Zhang, W. Liu, and J. Caverlee, "Consistency-aware recommendation for user-generated item list continuation," in Proceedings of the 13th International Conference WSDM, 2020, pp. 250–258.

[97] D. Jannach, L. Lerche, and I. Kamehkhosh, "Beyond "hitting the hits" generating coherent music playlist continuations with the right tracks," in Proceedings of the 9th ACM Conference on Recommender Systems, 2015, pp. 187–194.

[98] A. Vall, M. Dorfer, H. Eghbal-Zadeh, M. Schedl, K. Burjorjee, and G. Widmer, "Feature-combination hybrid recommender systems for automated music playlist continuation," User Modeling and User-Adapted Interaction, vol. 29, no. 2, pp. 527–572, 2019.

[99] G. Kahn, P. Abbeel, and S. Levine, "Badgr: An autonomous self-supervised learning-based navigation system," IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 1312–1319, 2021.

[100] W. Luan, G. Liu, C. Jiang, and M. Zhou, "Mptr: A maximal-marginal-relevance-based personalized trip recommendation method," IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 11, pp. 3461–3474, 2018.

[101] X. Zhao, D. Xu, and S. Long, "Collaborative recommendation: A new perspective for personalized learning path generation." in Distance Education in China, 2017, pp. 24–34.

[102] N. Belacel, G. Durand, and F. Laplante, "A binary integer programming model for global optimization of learning path discovery." in EDM (Workshops). Citeseer, 2014.

[103] N. T. Son, J. Jaafar, I. A. Aziz, and B. N. Anh, "Meta-heuristic algorithms for learning path recommender at mooc," IEEE Access, vol. 9, pp. 59 093–59 107, 2021.

[104] J.-M. Su, S.-S. Tseng, W. Wang, J.-F. Weng, J. T. D. Yang, and W.-N. Tsai, "Learning portfolio analysis and mining for scorm compliant environment," Journal of Educational Technology & Society, vol. 9, no. 1, pp. 262–275, 2006.

[105] T.-C. Hsieh and T.-I. Wang, "A mining-based approach on discovering courses pattern for constructing suitable learning path," Expert systems with applications, vol. 37, no. 6, pp. 4156–4167, 2010.

[106] F. Colace, M. De Santo, and L. Greco, "E-learning and personalized learning path: A proposal based on the adaptive educational hypermedia system." International Journal of Emerging Technologies in Learning, vol. 9, no. 2, 2014.

[107] P. De Bra, L. Aroyo, and A. Cristea, "Adaptive web-based educational hypermedia," in Web Dynamics. Springer, 2004, pp. 387–410.

[108] K. Haruna, M. Akmar Ismail, S. Suhendroyono, D. Damiasih, A. C. Pierewan, H. Chiroma, and T. Herawan, "Context-aware recommender system: A review of recent developmental process and future research direction," Applied Sciences, vol. 7, no. 12, p. 1211, 2017.

[109] L. Peska and P. Vojtas, "Off-line vs. on-line evaluation of recommender systems in small e-commerce," in Proceedings of the 31st ACM Conference on Hypertext and Social Media, 2020, pp. 291–300.

[110] M. Quadrana, "Algorithms for sequence-aware recommender systems," Ph.D. dissertation, Politecnico Di Milano EIB Doctoral Program in Information Technology, 2017.

[111] M. Quadrana, P. Cremonesi, and D. Jannach, "Sequence-aware recommender systems," ACM Computing Surveys (CSUR), vol. 51, no. 4, pp. 1–36, 2018.

[112] A. Gunawardana and G. Shani, "Evaluating recommender systems," in Recommender systems handbook. Springer, 2015, pp. 265–308.

[113] T. Silveira, M. Zhang, X. Lin, Y. Liu, and S. Ma, "How good your recommender system is? a survey on evaluations in recommendation," International Journal of Machine Learning and Cybernetics, vol. 10, no. 5, pp. 813–831, 2019.

[114] P. W. Farris, N. Bendle, P. E. Pfeifer, and D. Reibstein, Marketing metrics: The definitive guide to measuring marketing performance. Pearson Education, 2010.

[115] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," ACM Transactions on Management Information Systems (TMIS), vol. 6, no. 4, pp. 1–19, 2015.

