Towards the Addition of Recommendation to Visualising Learning Dashboards

Solmaz Abdi
The University of Queensland
solmaz.abdi@uq.edu.au

Hassan Khosravi
The University of Queensland
h.khosravi@uq.edu.au

ABSTRACT

There have been significant contributions from the learning analytics community on creating visualisations within student-facing learning dashboards to provide insight and enhance learners' understanding of interactions with learning environments. While learning dashboards have been well-received in the research community, most of the developed dashboards, to date, have had limited ability in providing actionable insight for improving learning. In a separate body of work, inspired by the success of recommender systems, researchers have utilised the digital traces left by learners towards providing recommendation of resources that will assist students in overcoming their shortcomings. Despite the success of recommender systems in many other domains, they have not been well-adopted in the context of higher education. This may be because recommender systems often do not provide rationale for their recommendations. As a potential solution for addressing the twin challenges of visualisation without recommendation and recommendation without justification, we have designed, implemented, and validated an open-source student-facing learning platform called RiPPLE that couples visualisation and recommendation. We have evaluated the approach using syntheticdata sets. Our results indicate that RiPPLE can provide accurate personalised and justified recommendation for learners.

KEYWORDS

Visualisations, Recommender Systems, Learning Dashboards

1 INTRODUCTION

The field of visualisation has been utilised broadly to allow users to employ a variety of visual displays to explore and interpret their data [7]. With the increase in the use of educational technologies and the advancements in the areas of learning analytics and educational data mining, a new field, commonly known as "Learning Dashboards" has emerged to help make sense of data sets in learning and education [5]. Based on a recent comprehensive survey on learning dashboards[9], "A learning dashboard is a single display that aggregates different indicators about learner(s), learning process(es) and/or learning context(s) into one or multiple visualisations." Learning dashboards may be reviewed in a variety of contexts including their intention, target audience, data sources, visualisation choice, and comparison choices.

While learning dashboards have been successful in enabling learners to better trace their interactions with a variety of learning environments, their ability in providing actionable insight and recommendation on how learners can overcome their shortcomings is quite limited [10]. As such, learners often find it challenging to see how learning dashboards can have a positive impact on their actual learning [1].

Recommender System for Technology Enhanced Learning (RecSysTEL) is an active and rapidly evolving research field that harness the digital traces left by learners through the use of educational tools and technologies to provide recommendations. In recent studies, [3] performed an extensive classification of 82 different RecSysTEL environments, and [6] reviewed the various evaluation strategies that have been applied in the field.

Despite extensive theoretical work spanning over 15 years in the area, RecSysTELs have not been well-adopted in higher education. A potential reason is that most recommender systems operate as a "black box" and give users no insight into the rationale of their choice [13]. While this may not be an issue when recommending movies, for instance, it does seem to challenge the core mission of many universities, which is to to develop critical thinkers that can effectively articulate their strengths and have an appreciation for inquiry and investigation [8].

As a potential solution for addressing the twin challenges of visualisation without recommendation and recommendation without justification, we have designed, implemented, and validated a novel open-source course-level platform called RiPPLE (Recommendation in Personalised Peer Learning Environments) that couples the visualisation and recommendation components. RiPPLE maintains a repository of tagged multiple-choice questions (MCQs) that students can use for assessing and enhancing their learning. RiPPLE mines data collected by the platform to approximate students' competencies. Data is presented to the students through a goal oriented visualisation widget that enables them to view and compare their competencies based on their personal preferences for visualisation. RiPPLE then recommends personalised questions based on students' competencies, which can assist them in overcoming their knowledge gaps.

The approach is evaluated using synthetic data set. The synthetic data sets are generated using a 2-parameter logistic Latent Trait Model from classical Item Response Theory [4], as recommended by [2]. Our results indicate that RiPPLE is scalable and can provide insightful visualisations coupled with accurate, personalised recommendations targeting the knowledge gaps of learners.

The rest of this paper is organised as follows: Section 2 presents the proposed learning platform, RiPPLE. Section 3 presents the conducted experiments on synthetic data sets. Finally, Section 4 presents concluding notes and a discussion on the current limitations of this work.+

2 THE PROPOSED LEARNING PLATFORM

RiPPLE provides a course-level, student-facing platform for the self-selection of formative assessments. RiPPLE maintains a repository of tagged multiple-choice questions (MCQs) that students can use for assessing and enhancing their learning. RiPPLE mines data collected by the system to approximate students' competencies, visualise students' competencies through an interactive visualisation widget, and recommend personalised questions that assist learners in overcoming their knowledge gaps. RiPPLE allows learners to track their performance while receiving personalised recommendation on what questions they should do next.

