What Should I Do Next? Adaptive Sequencing in the Context of Open Social Student Modeling

Submitted in fulfillment of the requirement of
the ISSP 2990 Independent Study course

By:
Roya Hosseini

Advised By:
Dr. Peter Brusilovsky

Intelligent Systems Program
University of Pittsburgh
Pittsburgh, USA

April 28, 2015

Contents

Collaboraters

The present work was done in collaboration with I-Han Hsiao[1] Julio Guerra[2].

[1]School of Computing, Informatics & Decision Systems Engineering, Arizona State University, Tempe, AZ, USA

[2]School of Information Sciences, University of Pittsburgh, Pittsburgh, PA, USA

Abstract

One of the original goals of intelligent educational systems is to guide every student to the most appropriate educational content. In our past work, we explored both knowledge-based and social guidance approaches and learned that each of these approaches have weak sides. In this paper we explored an idea of combining social guidance with more traditional knowledge-based guidance in a hope to support more optimal content navigation. We proposed a greedy sequencing approach aiming at maximizing student's level of knowledge and implemented it in the context of an open social student modeling interface. We performed a classroom study examining the impact of this combined guidance approach. The results of a classroom study shows that greedy guidance approach positively affected students' navigation, increased the speed of learning for strong students, and improved the performance of students, both in the system and end-of-course assessments.

keywords: personalized guidance, open social student modeling, adaptive navigation support, E-learning, Java programming

1

Introduction

One of the original goals of intelligent educational systems is to guide every student to the most appropriate educational content. Starting with the first reported ITS system SCHOLAR [8], a range of knowledge-based guidance technologies were reported. Different technologies in this group were known as instructional planning [1], course sequencing [4], course generation [14], and adaptive navigation support. All these knowledge-based approaches were based on the same principle: using a combination of domain models, course goals, and overlay student models, the sequencing engine decided which content is the most appropriate for an individual student at every given moment and delivered it to the student through the interface either bringing the student to the right content directly (as in sequencing) or through suggested links (as in course generation and navigation support). Despite the known power of this technology, its applications are still rare due to large amount of efforts required to build the domain models and analyse contents.

In our recent research we discovered and evaluated a new approach to guide students to the "right" content based on the ideas of open social student modeling (OSSM) [12]. OSSM is a recent expansion of open student modeling (OSM), a popular approach that makes traditionally hidden student models available to the learners for exploration [6, **?**, **?**]. OSM is known for its ability to increase student engagement, motivation, and knowledge reflection. The idea of OSSM is to enhance its cognitive aspects with social aspects by allowing students to explore each other models or cumulative model of the class [5]. In our studies we explored several versions of visual OSSM based on comparative visualization of the student's own open knowledge model and the models of students with similar learning goals. While our original motivation was to increase student engagement, which is a known value of social approaches, the studies also demonstrated the navigation support power of OSSM. It was able to guide students to most appropriate self-assessment problems [12] almost as efficiently as the knowledge-based guidance that we explored in the past [10]. Since the main power of OSSM came from the community of learners, it also required considerably simpler domain and user models to be efficient. Yet, the studies also revealed that OSSM approach makes students more conservative in their work with content, which decreases the 'personalization' power of 'social' guidance.

The work reported in this paper explores an idea of combining social guidance with more traditional knowledge-based guidance in a hope to support more optimal content navigation. This idea was motivated by the success of hybrid approaches in recommender systems that demonstrated several efficient ways to combine content-based and collaborative filtering approaches [7]. We introduce a greedy sequencing approach for selecting learning activities that could maximize student's level of knowledge and demonstrate how this approach could be implemented in the context of OSSM. We also present a classroom study examining the added impact of this combined guidance approach.

The remainder of this paper presents the sequencing approach and the implementation of that in the OSSM interface and reports the results of the evaluation. We conclude with a discussion of the results and plans for the future work.

2

Adaptive Sequencing in the Context of OSSM

In our study, adaptive guidance was implemented in the context of a specific OSSM interface called Mastery Grids. To explain the technology, we start with a brief presentation of Mastery Grids, follow by explaining how the suggestions generated by the sequencing algorithm were added to the OSSM interface and finally explain the details of our specific sequencing approach that we call Greedy Sequencing.