[116] M. Rossetti, F. Stella, and M. Zanker, "Contrasting offline and online results when evaluating recommendation algorithms," in Proc. 10th ACM conference on recommender systems, 2016, pp. 31–34.

[117] J. Beel, M. Genzmehr, S. Langer, A. Nürnberger, and B. Gipp, "A comparative analysis of offline and online evaluations and discussion of research paper recommender system evaluation," in Proceedings of the international workshop on reproducibility and replication in recommender systems evaluation, 2013, pp. 7–14.

[118] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," arXiv preprint arXiv:2010.16061, 2020.

[119] M. Buckland and F. Gey, "The relationship between recall and precision," Journal of the American society for information science, vol. 45, no. 1, pp. 12–19, 1994.

[120] V. Codina, F. Ricci, and L. Ceccaroni, "Distributional semantic pre-filtering in context-aware recommender systems," User Modeling and User-Adapted Interaction, vol. 26, no. 1, pp. 1–32, 2016.

[121] M. Zhou, Z. Ding, J. Tang, and D. Yin, "Micro behaviors: A new perspective in e-commerce recommender systems," in Proceedings of the eleventh ACM international conference on web search and data mining, 2018, pp. 727–735.

[122] Q. Liu, S. Wu, D. Wang, Z. Li, and L. Wang, "Context-aware sequential recommendation," in 2016 IEEE 16th International Conference on Data Mining (ICDM). IEEE, 2016, pp. 1053–1058.

[123] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in Proceedings of the 10th ACM conference on recommender systems, 2016, pp. 233–240.

[124] S. Rendle, W. Krichene, L. Zhang, and J. Anderson, "Neural collaborative filtering vs. matrix factorization revisited," in Fourteenth ACM Conference on Recommender Systems, 2020, pp. 240–248.

[125] S. Robertson, "A new interpretation of average precision," in Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, 2008, pp. 689–690.

[126] K. Haruna, M. Akmar Ismail, D. Damiasih, J. Sutopo, and T. Herawan, "A collaborative approach for research paper recommender system," PloS one, vol. 12, no. 10, p. e0184516, 2017.

[127] T. Ebesu and Y. Fang, "Neural citation network for context-aware citation recommendation," in Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval, 2017, pp. 1093–1096.

[128] A. Doryab, V. Bellotti, A. Yousfi, S. Wu, J. M. Carroll, and A. K. Dey, "If it's convenient: Leveraging context in peer-to-peer variable service transaction recommendations," Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 1, no. 3, pp. 1–28, 2017.

[129] P. Henderson and V. Ferrari, "End-to-end training of object class detectors for mean average precision," in Asian Conference on Computer Vision. Springer, 2016, pp. 198–213.

[130] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in 2018 IEEE International Conference on Data Mining (ICDM). IEEE, 2018, pp. 197–206.

[131] G. Dupret, "Discounted cumulative gain and user decision models," in International Symposium on String Processing and Information Retrieval. Springer, 2011, pp. 2–13.

[132] Z. Kang, C. Peng, and Q. Cheng, "Top-n recommender system via matrix completion," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, no. 1, 2016.

[133] N. Zheng and Q. Li, "A recommender system based on tag and time information for social tagging systems," Expert systems with Applications, vol. 38, no. 4, pp. 4575–4587, 2011.

[134] F. Ai, Y. Chen, Y. Guo, Y. Zhao, Z. Wang, G. Fu, and G. Wang, "Concept-aware deep knowledge tracing and exercise recommendation in an online learning system." in Proceedings of the 12th EDM conference, 2019, pp. 240–245.

[135] L. Song, C. Tekin, and M. Van Der Schaar, "Online learning in large-scale contextual recommender systems," IEEE Transactions on Services Computing, vol. 9, no. 3, pp. 433–445, 2014.

[136] F. Shi, C. Ghedira, and J.-L. Marini, "Context adaptation for smart recommender systems," IT Professional, vol. 17, no. 6, pp. 18–26, 2015.

[137] M. Hasan, T. Hasan, F. Roy, and L. Jamal, "A comprehensive collaborating filtering approach using extended matrix factorization and autoencoder in recommender system," International Journal of Advanced Computer Science and Applications, pp. 505–513, 2019.

