The Case for Graph-based Recommendations

Iulia Popescu*, Kurt Portelli†, Christos Anagnostopoulos*, Nikos Ntarmos*
School of Computing Science, University of Glasgow, Glasgow, G12 8RZ, UK
*{*firstname.lastname@glasgow.ac.uk*}, †k.portelli.1@research.gla.ac.uk*

***Abstract*—Recommender systems have been intensively used to create personalised profiles, which enhance the user experience. In certain areas, such as e-learning, this approach is short-sighted, since each student masters each concept through different means. The progress from one concept to the next, or from one lesson to another, does not necessarily follow a fixed pattern. Given these settings, we can no longer use simple structures (vectors, strings, etc.) to represent each user's interactions with the system, because the sequence of events and their mapping to user's intentions, build up into more complex synergies. As a consequence, we propose a graph-based interpretation of the problem and identify the challenges behind (a) using graphs to model the users' journeys and hence as the input to the recommender system, and (b) producing recommendations in the form of graphs of actions to be taken.**

***Keywords*-Graph-based user representation; Graph-based recommendations; Graph clustering; Graph mining; Recommender systems; Information models; Geo-distributed systems.**

I. INTRODUCTION

Massive Open Online Courses (MOOCs) have risen rapidly in popularity and impact over the last few years; the number of major MOOC providers (e.g., Coursera, Udacity, edX, etc.) and the funding they have attracted since their inception [1], attest to the fact that MOOCs and online education in general are here to stay. MOOC users benefit from accessing a multitude of resources (courses, tutorials) in an affordable and flexible way. However, the abundance of choices (the large number of overlapping courses offered across several MOOCs and/or by several providers) presents a potential that goes untapped, as it becomes difficult to find those courses that best fit a learner's personal learning styles, strengths, weaknesses and interests.

In the pedagogical literature it has been discussed that what works for a subset of students will seldom work for all the students [2]. Frequently, when students fall behind at school or fail to grasp certain concepts, the teacher guides them by adapting the learning material to their personal needs. In an e-learning environment with a large number of students this is not practical. Instead, what we should be doing, is to take advantage of the diversity of the courses offered by different providers and instructors. For example, if a student wants to enhance his/her understanding in a certain field/area, and there are multiple courses that aim to teach the same necessary concepts, a recommender system (RS) can recommend a mixture of (parts of) lessons from several course providers/instructors so as to tailor the content to fit each student's learning style and cognitive skills.

The main motivation for this work lies within the insufficient research on the automated guidance of students [3] and the optimization of their course selection based on their personal needs and preferences [4]. The proposed learning model resembles that of the Swedish School Plan and the Dutch Kunskapsskolan [5], which aim to provide a personalised educational path for each student. This is achieved by setting long term goals, which are also discussed with teachers and parents and then pupils can progress at their own pace and level. Students' progression is carefully supervised by mentors and coaches, who can adjust the personalised curriculum if needed.

II. BACKGROUND

Intensive research [6] has been conducted on RSs that assist users in finding and selecting desirable products and/or services from a vast set of options (e.g., Netflix recommends films a user is likely to watch, Facebook suggests people who might be acquainted to an individual). However, more complex systems, such as course recommenders, are comprised of several features that make the traditional RSs [7] unsuitable. The complexity of the features emerges from the diversity of components, which interact and influence each other, as well as the main output (recommendation).

One of the main drawbacks of using such traditional approaches for producing recommendations relies on the fact that we can only suggest items, or sets of items, that have no prerequisites. In our problem, each item (course) has a list of prerequisites that the user has to have completed before moving on to study the next concept/lesson. Therefore, to successfully cover a knowledge gap, the RS might have to recommend a collection of lessons representing a learning journey, instead of suggesting just one course.

Parameswaran et al. [8], [9] looked into generating recommendations with prerequisites, in which the goal is to recommend a set of items given certain ordering constraints. In [8], three types of prerequisite graphs are presented: *AND*, *OR*, and *Chain*. All three types are relevant for our problem in the following ways. For the *AND* graph, one cannot take a node, unless all parents are taken. To illustrate this, consider this example: a student cannot take Bioinformatics, unless all of Mathematics, Computer Science, and Biology are taken. The *OR* graph might be more common for courses with overlapping content. For example, if a student wants to take Android Software Development, a Java Programming course needs to be taken too. Although, there might exist multiple Java Programming courses offered by different providers, only one of them needs to be taken by the student. The last type of graphs, *Chain*, comprises a set of items ordered precisely. For instance, if one needs to take Math 3, he/she firstly needs to complete Math 1, then Math 2, and then, he/she can move on to Math 3.