2.1 Mastery Grids, an OSSM Interface

Mastery Grids is an OSSM interface that combines visual open student model presentation with the interface to access online course materials. The design of Mastery Grids was informed by our earlier studies of OSSM [12] where we discovered that students achieve higher success rates and get more engaged with non-mandatory contents in the presence of OSSM. The first version of Mastery Grids confirmed these effects in a classroom study [13].

Figure 2.1 shows a screenshot of Mastery Grids. The system organizes course contents into topics, displayed as columns of the grid. The first row shows topic-by-topic knowledge progress of the current student by using green colors of different density, the darker the higher the progress. The third row shows the aggregated progress of the rest of the students of the class in shades of orange. The second row presents a differential color comparing the students progress and the class progress. For example, in Figure 2.1 the student has a higher progress than the class in most of the topics where the cells in the second row are green, but the class is more advanced in two of the topics (13th and 20th column) where the cells in the second row are orange. The student has same progress as the class in four topics with light gray color (11th, 15th, 18th, and 19th column). By clicking in cells, the student can access the content inside the topic. For example, in Figure

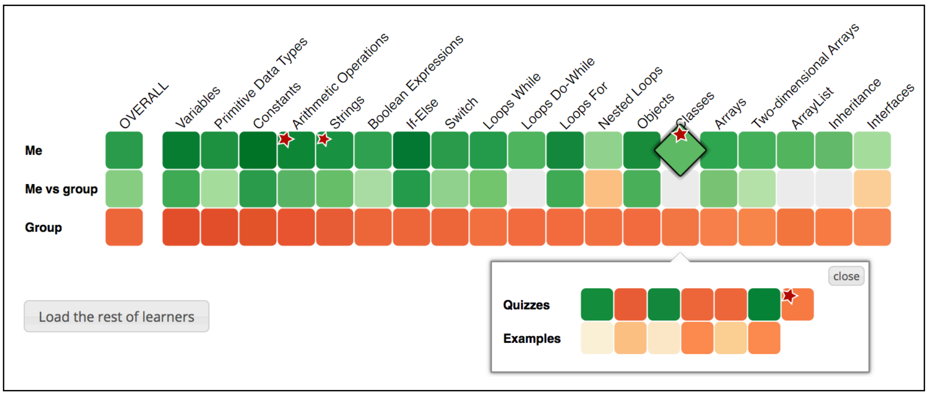

Figure 2.1: The presentation of recommendations in the context of Mastery Grids' OSSM interface, a cell with a star symbol represents a recommended item

2.1, the student has clicked the topic *Classes* and the system displays cells to access questions and examples related with this topic. Additionally, by clicking the button "Load the rest of learners", an anonymized ranked list of individual student models is shown in a grid form (Figure 2.2).

2.2 Enhancing OSSM Interface with Sequencing

To implement adaptive sequencing in the context of Mastery Grids interface, we used top three content item recommendations generated by the adaptive sequencing approach and displayed their presence in the topic using red stars that appear on both, recommended items and their containing topics. The size of the stars shows the position of the recommended items in the top − 3 list. Note that our approach to *sequencing* is consistent with the navigation support nature of the interface: it does not force students to go to the sequenced content, but informs the students and helps them to make their next navigational step. The resulting interface combines the social guidance of OSSM with the personal guidance provided by sequencing.

2.3 Greedy Sequencing

The intelligence behind the sequencing interface is provided by a sequencing algorithm that we call *Greedy Sequencing* (GS). This algorithm was specifically developed to compensate the conforming nature of OSSM on student navigation. The goal of GS is to guide student in the space of learning materials by proactively recommending student activities that could *maximize* the chance to gain new knowledge while avoiding content that is too com-