[138] C.-N. 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, 2005, pp. 22–32.

[139] Y. Yao, "Measuring retrieval effectiveness based on user preference of documents," Journal of the American Society for Information science, vol. 46, no. 2, pp. 133–145, 1995.

[140] T. Di Noia, V. C. Ostuni, J. Rosati, P. Tomeo, and E. Di Sciascio, "An analysis of users' propensity toward diversity in recommendations," in Proceedings of the 8th ACM Conference on Recommender Systems, 2014, pp. 285–288.

[141] J. Beel and S. Langer, "A comparison of offline evaluations, online evaluations, and user studies in the context of research-paper recommender systems," in International conference on theory and practice of digital libraries. Springer, 2015, pp. 153–168.

[142] M. Monti, "Multicriteria evaluation for top-k and sequence-based recommender systems," http://www.phd-dauin.polito.it/events/phdday2019/monti.pdf, 2019.

[143] C. Su, "Designing and developing a novel hybrid adaptive learning path recommendation system (alprs) for gamification mathematics geometry course," Eurasia Journal of Mathematics, Science and Technology Education, vol. 13, no. 6, pp. 2275–2298, 2017.

[144] A. Jameson, M. C. Willemsen, A. Felfernig, M. De Gemmis, P. Lops, G. Semeraro, and L. Chen, "Human decision making and recommender systems," in Recommender Systems Handbook. Springer, 2015, pp. 611–648.

[145] L. Huang, M. Fu, F. Li, H. Qu, Y. Liu, and W. Chen, "A deep reinforcement learning based long-term recommender system," Knowledge-Based Systems, vol. 213, p. 106706, 2021.

[146] O. Sigaud and O. Buffet, Markov decision processes in artificial intelligence. New York: John Wiley and Sons, 2013.

[147] A. R. Cassandra, "A survey of pomdp applications," in Working notes of AAAI 1998 fall symposium on planning with partially observable Markov decision processes, vol. 1724, 1998.

[148] C. Amato, D. S. Bernstein, and S. Zilberstein, "Optimizing fixed-size stochastic controllers for pomdps and decentralized pomdps," Autonomous Agents and Multi-Agent Systems, vol. 21, no. 3, pp. 293–320, 2010.

[149] S. Young, M. Gašić, B. Thomson, and J. D. Williams, "Pomdp-based statistical spoken dialog systems: A review," Proceedings of the IEEE, vol. 101, no. 5, pp. 1160–1179, 2013.

[150] M. Chen, E. Frazzoli, D. Hsu, and W. S. Lee, "Pomdp-lite for robust robot planning under uncertainty," in 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2016, pp. 5427–5433.

[151] Z. Huang, Q. Liu, C. Zhai, Y. Yin, E. Chen, W. Gao, and G. Hu, "Exploring multi-objective exercise recommendations in online education systems," in Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 1261–1270.

[152] T. Kimura, K. Shiba, C.-C. Chen, M. Sogabe, K. Sakamoto, and T. Sogabe, "Variational quantum circuit-based reinforcement learning for pomdp and experimental implementation," Mathematical Problems in Engineering, vol. 2021, 2021.

[153] S. P. Choi and F. S. Lam, "Modelling the process of learning analytics using a reinforcement learning framework," in Innovations in open and flexible education. Springer, 2018, pp. 243–251.

[154] G. Barata, S. Gama, J. Jorge, and D. Gonçalves, "Early prediction of student profiles based on performance and gaming preferences," IEEE Transactions on Learning Technologies, vol. 9, no. 3, pp. 272–284, 2016.

[155] Y. Choi, Y. Lee, D. Shin, J. Cho, S. Park, S. Lee, J. Baek, C. Bae, B. Kim, and J. Heo, "Ednet: A large-scale hierarchical dataset in education," in Proceedings AIED, 2020, pp. 69–73.

[156] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," BMC genomics, vol. 21, no. 1, pp. 1–13, 2020.

[157] A. Vehtari, A. Gelman, and J. Gabry, "Practical bayesian model evaluation using leave-one-out cross-validation and waic," Statistics and computing, vol. 27, no. 5, pp. 1413–1432, 2017.