The remainder of this section is organised as follows: Section 2.1 introduces a formal problem statement. Section 2.2 discusses how competencies are approximated and visualised, and Section 2.3 discusses how personalised recommendations are computed.

2.1 Problem Formulation

A problem statement using formal notation is presented in this section.

Suppose we have N users $U = \{u_1, u_2, ..., u_N\}$ enrolled in a specific course, M multiple choice questions $Q = \{q_1, q_2, ..., q_M\}$ contributed to the course, and L different topics related to the course. We define an Answer matrix $A_{N\times M}$ that determines the correctness of the answered questions by the users: if user u has answered question i correctly then $a_{ui} = 1$ otherwise it is 0. A Difficulty matrix, denoted by $D_{N\times M}$, shows the expressed difficulty level of questions answered by users. In this matrix, d_{ui} determines the difficulty level of question i for user u and is a value in range of 0 to 1. A Tag matrix, denoted by $T_{M\times L}$, determines the associated tags of each question. In this matrix, there are 0 to L possible tags for each question. $t_{ij} = 0$ indicates that question i is not tagged with topic j and $t_{ij} = \frac{1}{g}$ shows that question i is tagged with $1 \leq g \leq L$ associated topics including j.

We aim to learn:

- A student-topic learning profile $LP_{N\times L}$, where lp_u is an approximation of user u's competencies. This matrix will be used for visualising students' competencies.
- A student-question matrix $O_{N\times M}$ such that o_{ui} demonstrates the personalised rating of question i for user u. This matrix will be used for recommending questions.

2.2 Visualisation

2.2.1 Interface. The interface of the platform,provides a goal oriented visualisation widget that enable learners to select their desired visualisation that better suits their comprehension and personal preference. The visualisation widget also allows learners to compare their performance against a range of options such as comparing their performance with their own predetermined goals, compare their performance with a selected distribution (e.g. top 20%) of the peers and compare their performance with a selected distribution of learners previously enrolled in the course.

2.2.2 Algorithm. The definition of the learning profile was first introduced in [11]. Three main steps used for generating the learning profile, which is used for approximating and visualising competencies as follows:

In the first step, $D_{N\times M}$ is used for computing a vector $\bar{d}_M$ that stores the average difficulty of each question across all of the users. The average difficulty of a question i may be computed using the following formula:

$$\bar{d}_i = \frac{\sum_{(u,i)\in D_{ds}} d_{ui}}{dn_i}$$

where $(u, i) \in D_{ds}$ represents (u, i) pairs such that the difficulty rating of user u for question i is presented in the data set and dn_i denotes the number of users that have rated the difficulty level of question i.

In the second step, $A_{N\times M}$ and $\bar{d}_M$ are used in a scoring function that maps user performance to competencies using:

$$g_{ui} = (1 - a_{ui})\left(\frac{0.5 - a_{ui}}{1+ \bar{d}_i}\right) + a_{ui}\left(\frac{0.5 - a_{ui}}{2- \bar{d}_i}\right)$$

where g_{ui} determines user u's lack of competence (knowledge gap) on question i, independent of their performance on other questions. The first part of the equation is positive, indicating a knowledge gap for an incorrectly answered question i weighted by $\hat{d}_i$. The second part contributes to the score with a negative value, indicating competencies, when the question is answered correctly.

In the third step, a student-topic learning profile $LP_{N\times L}$ is produced, where lp_u is an approximation of user u's competencies across all of the topics associated with the course. lp_{uj}<0 shows that u has demonstrated some knowledge on topic j, lp_{uj}>0 represents that u has demonstrated some lack of knowledge on topic j and 0 is neutral. $LP_{N\times M}$ is produced using the following formula:

$$LP_{N\times L} = \frac{N \times M \times T_{M\times L}}{S_{N\times M} \times T_{M\times L}}$$

where $s_{ui} = 1$ if user u has attempted question i and 0 otherwise. In this formula, the numerator provides unnormalised topic-level knowledge gap, which are normalised by the denominator.