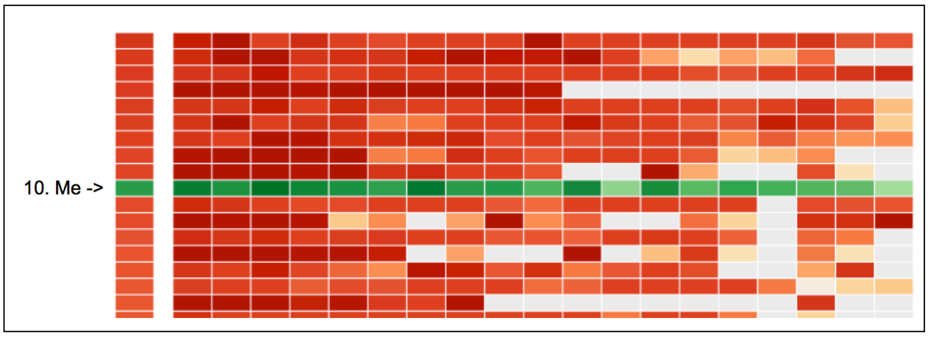

Figure 2.2: List of peer models ordered by progress on topics in the course. The student with the highest progress appears in the top of the list. This list is anonymized and current student can see herself in the position that is obtained according to her topic-based progress (here, the student, shown by "Me", is in position 10)

plex to comprehend. As other knowledge-based sequencing approaches, GS utilizes information about concepts associated with content, more specifically, prerequisite and outcome concepts for each activity. Prerequisites are the concepts that students need to master before starting to work with the activity. Outcomes are the concepts that are being learned in the process of work with the activity. In our work all concepts associated with an activity were determined using our concept parser [9]. The parser indexes the activities with concepts of Java ontology[1]. The extracted concepts for each activity are then separated into prerequisites and outcomes. We marked a concept in the activity as prerequisite if it has appeared in the prior topics, and as outcome if it is the first topic where it appears.

The GS algorithm ranks activities by balancing the knowledge level of student in the prerequisite concepts and the knowledge that can be gained from the outcome concepts. The rank of an activity is calculated using (2.1) based on student's level of knowledge in prerequisite and outcome concepts of that activity:

$$R = \frac{n_p P + n_o O}{n_p + n_o} \quad (2.1) \qquad P = \frac{\sum_i^{n_p} k_i w_i}{\sum_i^{n_p} w_i} \quad (2.2) \qquad O = \frac{\sum_i^{n_o} (1 - k_i)\, w_i}{\sum_i^{n_o} w_i} \quad (2.3)$$

where n_p and n_o are the number of prerequisite and outcome concepts in the activity, respectively; P represents the ratio of known prerequisites and is the weighted average of student's knowledge in the prerequisite concepts of the activity; and O represents the ratio of unknown outcomes and is the amount not learned in each of the outcome concepts. These two ratios can

[1] http://www.sis.pitt.edu/~paws/ont/java.owl

be calculated using (2.2) and (2.3), respectively.

In (2.2) and (2.3), k_i is the knowledge level of the student in the concept i, has the minimum value of 0 (no knowledge) and asymptotically reaches 1 (maximum knowledge). The term $1 - k_i$ in (2.3) is the amount that is not learned in the outcome concept. The w_i is the smoothed weight of the concept obtained by performing *log* function on TF-IDF values of the concepts. The rank R of an activity is in the interval $[0, 1]$ with 1 representing the highest rank.

3

Study Design

To explore the effect of GS on student navigation and performance, we ran a classroom study in an undergraduate course of Object-Oriented Programming & Data Structures offered by the Computer Science Engineering program in the Arizona State University during Fall 2014. The course focused on data structures with Java. In this course, Mastery Grids interface extended with GS was used to access supplementary course materials. In total, 143 students were enrolled in the course. They were all informed by the instructor about the nature of learning contents that could be accessed using Mastery Grids. The instructor encouraged students to explore these contents, but indicated that the use of this system was non-mandatory.

To investigate how students navigate with and without the presence of the sequencing, we split the course into two parts. Part 1, from Aug. 21 to Sep. 25, used Mastery Grids system with no sequencing. In part 2, from Sep. 26 to Oct. 21, sequencing was enabled. In the beginning of the course students took a pretest evaluating their initial knowledge of Java programming concepts. To measure students' knowledge gain, a posttest was administered on Oct. 21. The pretest and posttest had same set of questions and the score ranges from 0 to 21. At the end of the semester we collected questionnaires that asked students to report their opinion about the sequencing in the Mastery Grids system.