[158] B. Vesin, A. Klašnja-Milićević, M. Ivanović, and Z. Budimac, "Applying recommender systems and adaptive hypermedia for e-learning personalizatio," Computing and informatics, vol. 32, no. 3, pp. 629–659, 2013.

[159] B. Taraghi, A. Saranti, M. Ebner, and M. Schön, "Markov chain and classification of difficulty levels enhances the learning path in one digit multiplication," in International Conference on Learning and Collaboration Technologies. Springer, 2014, pp. 322–333.

[160] G. Durand, F. Laplante, and R. Kop, "A learning design recommendation system based on markov decision processes," in 17th ACM SIGKDD conference on knowledge discovery and data mining, 2011.

[161] H. Kurniawati, D. Hsu, and W. S. Lee, "Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces." in Robotics: Science and systems, vol. 2008, 2008.

[162] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of monte carlo tree search methods," IEEE Transactions on Computational Intelligence and AI in games, vol. 4, no. 1, pp. 1–43, 2012.

[163] L. Zou, L. Xia, Z. Ding, J. Song, W. Liu, and D. Yin, "Reinforcement learning to optimize long-term user engagement in recommender systems," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2810–2818.

[164] Y. Wang, "A hybrid recommendation for music based on reinforcement learning," in Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, 2020, pp. 91–103.

[165] J. Pajarinen and J. Peltonen, "Periodic finite state controllers for efficient pomdp and dec-pomdp planning," Advances in neural information processing systems, vol. 24, 2011.

Résumé en Français

Au cours des deux dernières décennies, la technologie Internet a été de plus en plus utilisée dans notre vie quotidienne. Nous faisons des achats en ligne, nous recherchons des informations en utilisant des moteurs de recherche, nous passons une partie importante de notre vie sociale en ligne et nous apprenons même en ligne. Comme presque tous les services, produits, informations, etc. sont à la disposition de chaque utilisateur de l'Internet, nos choix deviennent presque illimités. Par conséquent, le besoin de recommandations efficaces pour aider les utilisateurs à trouver des contenus utiles est apparu.

La tâche des systèmes de recommandation consiste à transformer les activités et les interactions en ligne des utilisateurs en prédictions sur leurs éventuels goûts et intérêts futurs afin de leur fournir des recommandations personnalisées. Le potentiel énorme des systèmes de recommandation a été remarqué pour la première fois par Goldberg et al. à l'avant-garde de la révolution de l'information [1]. La recherche sur les systèmes de recommandation (SR) est au carrefour de plusieurs disciplines telles que l'informatique, la sociologie [2], l'économie [3], la finance [4], l'éducation [5], etc. En outre, bien qu'il s'agisse d'un domaine dominé à l'origine par les informaticiens, la recommandation fait appel à des contributions provenant de diverses directions et constitue désormais un sujet d'intérêt pour les mathématiciens, les physiciens et les psychologues [6].

De nombreux travaux dans la littérature se sont intéressés à la conception de systèmes de recommandation, qui varient selon le type de données utilisées (notes, traces d'activités, description de ressources, etc.), le type d'approche (apprentissage profond, factorisation matricielle, réseaux bayésiens), etc. Plusieurs classifications des RS ont été proposées en fonction de ces éléments : à base de mémoire/à base de modèle, collaboratif/contenu/hybride, etc. [7]. Dans ces classifications, nous pouvons identifier celle qui considère l'horizon temporel du *but* visé par la recommandation : objectif à court terme et objectif à long terme [8, 9].

Les systèmes de recommandation à court terme (ST-RS) fournissent une seule recommandation. Cette recommandation est destinée à être la prochaine action de l'utilisateur, et l'objectif à court terme est généralement la satisfaction de l'utilisateur ; une fois la recommandation adoptée, la satisfaction est donc immédiate (objectif à court terme). Il convient de mentionner que l'impact de la recommandation sur le comportement, les préférences ou la satisfaction futurs de l'utilisateur n'est pas du tout pris en compte.
Par exemple, dans le domaine du commerce électronique, lorsque les utilisateurs font des achats en ligne, ils sont généralement confrontés à de nombreux choix et sont noyés dans le volume des ressources. Cependant, ils cherchent à acheter une ressource ayant le meilleur rapport qualité/prix, compte tenu de leurs préférences. Le système de recommandation recommande une ressource qui répond à ces préférences. Notez ici qu'un ensemble de recommandations peut être fourni, où chaque ressource répond à ces préférences et l'utilisateur peut choisir une ou plusieurs ressources dans cet ensemble, généralement une ressource. Dans le contexte de la musique en ligne, les utilisateurs se voient recommander une chanson ou un ensemble de chansons qu'ils

