Evaluation of a Data Mining Approach to Providing Adaptive Support in an Open-Ended Learning Environment: A Pilot Study

Samad Kardan[1] and Cristina Conati[1]

[1]Department of Computer Science, University of British Columbia
2366 Main Mall, Vancouver, BC, V6T1Z4, Canada

{skardan, conati}@cs.ubc.ca

Abstract. This paper describes the initial evaluation results for providing adaptive support based on effective/detrimental interaction patterns discovered by applying data mining on user interaction data for an Interactive Simulation. Previously, we presented the process of building a classifier user model for the AIspace CSP applet, an open-ended interactive simulation which helps with learning how to solve constraint satisfaction problems. In a later work, we presented a methodology for providing adaptive interventions based on the class association rules that form our classifier user model. In this work, we discuss how to use the generated adaptation rules for delivering adaptive support in the form of hints. The initial qualitative evaluation of the resulting support mechanism, as well as a quantitative evaluation using eye tracking and action logs, show that the interventions were well-received by users.

Keywords: Adaptive Interventions, Interactive Simulations, Eye Tracking

1 Introduction

Interactive Simulations (IS hereafter) are increasingly used as learning tools, where they present an open-ended and exploratory environment to support learning in many different disciplines. These ISs are designed to foster exploratory learning by giving students the opportunity to practically and proactively experiment with concrete examples of concepts and processes they have learned theoretically. However, it has been shown that if the students are left to experiment and explore without any additional support, many will show suboptimal interaction behaviors (e.g., [1]) and may not learn well from this form of interaction (e.g., [2]). These students can benefit from having additional support in the form of scaffolding while interacting with this type of Open-Ended Learning Environments (OELEs) (e.g., [3]). The Constraint Satisfaction Problem (CSP) Applet is one of the collection of interactive tools for learning common Artificial Intelligence algorithms, called AIspace [4]. The CSP applet is an Interactive Simulation designed to help students deepen their understanding of solving constraint satisfaction problems. We intend to add adaptive support to the CSP applet to help students use the applet effectively for learning. Implementing adaptive interventions requires adding two components to an OELE: (1) a **user model** that determines if and when to intervene, with additional information on which interventions are appropriate at the time; and (2) an **intervention mechanism** that delivers different interventions based on the assessment of the student model.

Due to the open-ended nature of the interactions with ISs, providing intelligent support is challenging because many different possible behaviors should be taken into account and most often it is not known a priori which behaviors are effective and which ones are not. All this makes developing a successful intelligent support mechanism time consuming [5]. To address these challenges in a timely and generalizable manner, we employ Educational Data Mining [6] methodologies. Our goal is to find relevant patterns in user interaction data in an IS (e.g. the CSP applet) that leads to different levels of user performance. Then, build a user model based on these patterns and finally, use these patterns to extract adaptation rules for delivering relevant adaptive interventions.

To achieve this goal, first we developed a user modeling framework that utilizes user clustering and class association rules mining to identify relevant user types/behaviors from interface actions [7]. Then, we devised a methodology for using the discovered association rules to generate adaptation rules which are then transformed to adaptive interventions [8]. This paper describes the initial evaluation of adaptive interventions that are implemented following our proposed process.

The rest of the paper is organized as follows: First, we briefly describe the CSP applet, the user modeling framework used for extracting user behaviors (i.e., the class association rules), and the methodology for generating adaptation rules based on these behaviors. Then, we discuss the different dimensions for providing interventions based on these adaptation rules. Finally, we present the results of a pilot study with a new version of the CSP applet that implements the proposed support mechanism.

2 The AIspace CSP applet

A CSP consists of a set of variables, variable domains and a set of constraints on legal variable-value assignments. Solving a CSP requires finding an assignment that satisfies all constraints. The CSP applet illustrates the Arc Consistency 3 (AC-3) algorithm for solving CSPs represented as networks of variable nodes and constraint arcs. AC-3 iteratively makes individual arcs consistent by removing variable domain values inconsistent with a given constraint, until all arcs have been considered and the network is consistent. Then, if there remains a variable with more than one domain value, a procedure called domain splitting is applied to that variable in order to split the CSP into disjoint cases so that AC-3 can recursively solve each case.

The CSP applet demonstrates the AC-3 algorithm dynamics via interactive visualizations on graphs using color and highlighting, and graphical state changes are reinforced through textual messages. The applet provides several mechanisms for the interactive execution of the AC-3 algorithm on a set of available CSPs. These mechanisms are accessible through the toolbar, or through direct manipulation of graph elements. The user can perform seven different actions: (1) Fine Step: use the fine step button to see how AC-3 goes through its three basic steps (selecting an arc, testing it for consistency, removing domain values to make the arc consistent); (2) Direct Arc Click: directly click on an arc to apply all these steps at once; (3) Auto AC:

automatically fine step on all arcs one by one using the auto arc consistency button; (4) Stop: pause auto arc consistency; (5) Domain Split: select a variable to split on, and specify a subset of its values for further application of AC-3 (see pop-up box in the bottom right of Fig. 1); (6) Backtrack: recover alternative sub-networks during domain splitting; (7) Reset: return the graph to its initial status.