The learning materials in the course included parameterized questions on the semantics of Java, administered by the QuizJET system [10], and annotated code examples, administered by the Webex system. The parameterized nature of semantics questions allowed students to attempt the same question several times, each time with a different parameter. As a result the correct answer is different across attempts on the same question. An annotated code example is a complete program that has expert's annotation (comments and explanations) for some code lines. The annotations could be interactively explored by clicking on annotated lines. The learning materials were organized into topics defined by the course instructor. Overall, there were 111 questions and 103 examples spread over 19 topics in the course.

4

Evaluation

We collected student logs for the analysis period between the pretest and the posttest. The data consisted of students' attempts on topics and activities as well as information showing whether attempted topics and activities (questions or examples) were recommended by the system or not. We removed from the data all sessions with duration less than 30 seconds. Then, we excluded students that were not sufficiently active in the system by discarding data of those who had less than 30 attempts on questions, i.e. about $\frac{1}{4}th$ of available questions. In total, there were 86 students using the system during the analysis period. Out of this number, there were 21 students with no attempt to solve questions and 12 students with less than 30 attempts on questions. After discarding less active students, we had the data of 53 students for our analysis.

4.1 Navigational Pattern Analysis

While OSSM interface demonstrated good ability to move the students timely along the common path through the topic sequence, the goal of the GS algorithm was to help the students in breaking out from the common path when it is personally beneficial, not staying too long on already sufficiently mastered topics, while also making sure that knowledge from the past topics are mastered. To see to what extent the GS encouraged non-sequential navigation, we classified students' moves from current to next activity into four groups (patterns):

- Within-Topic: moving between activities in the same topic
- Next-Topic: moving from an activity in a topic to the activity in the next topic (according to the sequence of topics in the course)
- Jump-Forward: jumping to an activity in a topic two or more steps further

Table 4.1: Frequencies of the four topic-based navigational patterns in three contexts

Pattern	Part 1	Part 2-N	Part 2-R
Within-Topic	1801	4569	451
Next-Topic	431	689	189
Jump-Forward	216	287	162
Jump-Backward	219	328	161
Total	2667	5873	963

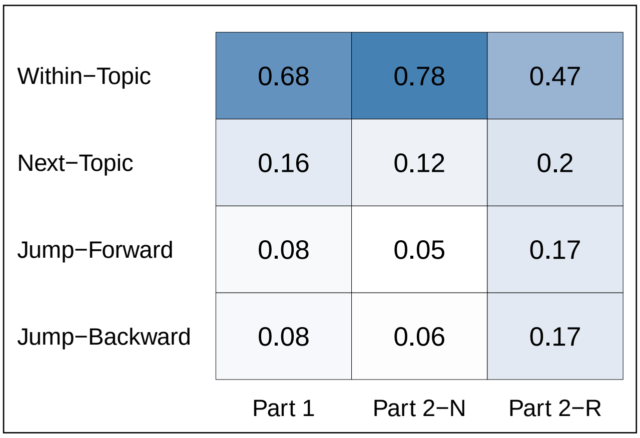

Figure 4.1: Relative frequencies of four topic-based navigational patterns in part 1, not recommended items in part 2-N, and recommended items in part 2-R

- Jump-Backward: jumping to an activity in an earlier topic

The Within-Topic and Next-Topic groups represent sequential navigation and the Jump-Forward and Jump-Backward groups represent non-sequential navigation.

Table 4.1 shows the frequency of each pattern in the three contexts: part 1 and part 2 separating student navigation to *not-recommended* (part 2-N) and *recommended* activities (part 2-R). Relative frequencies of the four patterns in each context are shown in Figure 4.1. The value in each cell is the probability (relative frequency) of the corresponding pattern in the corresponding context. The light blue color of the cell denotes the lower and dark blue color denotes the higher probability.

According to this table, when students make navigation decision without sequencing (Part 1) or ignore it (Part 2-N), they mostly follow sequential pattern working Within-Topic until they feel that it is sufficient and then moving to Next-Topic. This shows that students tend to attempt most of the activities in the topic before moving to the next one even if it is not the best strategy for their knowledge. The OSSM does hint the students when to move, but its guidance is quite conservative since it is defined by the class as a whole. On the other hand, when students follow GS recommendations, their "groupthink" stay on the current topic shortens considerably, they move to the next topic faster and remarkably expand their non-sequential navigation. This is a good evidence that GS promoted the non-sequential navigation in our study. However, still we cannot conclude whether following the recommendations made by sequencing could benefit learning by directing them to relevant activity more efficiently. We examine this question in the next section.