pourraient aimer, sur la base de leurs préférences ou interactions précédentes. Les ST-RS sont les premiers RS qui ont été proposés dans la littérature. Ils sont largement étudiés dans les domaines de la recherche et de l'industrie [1] et sont toujours d'un grand intérêt [10]. Le terme de court terme a été utilisé pour la première fois dans [11].

Les systèmes de recommandation à long terme (LT-RS) visent à atteindre un objectif qui n'est pas immédiat, mais qui survient plus tard. Cet objectif ne peut généralement pas être satisfait en une seule étape (une recommandation), mais peut être atteint par une série de recommandations. Ainsi, les systèmes de distribution à long terme visent à construire une séquence de ressources (recommandations) qui contribuent à atteindre l'objectif prédéfini, qui ne peut être atteint en une seule étape et qui peut se produire dans un avenir lointain.
Par exemple, lorsque les apprenants étudient en ligne, ils peuvent recevoir une recommandation d'une séquence de ressources pédagogiques conçues pour acquérir plus rapidement plus de connaissances et atteindre un objectif pédagogique. Dans le domaine du patrimoine culturel, une peinture spécifique peut être appréciée par un visiteur débutant uniquement s'il dispose de certaines conditions préalables. Pour s'assurer que ce visiteur apprécie cette peinture, il faut recommander une séquence de recommandations qui peut commencer par une série de peintures faciles d'accès. Ensuite, des tableaux plus complexes peuvent être recommandés et enfin ce tableau spécifique. Notez que même si une séquence complète de ressources est construite par les LT-RS, cette séquence peut être montrée aux utilisateurs soit ressource par ressource, soit la séquence entière en une seule fois.

La séquence de recommandations est définie dans cette thèse de doctorat comme un *chemin* dans le domaine des RS, où :

Definition 6 *Un chemin recommandé est une séquence personnalisée de ressources qui est recommandée à un utilisateur cible pour atteindre un objectif prédéfini, y compris un objectif à long terme [12, 13, 14].*

La littérature s'est concentrée sur les SRL-T plus récemment que sur les SRL-ST [15]. En outre, les ST-RS et les LT-RS sont différents à deux points de vue :

- Les ST-RS recommandent des ressources qui sont généralement uniquement liées au profil de l'utilisateur cible (l'historique de ses interactions, ses préférences), et la fonction objectif est généralement simple, comme la maximisation de la probabilité que l'utilisateur cible aime la ressource. Les SRL recommandent également des ressources qui doivent être liées au profil de l'utilisateur cible, mais qui doivent en outre contribuer à atteindre l'objectif prédéfini. Notez ici que l'objectif peut être éloigné du profil de l'utilisateur en termes de contenu.

- construction d'un chemin qui contribue à atteindre un objectif prédéfini est plus complexe que la recommandation d'une seule ressource ou d'un ensemble de ressources. En effet, le résultat n'est pas un élément unique, mais une séquence d'éléments, où les éléments qui constituent ce chemin doivent être cohérents entre eux, et ce chemin doit être cohérent avec le profil de l'utilisateur et contribuer progressivement à atteindre l'objectif. Pour que ce chemin soit précis, des informations supplémentaires peuvent être nécessaires : sur les utilisateurs, les ressources, etc. [16].

Les LT-RS sont donc très différents des ST-RS, non seulement en termes de résultat, d'objectif, mais ils sont aussi plus complexes. En tant que sujet important et émergent dans les systèmes

de recommandation, les LT-RSs ont encore plusieurs défis à relever, notamment la complexité des modèles et leur évaluation.