Elbadrawy et al. [10] investigated how students can plan their degree by selecting the relevant courses, using nearest neighbour similarity graphs and matrix factorization models. The focus of their research was centred on predicting the grades a student is going to achieve, so that they can make educated decisions on what other courses they should take to improve their overall academic performance. In our case, we are more interested in capturing the concepts a student is struggling with and be able to recommend courses that teach or reinforce those concepts.

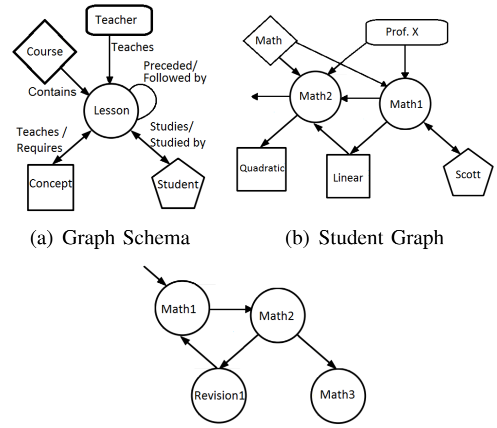
Another study [11] explores how to suggest items in a heterogeneous information network environment, by using various context information (e.g., user feedback) about the entities to increase the quality of recommendations. However, in our problem, we do not often benefit from implicit user feedback and therefore it becomes difficult to provide high quality course recommendations. Instead, we would have to find other context information that is relevant for generating the suggestions.

III. Proposed Solution

We consider an e-learning platform as an environment where students with different backgrounds follow courses online. Courses consist of different activities, such as lessons, tests, questionnaires, etc. In turn, each lesson either *requires/uses* or *teaches/reinforces* certain concepts. The curriculum can thus be modelled as a graph, where nodes are courses/activities and concepts, and edges signify transitions from one activity to the next, or a *require/use/teach/reinforce* relationship to a concept.

To better understand and define the relationships in the graph we provide a graph schema, shown in Figure 1(a). The schema defines the constraints of the graph and the relationships amongst objects. Based on this schema, multiple graph instances can be created. Figure 1(b) depicts the graph instance for Student S_1. It portrays the sequence of lessons he/she studied along with the other relationships. The relationships store labels and weights which are necessary to evaluate how strong the link is.

Course recommendations are then generated using both the sequence of lessons taken and the relationships amongst courses. The purpose of a recommendation is to fill a knowledge gap and sometimes suggesting a single lesson/course does not accomplish this goal. Assuming that a student needs to learn about a new concept, the system might suggest a



(a) Graph Schema (b) Student Graph

(c) Graph Recommendation

Figure 1. Example graph schema and matching student graph instance accompanied by a graph-structured recommendation.

lesson that has a dependency on another lesson. It might also happen that the optimal recommendation contains courses, or pieces of courses, from several providers. Therefore, a suggestion based on a sequence of lessons is not suitable in this scenario. Instead, the recommendation takes the form of a directed graph $G = (V, E, L)$, with the vertices V representing the activities a user should do, the edges E determining the path to follow, and the labels L contain conditions that guide the user through the activities.

A simple scenario of a graph-based recommendation is shown in Figure 1(c). In this case, the RS suggests that the user should start by learning *Math 1* and then move on to *Math 2*. The expressiveness of graphs allows us to either direct the user towards *Math 3* or redirect him/her towards a revision course called *Revision 1* depending on his/her performance. This methodology can be adapted to other systems. For example, a trip builder that plans a vacation with various tourist attractions. If one would structure/output the suggestions as a sequence of items, then it would be cumbersome to re-adapt them if the user's interests change. If a graph-based approach would be used, the user would benefit from the flexibly and scalability of the system, which can receive his/her feedback and alter the points of interest accordingly in near-real-time.

IV. Challenges

The proposed setting opens several research challenges discussed below.

Challenge 1. Each user is represented as an arbitrary graph, and to produce recommendations we need a reliable metric to group users based on their perceived similarity. An initial approach would be to build a distance function between graphs, which determines the similarity level between two or more users. However, finding the optimal metric [12] for clustering the users is an open problem.

Challenge 2. Although, graph similarity and subgraph isomorphism are NP-Complete problems, several algorithms and heuristics have been developed to reduce some instances of the problem to efficiently compute similarity in polynomial time [13]. However, these algorithms will have to be revisited to become capable of producing recommendations in near-time at a massive scale, as the number of graphs to consider will be very large, and the graphs will change frequently over time as students move through the curriculum and interact with the system.