4.2 The Value of GS: Amount of Learning and Speed

The mere presence of personalized guidance is not sufficient to provide impact, what matters is whether the students choose to follow the guidance or ignore it. We examined the added value of GS by comparing the *amount of learning* and *learning speed* of students who did not follow guidance *(non-followers)* and the ones who did *(followers)*. To this end, we used normalized learning gain and learning speed as our evaluation measures. The normalized learning gain ($nGain$) is defined as the actual gain divided by the possible gain and is obtained using the score of the student in pretest and posttest. The speed of learning is defined as the number of questions student attempted (n_q) to get 1 point increase in the normalized learning gain: ($nGain/n_q$). We multiplied this number by 100 to express it as percentage ($\%speed$). To separate *non-followers* and *followers*, we calculated the *following ratio* per student that represents the fraction of activity accesses made when following recommendations. The ratio considers attempts on questions made in the second part of the study when sequencing was available.

Figure 4.2 shows the distribution of the *following ratio*. As we can see from the skewed distribution, most of the students have the ratio of 0.2 or less, i.e., had followed recommendation in less than $\frac{1}{5}th$ of their attempts. We selected $\frac{1}{5}$ to be the cut-off for separating *non-followers* from *followers*. The *non-followers* group consists of 36 students with following ratio of less than $\frac{1}{5}$ and the *followers* group contained 17 students with the *following ratio* of greater than or equal to $\frac{1}{5}$. There were 8 students in the *non-followers* group and 6 in the *followers* group who either had missing pretest or posttest or seemed not motivated to work seriously on posttest as they got lower score than in pretest. We filtered out those students and we finally had 28 and 11 students left in the *non-followers* and *followers* group, respectively. We found that there were no significant differences between the groups in the normalized learning gain. Speed of learning was higher among the *followers*

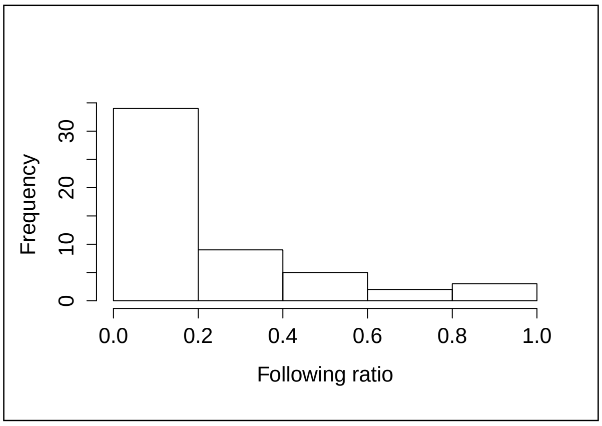

Figure 4.2: Histogram of following ratio of the students participated in the study

($M = 0.97\%, SD = 0.88\%$) than *non-followers* ($M = 0.54\%, SD = 0.27\%$) but only reached borderline significance when compared to the *non-followers* group ($p = .083$ using Welch t-test).

Since learning gain and learning speed might vary across students with different prior knowledge, we also compared *followers* and *non-followers* with low and high prior knowledge separately. If the pretest was less than the median of pretest scores, i.e. 11, a follower/non-follower was labeled as *low pretest*, otherwise *high pretest*. Table 4.2 provides a more detailed summary of these two parameters in *followers* and *non-followers* within the low and high pretest groups. The t-test was used in all of the comparisons since parametric statistics assumptions were held.

We found that the mean of the normalized learning gain is not significantly different across *non-followers* and *followers* with low or high pretest, but the speed is much higher for *followers* reaching significant difference for students with high pretest. This implies that the GS may provide an efficient guidance that leads to a shorter learning path, at least for students with higher pretest score. While this result seems promising, we have to take into account other possible explanations given the design of our study. For example, since students were not randomly assigned to conditions, it could be that the student that followed recommendations were more diligent students, so that their improved performance was due to a selection effect known as 'selection bias'. For this reason, the above analysis needs another evaluation and we hope to address this concern in a future study.