Parallèlement à la présence constante d'Internet dans notre vie quotidienne, il est aussi progressivement utilisé dans le domaine de l'éducation. Les plates-formes d'enseignement en ligne, telles que les systèmes de gestion de l'apprentissage (LMS), les cours en ligne ouverts et massifs (MOOC) et leurs outils associés, font désormais partie intégrante des activités de nombreux enseignants et apprenants [13]. Cette adoption généralisée dans l'enseignement se heurte également à une limite : les apprenants sont noyés dans le grand nombre de ressources pédagogiques auxquelles ils peuvent accéder [17]. Cela les empêche de se concentrer sur les ressources d'apprentissage adéquates, c'est-à-dire celles qui correspondent à leur contexte, ce qui limite leur expérience d'apprentissage. En outre, les apprenants peuvent être confrontés à une surcharge cognitive et à une désorientation, notamment lorsqu'ils doivent trouver un équilibre entre le temps d'apprentissage limité disponible et les multiples ressources d'apprentissage. L'enseignement en ligne doit être adapté aux besoins de l'apprenant dans divers scénarios d'apprentissage. En conclusion, l'enseignement en ligne a besoin d'un système de recommandation. En outre, dans l'enseignement traditionnel en face à face, l'enseignement est personnalisé au niveau du groupe : chaque apprenant d'un groupe est enseigné de la même manière. Les RS sont l'occasion de proposer un enseignement plus personnalisé à chaque apprenant, en fonction de son parcours, de ses capacités de mémorisation et de ses performances. Par conséquent, la recommandation de ressources d'apprentissage personnalisées pour les apprenants dans les environnements d'apprentissage en ligne devient une exigence importante [18].

De notre point de vue, un aspect essentiel de l'éducation consiste à organiser les activités d'apprentissage de manière à ce que les apprenants soient guidés vers la bonne ressource, au bon endroit et au bon moment. Dans ce contexte, les systèmes de recommandation pédagogique [19] visent à sélectionner et à recommander des ressources d'apprentissage aux apprenants afin d'améliorer leur expérience d'apprentissage. Cela peut se faire en fournissant les ressources étape par étape ou en fournissant des ressources sous forme de parcours d'apprentissage en E-learning. Ces systèmes de recommandation peuvent s'appuyer sur le profil de l'apprenant cible, qui peut être construit à partir des traces d'interaction de cet apprenant avec le SGA [20, 21], de ses performances académiques [22], etc.

L'apprentissage est une activité de longue haleine. Par exemple, il faut environ dix ans pour devenir médecin ; l'apprentissage d'une langue étrangère nécessite plusieurs années ; l'apprentissage de la cuisine ou du jardinage demande au moins quelques semaines, etc. Dans ces activités, l'objectif d'apprentissage n'est généralement pas immédiat, mais s'inscrit dans la durée. Le RS associé vise donc à atteindre cet objectif en générant et recommandant une séquence de ressources d'apprentissage, un parcours. Par exemple, ce parcours peut viser l'augmentation des connaissances de cet apprenant, sous une contrainte de temps, la réussite à un examen, etc. En éducation, la recommandation d'une séquence de ressources d'apprentissage est appelée système de recommandation de parcours d'apprentissage (LPRS) [23]. Lorsque les apprenants adoptent un parcours d'apprentissage recommandé, ils peuvent atteindre l'objectif à long terme attendu. Par exemple, ce parcours peut permettre à un apprenant donné de maîtriser de nouvelles connaissances, il peut même conduire à la réussite professionnelle, etc. [24, 25, 26]. Un parcours d'apprentissage recommandé peut être défini comme suit :

Definition 7 *Un parcours d'apprentissage recommandé est une séquence personnalisée de ressources d'apprentissage, recommandée à un apprenant cible pour l'aider à atteindre un objectif d'apprentissage prédéfini.*