Challenge 3. Producing graph-structured recommendations will not be trivial due to the multiple factors that will be represented: relationships, dependencies, transitions, etc. The expressiveness of graphs will allow us to model the students based on different aspects, such as, social neighbourhood, educational background, aspirations, etc. Therefore, the multiple levels of information can be represented as social graphs, curriculum graphs, learning journey graphs, and concept graphs, which could potentially be combined into hybrid graph models for a broader perspective. This approach permits the development of a personalised curriculum for students, which contains complex interdependencies amongst modules and concepts that cannot be represented using simple data structures (e.g., lists, dictionaries).

Challenge 4. Assuming we have a solution to providing graph-based recommendations for students and courses from a single provider, scaling it to multiple, geo-distributed providers becomes an issue of its own. For example, multiple MOOC providers from different geographic locations might be interested in sharing their courses to increase traffic. Students from one MOOC platform should be able to receive recommendations that include (parts of) courses from other platforms. Adapting the algorithms discussed above to such a setting without breaching privacy constraints (e.g., finding similar students without exchanging their student data) is a formidable task.

V. Conclusion

This paper proposes the use of graph-based recommendations to model users' preferences, interests, and needs by taking into account constraints, such as cognitive skills, status changes, conditional transitions, prerequisites, etc. One of the targeted user groups in our research is comprised by students, who are trying to plan their degree/career by taking the most suitable courses, which would allow them to develop and grow into successful professionals. The innovation of the proposed solution relies on how we intend to structure and model the output of the RS. We made the case for representing the recommendations as graphs because this would allow the RS to output complex, but expressive suggestions, which often rely on items from distinct sets (categories) offered by various providers in a geo-distributed context. Additionally, the proposed solution could potentially be used for extended purposes such as, generating recommendations for educators/instructors, monitoring attendance, enrolment, drop-outs, creating and managing practical/vocational courses, and many more.

Acknowledgements

This research is co-funded by the Erasmus+ Programme of the European Union under the PRIMES project (no. 2016-1-UK01-KA201-024631).

References

[1] J. Keats, "Are MOOCs in danger of becoming irrelevant?" 2016, http://www.wired.co.uk/article/improving-moocs-jonathon-keats.

[2] R. Riding and S. Rayner, *Cognitive styles and learning strategies: Understanding style differences in learning and behavior*. Routledge, 2013.

[3] N. Sonwalkar, "The first adaptive MOOC: Case study on pedagogy framework and scalable cloud architecture – Part I," *MOOCs Forum*, vol. 1, no. P, pp. 22–29, Sep. 2013.

[4] Z. Lustigova and V. Novotna, "Data mining analyses of learning preferences among gifted students," in *The Future of Education*, 2017, p. 200.

[5] O. Eiken, "The Kunskapsskolan (The Knowledge School): A personalised approach to education," OECD Publishing, 2011, https://www.oecd.org/edu/innovation-education/centreforeffectivelearningenvironmentscele/47211890.pdf.

[6] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer US, 2011, ch. 1, pp. 1–35.

[7] M. Kunaver and T. Požrl, "Diversity in recommender systems – a survey," *Knowledge-Based Systems*, vol. 123, pp. 154–162, 2017.

[8] A. G. Parameswaran, H. Garcia-Molina, and J. D. Ullman, "Evaluating, combining and generalizing recommendations with prerequisites," in *Proc. ACM CIKM*, 2010, pp. 919–928.

[9] A. Parameswaran, P. Venetis, and H. Garcia-Molina, "Recommendation systems with complex constraints: A course recommendation perspective," *ACM Trans. Inf. Sys. (TOIS)*, vol. 29, no. 4, p. 20, 2011.

[10] A. Elbadrawy and G. Karypis, "Domain-aware grade prediction and top-n course recommendation," in *Proc. ACM RecSys*, 2016, pp. 183–190.

[11] X. Yu, X. Ren, Y. Sun, B. Sturt, U. Khandelwal, Q. Gu, B. Norick, and J. Han, "Recommendation in heterogeneous information networks with implicit user feedback," in *Proc. ACM RecSys*, 2013, pp. 347–350.

[12] N. D. Phuong *et al.*, "Collaborative filtering with a graph-based similarity measure," in *Proc. IEEE ComManTel*, 2014, pp. 251–256.

[13] T. Gärtner, P. Flach, and S. Wrobel, "On graph kernels: Hardness results and efficient alternatives," in *Proc. COLT/Kernel*, 2003, pp. 129–143.