Table 4.2: Mean±SD of evaluations measures for *non-followers* & *followers* separated by pretest group

	Low pretest (n=20)			High pretest (n=19)		
	Non-followers (n=14)	Followers (n=6)	p-value	Non-followers (n=14)	Followers (n=5)	p-value
ngain	0.51±0.28	0.42±0.19	.440	0.48±0.26	0.46±0.29	.870
%speed	0.55%±0.34%	0.97%±0.88%	.128	0.54%±0.27%	1.02%±0.70%	.039 *

Significance level * :< .05

4.3 The impact of GS on System and Class Performance

To see the effect of attempts suggested by sequencing on students' performance, we fitted mixed models to predict the performance of the students in (1) attempts on self-assessment questions in the system (*in-system performance*), and (2) final exam taken at the end of the term (*out-of-system performance*). In all models, a random effect was included to account for unobserved variations between students. The models used the filtered data that had attempts of 53 active students (See Section 4).

To identify the influence of GS on student *in-system performance*, we explored whether the student had a higher chance to answer the question correctly if it was suggested by GS. The variables of interest were (1) *correctness of attempt*, a binary variable showing correct or incorrect answer and (2) *attempt type* showing whether attempt was offered by sequencing or not. We fitted a logistic mixed effects model with *attempt type* as the fixed effect and *correctness of attempt* as the response variable. The data consisted of 5275 attempts on questions that were not offered by GS and 485 attempts on questions that were offered by GS. The results indicated that the *attempt type* was significant predictor of the correctness ($\chi^2(1, 5760) = 14.17, p < .001$). The success was more frequent for questions recommended by GS: the odds of having correct answer when a question was offered by GS was 1.59 times the odds of having correct answer when a question was not offered by GS. This indicates that GS guided student to questions of proper difficulty.

To identify the influence of GS on student *out-of-system performance*, we explored how the work in the system affected the score on the final exam that ranges from 0 to 100. To address this question, we used the filtered data and counted separately total number of attempts on activities recommended by GS and not recommended by GS made by 40 students who had taken the final exam and used the system. We considered mixed models for predicting the score with different set of predictors: (NQ) total number of attempts on questions not recommended, (NQ_{GS}) total number of attempts on questions recommended by GS, (NA) total number of attempts on activities (questions or examples) not recommended, (NA_{GS}) total number of attempts on activities (questions or examples) recommended by GS. Table 4.3 reports summary of the estimated effects for the two fitted models: A and B. An

Table 4.3: Summary of the model fits for predicting students' score in the final exam

Model A		Model B	
	$\beta \pm SE$		$\beta \pm SE$
$Intercept$	68.50±6.24***	Intercept	68.84±5.37***
NQ	0.11±0.06	NA	0.06±0.03*
NQ_{GS}	0.69±0.30*	NA_{GS}	0.56±0.24*

Significance level $^{*} :< .05;^{**} :< .01;^{***} :< .001$

interesting finding was that in Model A, total number of questions accessed by recommendations of GS (NQ_{GS}) was significantly related to the final score. Attempting one question recommended by GS was associated with an increase of 0.69 (or 0.69%) in the final exam score ($SE = 0.30, p = .019$). Model B also showed a significant support for both the total number of attempts on activities that were not recommended (NA) and activities that were recommended by GS (NA_{GS}): attempting one activity recommended by GS was associated with an increase of 0.56 (or 0.56%) in the final grade ($SE = 0.24, p = .017$). At same time, attempting one activity that was not offered was associated with a much lower increase of 0.06 in the final score ($SE = 0.03, p = .045$). In other words, working on both, recommended and not recommended activities positively influenced the score, however, the impact of activities that were recommended by GS was about 9 times greater than the activities that were not recommended.

5

Subjective Evaluation

At the end of the term we applied a questionnaire consisting of 6 questions about the recommendation features in Mastery Grids with answers in a 5-point Likert Scale (1:Strongly Disagree to 5:Strongly Agree). The questions are listed in Table 5.1 and the distribution of the answers can be seen in Figure 5.1(a). Out of 95 students who participated, we kept only the answers of 51 students who used the system at least once.