La réalisation d'un LPRS est une tâche à haut risque. La dimension séquentielle de la recommandation rend le risque encore plus élevé, par rapport aux systèmes traditionnels de recommandation éducative à court terme. En effet, l'adoption d'un parcours d'apprentissage peut avoir des conséquences graves pour l'apprenant, si ce parcours ne correspond pas au profil de l'apprenant ou ne permet pas d'atteindre l'objectif attendu. Par exemple, la conséquence pour les étudiants de l'enseignement supérieur peut entraîner la persévérance scolaire, voire l'abandon de l'école. Par conséquent, il est difficile d'obtenir des SRL précises. Comme pour les RS de chemin ou de séquence, les LPRS peuvent nécessiter une grande quantité d'informations sur les apprenants pour modéliser plus précisément l'impact des ressources sur le niveau de connaissances, les compétences, la motivation, les préférences d'apprentissage, etc. [27].
La littérature a proposé une variété de modèles d'apprenants [28, 29], et de ST-RS éducatifs [19, 30]. Cependant, peu de travaux se sont intéressés aux LPRSs. Selon nous, la recommandation de parcours d'apprentissage n'est pas une direction de recherche scientifique nouvelle mais moins mature, tout en étant centrale dans l'éducation [12], [31], [18], [14]. Un système de recommandation qui exploite uniquement les traces d'interaction des utilisateurs avec le système a une vue partielle de l'environnement et du problème. Même si des informations supplémentaires telles que le contexte, la description du domaine, etc. sont disponibles et utilisées, la vue de la situation reste incomplète et le système ne peut pas être sûr de fournir la meilleure recommandation, ou de prendre la bonne décision. Par exemple, la durée de travail prévue d'un apprenant pour la journée en cours n'est pas connue, savoir si un apprenant est toujours en train d'étudier activement à un moment donné est impossible (sauf si la caméra est allumée), etc. Ce manque d'information se traduit par des informations incertaines, telles que le niveau de concentration, la motivation, le niveau de connaissance, etc. Notez que le manque d'information et l'incertitude ne sont pas des spécificités de l'enseignement en ligne, ce sont aussi des limites de l'enseignement en face à face.
De notre point de vue, la manière de gérer l'information partielle et l'incertitude est une préoccupation importante dans le domaine de l'éducation.

Partant du principe que le RS dans le domaine de l'éducation est difficile à traiter, les capacités de mémoire des apprenants méritent également d'être soulignées. Les capacités de mémoire des apprenants jouent un rôle important dans le processus d'apprentissage : [32]. Il existe de très nombreuses preuves que notre capacité à apprendre/se souvenir s'améliore avec les actions de révision et diminue avec le délai depuis la dernière exposition : la révision permet de mieux appréhender les connaissances des apprenants [33, 34, 35]. La révision joue un rôle crucial dans la conception de l'enseignement, ce qui conduit à un compromis entre l'enseignement de nouvelles connaissances et la révision de ce qui a déjà été enseigné [36]. Notez que l'apprentissage de nouvelles connaissances est généralement basé sur d'anciennes connaissances et nécessite un investissement en temps et des capacités cognitives importantes. En dehors de cela, des centaines d'études ont démontré qu'une révision correctement structurée de la matière au fil du temps produit un apprentissage supérieur à long terme [22].

Nous avons constaté qu'en réalité, la plupart des gens choisissent des stratégies d'apprentissage qui favorisent la mémoire à court terme, ce qui permet d'atteindre l'objectif à court terme de l'apprentissage. Dans les scénarios éducatifs réels, la plupart des apprenants préfèrent apprendre de nouvelles connaissances plutôt que de réviser. Dès 1967, Pimsleur a remarqué qu'il n'y avait pratiquement aucune incitation à faire réviser les apprenants dans les manuels scolaires ou dans la formation des enseignants et des formateurs [32]. La plupart des gens savent que

si l'on veut bien apprendre quelque chose, qu'il s'agisse d'un ensemble de faits, de concepts, de compétences ou de procédures, une seule exposition n'est généralement pas suffisante pour développer une bonne rétention à long terme [35]. Il est d'autant plus vrai que l'amélioration de la mémoire à long terme et la pérennité des connaissances sont des enjeux essentiels pour toute activité d'apprentissage. Cependant, peu d'apprenants remettent en question la valeur de la révision régulière des connaissances acquises précédemment [37]. En outre, même si des centaines d'études ont démontré qu'une révision bien structurée de la matière au fil du temps produit un apprentissage supérieur à long terme, peu d'enseignants ou de chercheurs mettent réellement cette stratégie en pratique pour consolider leurs connaissances dans la satisfaction d'un objectif à long terme.

En psychologie cognitive, une méthode basée sur un modèle de mémoire a été découverte il y a plus de 100 ans [38]. Plus précisément, la révision périodique des connaissances est appelée répétition espacée [39]. Un modèle de répétition espacée bien défini permet aux apprenants d'apprendre davantage, éventuellement en moins de temps [37]. La répétition espacée est efficace car elle permet d'utiliser les phénomènes psychologiques qui contribuent à l'apprentissage et à la mémoire [39].