LP is used as the data source for visualisation of competencies and knowledge gaps.

2.3 Recommendation

2.3.1 Interface. The interface of the RiPPLE enables learners to select questions using search and recommendation functionalities. Learners can sort questions based on their difficulty, quality, number of responses, number of comments or personalised rating. By selecting "Personalised Rating", the platform sorts the questions based on the outcome of recommender system. Moreover, learner can search for questions based on specific content that may be present in the questions or multiple choice answers.

The results of the search are presented as a list of question cards, allowing users to engage with questions that best suit their needs.

2.3.2 Algorithm. The three main steps used for generating the student-question matrix O are as follows:

The first step uses $LP_{N\times L}$ and $T_{M\times L}$ to produce an updated student-question matrix $P_{N\times M}$ using

$$P_{N\times M} = LP_{N\times L} \times T^{\mathsf{T}}_{N\times L}$$

in which p_{ui} approximates user u's knowledge gap of question i based on lp_u and the tags associated with i.

Step two employs matrix factorisation [12] to characterise users and questions using vectors of latent factors that extends $G_{N\times M}$ to form $\hat{G}_{N\times M}$, where $\hat{g}_{ui}$ predicts user u's lack of knowledge about a question i that has not been attempted by u.

In step three $\hat{G}_{N\times M}$ and $LP_{N\times L}$ are combined using

$$O_{N\times M} = \hat{G}_{N\times M} + \beta P_{N\times M}$$

where β is a parameter controlling the impact of the learning profile, which may be determined using a validation set. Matrix $O_{N\times M}$ is the output of the algorithm, in which o_{ui} represents the personalised rating of question i for user u tailored towards their knowledge gaps. Providing personalised recommendations from O is trivial. For a user u, the platform can find and recommend question i such that $o_{ui} = Max(o_{u1}, o_{u2} \ldots o_{uN})$ in linear time. In a similar fashion, by sorting questions based on their personalised rating from O_u, a list of k questions with the highest personalised rating may be recommended to learner u.

3 EXPERIMENTS USING SYNTHETIC DATA SETS

In this section, synthetic data sets are used for evaluating RiPPLE. Section 3.1 provides information on the experimental environment setup. Section 3.2 evaluates the scalability of the platform and Section 3.3 evaluates the accuracy of the recommendations under different settings.

3.1 Synthetic Data Experimental Environment Setup

Data sets. We use synthetic data set including a list of pre-defined questions with different topics and a list of users with their related attributes such as their pre-defined knowledge gaps generated based on the classical Item Response Theory [4].

Evaluation metric. Accuracy is used to validate the quality of the recommendations. This metric demonstrates the extend to which recommendations are effective in targeting users' biggest knowledge gaps using the following formula:

$$Accuracy = \frac{match}{|ds|}$$

where ds is the set of all pairs of (u, i) in the data set and $match$ is the number of instances $\in ds$ where the topic of the recommendation matches a student's most significant knowledge gap.

Parameter settings. In all experiments the parameters are set using the following default values unless otherwise stated: $N = 400$, $M = 1100$, L = 10, $\alpha = 0.1$, $\beta = 0.1$.

In each set of experiments f is considered as a second parameter as it can illustrate the use of the platform under two different interpretations: (1) different stages of the course where a smaller value of f indicates an earlier stage of the course and (2) different levels of learner' engagement, where a smaller value of f represents a lower level of engagement. The results of running all of the experiments for f in 0.01, 0.05, 0.1, 0.15, 0.25, 0.5 and 0.7 are reported.

3.2 Evaluating the Scalability of RiPPLE

The runtime of RiPPLE with respect to different number of learners that simultaneously seek recommendations (P) under different settings for f are reported in Figure 1. It is observed that regardless of the value of f, with the increase of P, runtime increases.

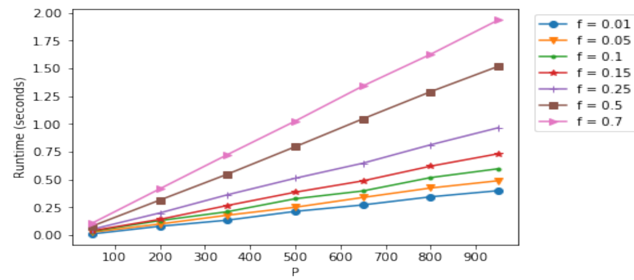

Figure 1: Changes in runtime as the number of learners that simultaneously seek recommendations is increased

As f is increased, the slope of runtime increment with respect to P becomes steeper. A strong correlation is observed between the increment of P and f on the steepness of runtime slope. The overall runtime of the platform while providing recommendations may be reduced using caching techniques, which will be investigated in future studies.