As the data shows, students seemed to agree that they like to receive recommendations ($Q1 : M = 4.10, SE = 0.11$) and that the use of red stars to represent recommendations was clear ($Q2 : M = 3.86, SE = 0.14$). They also disagreed that recommendations were distracting ($Q5 : M = 2.41, SE = 0.15$). At the same time, it was less clear to them why some contents were recommended ($Q4 : M = 3.82, SE = 0.15$), and they were interested to know the reasons ($Q6 : M = 4.20, SE = 0.11$). When we got a more detailed comparison among *followers* and *non-followers*, we noticed that *followers* ($M = 4.60, SE = 0.131, N = 15$) were even more curious than *non-followers* ($M = 4.07, SE = 0.135, N = 30$) to know why some topics or contents were recommended (see Figure 5.1(b)). This difference was significant using Mann-Whitney test ($U = 133.5, p = .012$).

Furthermore, we found that while the average class opinion was rather neutral on the usefulness of the recommendations ($Q3 : M = 3.06, SE = 0.16$), students with low pretest gave significantly higher score to the usefulness of the sequencing ($Q6 : M = 3.50, SE = 0.24, N = 22$) than high pretest students ($Q6 : M = 2.79, SE = 0.22, N = 24$). This difference was also significant using Mann-Whitney test ($U = 173.5, p = .037$) (see Figure 5.15.1(c)). This is an indication that the GS guidance helped low-score students as well.

Table 5.1: Subjective evaluation questions

#	Question
1	In general, I would like the system to recommend me topics & contents to focus on
2	It was clear for me that red stars were recommendations
3	Recommendations I received this semester in Mastery Grids were useful for me
4	I could not understand why some topics and contents were recommended to me
5	Recommendations distracted me from planning my work
6	It would be useful to see why some topics or contents are recommended to me

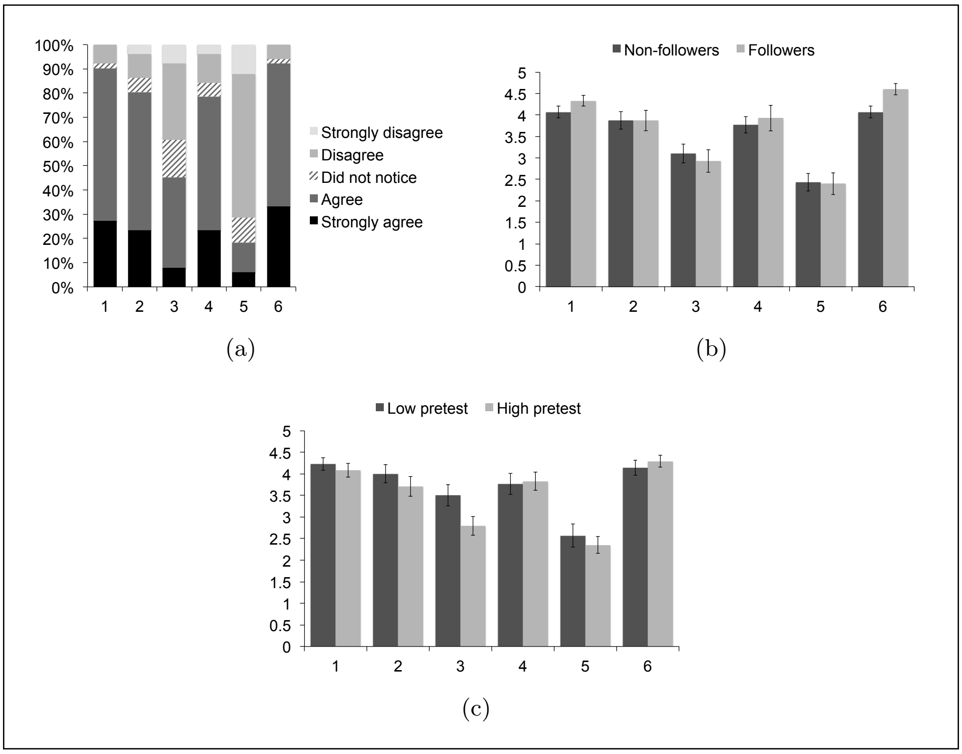

Figure 5.1: (a) Distribution of answers by question, (b) Average score by question for followers and non-followers, (c) Average score by question for pretest groups

6

Discussion and Future Work

This paper investigated the added value of knowledge-based guidance in the context of open social student modeling (OSSM). We presented a greedy sequencing (GS) approach that attempted to maximize student's knowledge and demonstrated how it was implemented in Mastery Grids, an OSSM interface for accessing learning materials. The evaluation of this combined approach delivered several interesting findings.