Bien qu'il n'y ait pas beaucoup de travaux dans ce domaine, sur la base des faits ci-dessus, le rôle de la capacité de mémoire dans l'apprentissage en ligne a attiré de plus en plus l'attention des chercheurs ces dernières années. Ces travaux sont principalement basés sur les caractéristiques suivantes de la capacité de mémoire : (1) la courbe d'oubli montre que nous pouvons prédire quand une personne oubliera des informations ; (2) l'effet d'intervalle montre que lorsque nous prédisons l'oubli, l'apprentissage précédent aura un impact exponentiel sur les avantages de la mémoire ; (3) les résultats du test représentent le principe selon lequel se tester soi-même dans le processus renforcera ces avantages. C'est également l'un de nos axes de recherche.

De notre point de vue, le LPRS est un sujet émergent qui fait encore face à des défis, parmi lesquels la gestion de la partialité et de l'incertitude des informations collectées sur les apprenants, et qui devrait également prendre en compte la mémoire des apprenants, dans le but d'améliorer la précision du parcours d'apprentissage recommandé.

En se rapprochant de l'interaction homme-machine, un bon système de recommandation de flux doit pouvoir contribuer à la satisfaction de l'utilisateur, qui peut se concrétiser par un taux de clics élevé, une acceptation des recommandations proposées, etc. Plus généralement, tout système de recommandation, qu'il soit traditionnel (ST-RS) ou à parcours (LT-RS), ne peut être conçu sans tenir compte de l'évaluation de ses performances. Ceci est également vrai dans le domaine de l'éducation.

Conformément aux méthodologies d'évaluation traditionnelles de la RS, l'évaluation de la recommandation d'une séquence ou d'un parcours d'apprentissage peut être effectuée en ligne ou hors ligne [40]. L'évaluation en ligne est utilisée dans un environnement réel et se concentre sur l'impact des recommandations sur les apprenants. Bien que très instructive, l'évaluation en ligne prend beaucoup de temps, nécessite la disponibilité d'utilisateurs réels et n'est souvent pas complètement reproductible.

À l'opposé, l'évaluation hors ligne se concentre sur la précision des recommandations en s'appuyant sur des ensembles de données statiques des activités d'apprentissage des apprenants et simule des recommandations de parcours. Elle est donc moins coûteuse, ce qui justifie sa popularité, mais fournit une estimation limitée de la précision du modèle de recommandation. L'évaluation hors ligne a été très étudiée pour les ST-RS traditionnels : MAE, RMSE, nDCG,

taux de clics, précision, nouveauté, sérendipité, etc. Bien que l'évaluation hors ligne soit plus facile à réaliser, il est difficile d'évaluer avec précision l'efficacité d'une recommandation de parcours d'apprentissage [41], [25]. Peu de mesures et de cadres d'évaluation ont été proposés pour l'évaluation de séquences de recommandations, notamment pour l'évaluation hors ligne [40].

L'évaluation hors ligne de l'efficacité d'un parcours d'apprentissage recommandé est donc une tâche difficile. De notre point de vue, elle mérite plus d'attention, en particulier parce qu'elle est facile à mettre en œuvre et peut être largement utilisée.

En résumé, avec le développement d'Internet, l'éducation en ligne a suscité de plus en plus d'attention. Comment parvenir à une éducation en ligne efficace nécessite une recherche approfondie sur une nouvelle direction de recherche scientifique : LPRS. La plupart de la littérature s'appuie uniquement sur des modèles statistiques, mais nous pensons que ces modèles pourraient être améliorés s'ils étaient conçus pour prendre en compte les caractéristiques de l'apprentissage, comme la mémoire des apprenants. Par ailleurs, l'évaluation du LPRS est de la plus haute importance mais manque encore de méthodes d'évaluation, notamment hors ligne.

Cette thèse de doctorat met en lumière les corrélations autour du LPRS sous trois angles : combinaison d'algorithmes et modèle de mémoire, application avec des données du monde réel, et nouvelles mesures d'évaluation.