3.3 Evaluating the Accuracy of the Recommendations in RiPPLE

In this section, the impact of varying data set generation and model parameters on the accuracy of the recommendations is analysed.

Impact of the sparsity on accuracy (α). Figure 2 presents the accuracy of the recommendations with respect to different values of α under different settings for f. For smaller values of α, which correspond to learners with sparser knowledge gaps, the system is able to generate highly accurate recommendations. As α is increased and learners with less-extreme predefined knowledge gaps are generated, the accuracy of the recommendations drop.

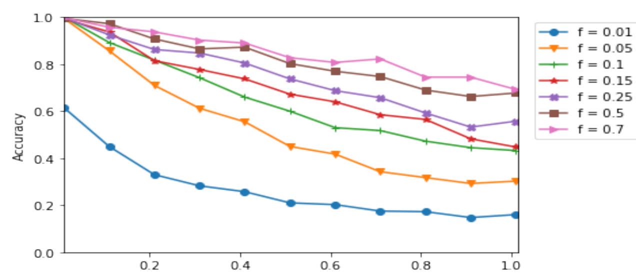

Figure 2: Impact of the sparsity of the pre-defined knowledge gaps on accuracy

As expected, the increase in f leads to increase in accuracy. For $f = 0.01$ and $f = 0.05$, the accuracy is lower due to the fact that RiPPLE has still insufficient information about learners. As f is increased to more than 0.1, RiPPLE is able to provide more accurate recommendations. For $f > 0.15$, which can be considered as the threshold for having an acceptable accuracy, further increase of f does not contribute significantly to the improvement of the accuracy.

Impact of the learning profile (β) on accuracy. Figure 3 presents how the accuracy of recommendations is changed as β is increased under different settings for f. In this experiment, $\beta = 0$ implies that recommendations are provided without the contribution of learning profile, which means that the system has no information about the relationship between topics and questions. So, regardless of f, the accuracy of recommendations is very low and is almost 10%. This result is expected for ten topics and shows the random nature of recommendations. By a very slight increasing of β to 0.05, accuracy is increased significantly. Use of $\beta > 0.05$ has no significant impact on the accuracy.

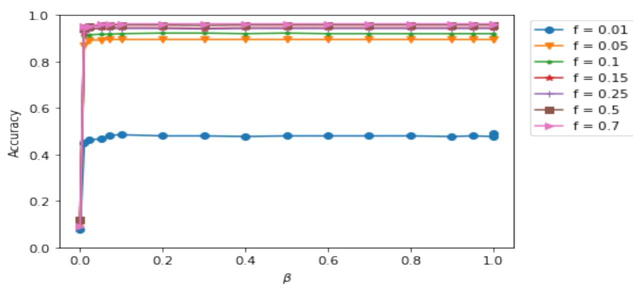

Figure 3: Impact of the learning profile on accuracy

Except for $f = 0.01$ where the accuracy always remains below 40%, an increase of f has no considerable impact on the accuracy of recommendations.

The results showed that the value of β has a considerable impact on the accuracy of recommendations.Also, for $f > 0.01$ no strong correlation is observed between the values of β and f.

Impact of number of topics on accuracy (L). Figure 4 shows the accuracy of recommendations with respect to different values of L under different settings for f. Experimental results for all values of f indicates that for $L < 10$, the accuracy is always above 80%. For $10 < L < 20$ which is the common number of topics for regular taught courses, accuracy decreases, but, the results are still reliable, showing the system's ability to target learners' biggest knowledge gaps. By increasing L to more than 20 it becomes more challenging to provide accurate recommendations and accuracy drops significantly.