The proposed approach encouraged non-sequential navigation patterns guiding weaker students to not-mastered materials in previous lectures and advancing stronger to master materials in future lectures. As a result, it increased learning speed of stronger students, leading to more optimal content navigation. In addition, we observed that the amount of work with materials selected by proposed approach was associated with achieving considerably higher score at the final exam. Although this does not mean that the proposed approach induced higher grade in exam, it still shows promising perspectives that could be further explored by future studies.

In the future works, we hope to address limitations of this study. First, it was focused on the domain of Java programming. Although the proposed GS approach can be adapted to other domains, more research is required before the findings of this study could be generalized. Second, the subjects in our study were undergraduates who knew about Java basics beforehand. This could, in fact, explain the reason that relatively few students followed the guidance in our study. We need to plan a future study in an introductory Java course where sequencing assistance will likely be more critical. Finally, the survey report demonstrated that the interface needs to be modified in order to encourage students to follow recommendations. We would also like to increase the transparency of the proposed approach by increasing students' awareness of the reasons to recommend specific learning content.

Bibliography

[1] Brecht, B., McCalla, G., Greer, J., Jones, M.: Planning the content of instruction. In: Proceedings of 4-th International Conference on AI and Education, Amsterdam. pp. 24–26 (1989)

[2] Brusilovsky, P., Sosnovsky, S., Yudelson, M.: Addictive links: The motivational value of adaptive link annotation. New Review of Hypermedia and Multimedia 15(1), 97–118 (2009)

[3] Brusilovsky, P., Sosnovsky, S.A., Yudelson, M., Chavan, G.: Interactive authoring support for adaptive educational systems. In: AIED. pp. 96–103 (2005)

[4] Brusilovsky, P.L.: A framework for intelligent knowledge sequencing and task sequencing. In: Intelligent tutoring systems. pp. 499–506. Springer (1992)

[5] Bull, S., Britland, M.: Group interaction prompted by a simple assessed open learner model that can be optionally released to peers. In: Proceedings of Workshop on Personalisation in E-Learning Environments at Individual and Group Level (PING), User Modeling. vol. 2007 (2007)

[6] Bull, S., Kay, J.: Student models that invite the learner in: The smili:() open learner modelling framework. International Journal of Artificial Intelligence in Education 17(2), 89–120 (2007)

[7] Burke, R.: Hybrid web recommender systems. In: The adaptive web, pp. 377–408. Springer (2007)

[8] Carbonell, J.R.: Ai in cai: An artificial-intelligence approach to computer-assisted instruction. Man-Machine Systems, IEEE Transactions on 11(4), 190–202 (1970)

[9] Hosseini, R., Brusilovsky, P.: Javaparser: A fine-grain concept indexing tool for java problems. In: The First Workshop on AI-supported Education for Computer Science (AIEDCS 2013). pp. 60–63 (2013)

[10] Hsiao, I.H., Sosnovsky, S., Brusilovsky, P.: Guiding students to the right questions: adaptive navigation support in an e-learning system for java programming. Journal of Computer Assisted Learning 26(4), 270–283 (2010)

[11] Hsiao, I.H., Bakalov, F., Brusilovsky, P., König-Ries, B.: Open social student modeling: visualizing student models with parallel introspectiveviews. In: User Modeling, Adaption and Personalization, pp. 171–182. Springer (2011)

[12] Hsiao, I.H., Bakalov, F., Brusilovsky, P., König-Ries, B.: Progressor: social navigation support through open social student modeling. New Review of Hypermedia and Multimedia 19(2), 112–131 (2013)

[13] Loboda, T.D., Guerra, J., Hosseini, R., Brusilovsky, P.: Mastery grids: An open source social educational progress visualization. In: Open Learning and Teaching in Educational Communities, pp. 235–248. Springer (2014)

[14] Vassileva, J., Deters, R.: Dynamic courseware generation on the www. British Journal of Educational Technology 29(1), 5–14 (1998)