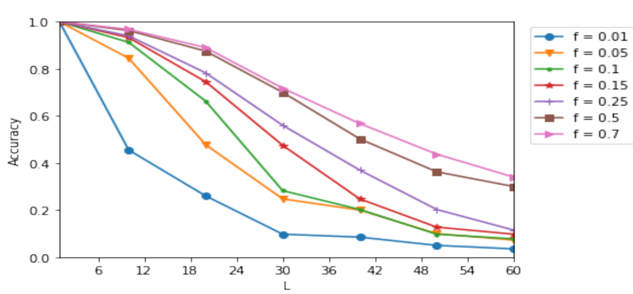

Figure 4: Impact of the number of topics on accuracy

For $L < 10$, f has no significant impact on the accuracy. By increasing L from 10 to 20, for $f < 0.05$, the accuracy drops sharply, while, for $f > 0.1$ the accuracy is always above 80%. For $L > 20$ a similar behaviour is observed with respect to all values of f, which is that the accuracy decline rate is more significant.

4 CONCLUSIONS

A novel student-facing learning platform called RiPPLE that couples visualisation and recommendation was introduced. As a first step, RiPPLE uses information on the learners, questions, and learner's responses to the questions alongside a scoring function to generate a learning profile. This profile is used to approximate learners' competencies and knowledge gaps. The learning profile is presented through a goal oriented visualisation widget that enables learners to view and compare their competencies based on their personal preferences. Finally, a recommendation engine employing matrix factorisation uses learners' responses and the learning profile to provide learners with personalised questions that best help them overcoming their knowledge gaps.

Experimental validation of RiPPLE used synthetic data set. The synthetic data sets assess the behaviour of RiPPLE under diverse circumstances. Our results indicate that RiPPLE can provide accurate, personalised, and justified recommendations to learners.

Our goal for the future is to validate the platform with A/B testing and a control group that would receive random recommendations and an experimental group that would receive targeted recommendations from RiPPLE. This experiment would allow us to determine whether or not the recommendations lead to measurable gains.

REFERENCES

[1] Outmane Bourkoukou, Essaid El Bachari, and Mohamed El Adnani. 2017. A Recommender Model in E-learning Environment. *Arabian Journal for Science and Engineering* 42, 2 (2017), 607–617.
[2] Michel C Desmarais and Ildiko Pelczer. 2010. On the faithfulness of simulated student performance data. In *Educational Data Mining 2010*.
[3] Hendrik Drachsler, Katrien Verbert, Olga C Santos, and Nikos Manouselis. 2015. Panorama of recommender systems to support learning. In *Recommender systems handbook*. Springer, 421–451.
[4] Fritz Drasgow and Charles L Hulin. 1990. Item response theory. (1990).
[5] Erik Duval. 2011. Attention please!: learning analytics for visualization and recommendation. In *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. ACM, 9–17.
[6] Mojisola Erdt, Alejandro Fernández, and Christoph Rensing. 2015. Evaluating Recommender Systems for Technology Enhanced Learning: A Quantitative Survey. *IEEE Transactions on Learning Technologies* 8, 4 (2015), 326–344.
[7] Usama M Fayyad, Andreas Wierse, and Georges G Grinstein. 2002. *Information visualization in data mining and knowledge discovery*. Morgan Kaufmann.
[8] Diane F Halpern. 1999. Teaching for critical thinking: Helping college students develop the skills and dispositions of a critical thinker. *New directions for teaching and learning* 1999, 80 (1999), 69–74.
[9] Adrian Holzer, Andrii Voznuik, Denis Gillet, Maria Jesús Rodríguez-Triana, Beat A Schwendimann, Luis P Prieto, Mina Shirvani Boroujeni, and Pierre Dillenbourg. 2016. Perceiving learning at a glance: A systematic literature review of learning dashboard research. *IEEE Transactions on Learning Technologies* (2016).
[10] Ioana Jivet, Maren Scheffel, Hendrik Drachsler, and Marcus Specht. 2017. Awareness is not enough. Pitfalls of learning analytics dashboards in the educational practice. (2017).
[11] Hassan Khosravi, Kendra Cooper, and Kirsty Kitto. 2017. RiPLE: Recommendation in Peer-Learning Environments Based on Knowledge Gaps and Interests. *JEDM-Journal of Educational Data Mining* 9, 1 (2017), 42–67.
[12] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 426–434.
[13] Katrien Verbert, Denis Parra, Peter Brusilovsky, and Erik Duval. 2013. Visualizing recommendations to support exploration, transparency and controllability. In *Proceedings of the 2013 international conference on Intelligent user interfaces*. ACM, 351–362.