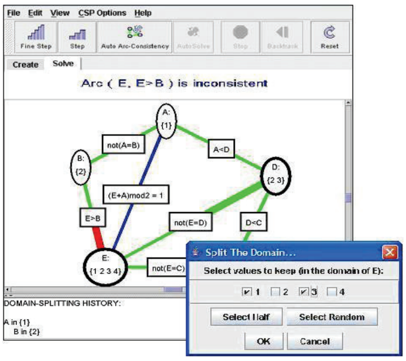

Fig. 1. CSP applet with example CSP problem

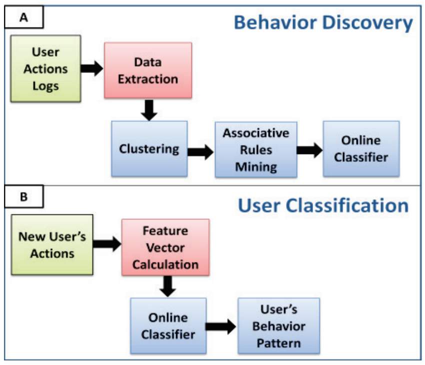

Fig. 2. General User Modeling Approach.

3 Mining Behavior Patterns

In this section we briefly describe the two main phases of our approach to building a classifier user model from interaction data first described in [7]: Behavior Discovery (Fig. 2A) and User Classification (Fig. 2B). In *Behavior Discovery*, raw unlabeled data from interaction logs is preprocessed into feature vectors representing individual users in terms of their interface actions. These vectors are the input to an unsupervised clustering algorithm (i.e., k-means with a modified initialization step, see [7]) that groups them according to their similarities. The resulting clusters represent users who interact similarly with the interface. These clusters are then analyzed to identify if/how they relate to learning. Afterwards, association rule mining is applied on each cluster to extract the common behavior patterns in the form of class association rules for each performance level. A Class Association rule is a rule in the form of X→ *c*, where X is a set of feature-value pairs and *c* is the predicted class label (i.e., the cluster) for the data points where X applies (see Table 1).

Our goal is to use these detected behaviors and information regarding their effectiveness as a guide for intelligent adaptive support during the interaction. Thus, in the *User Classification* phase (Fig. 2B), class association rules extracted in the Behavior Discovery phase are used to build an online classifier user model. This classifier is used to assess the performance of a new user based on her interactions.

In [7], we reported the result of applying our framework on the action logs collected from a study with 65 users using the CSP applet. For this dataset, the Behavior Discovery resulted in two clusters of users that achieved significantly different learning performance levels (high vs. low). We will refer to them as High Learning Gain (HLG) and Low Learning Gain (LLG) groups respectively. Also, the online classifier achieved an accuracy of over 80% in classifying new users as HLG or LLG by observing only the first 25 percent of their interactions.

In addition to assigning a label to the user, the user model also returns the observed rules that caused that classification decision. In [8], we described our proposed methodology for building an intervention mechanism based on the discovered behavior patterns which is briefly described in the next section.

4 Extracting Adaptation rules from Discovered Patterns

The class association rules generated in the Behavior Discovery phase represent the interaction behaviors of LLG and HLG. All of these rules are used in the classifier user model to determine the performance of a new user, and identify a set of behaviors that are either conducive or detrimental to learning. Ideally, one would want to design adaptive interventions that discourage all the detrimental behaviors, and encourage all the good ones. For instance, consider the following rule for the LLG:

Rule4: If Direct Arc Click frequency = Lowest **and** Direct Arc Click Pause Average = Lowest → Cluster LLG

This rule indicates that if the frequency of Direct Arc Click (DAC) action is lower than a threshold (the mechanism to set this threshold is described in [7]) and the average pause time between a DAC and the next action is also lower than a certain threshold then the user is considered a LLG. Here, we want to prevent this from happening and there are two possible interventions (***intervention items*** from now on) that can be delivered to address this rule: (1) Encouraging/enforcing the user to perform DAC more often; (2) Encouraging/enforcing the user to pause longer after DAC actions (possibly thinking about the DAC outcomes).

There may be several rules like the one above that are applicable at a given time. The number of rules, may pose a challenge considering factors such as the cost of implementation and effectiveness of the resulting intervention items, thus filtering the rules is necessary (see [8] for a detailed discussion). For each intervention item, we compute a score calculated as the sum of the weights of the rules which recommend that item within a given cluster (these weights indicate the importance of each rule in classifying a user [7]) and use this as an importance factor for that item. Then we apply a filtering strategy that keeps the most prominent behaviors and ignores the weaker ones while taking the diversity of the intervention items and their cost of implementation into account (see [8] for details). For our current study, we use 6 intervention items as selected by our filtering strategy, highlighted in Table 1.

Table 1. A selection of representative rules for HLG and LLG clusters in the CSP dataset

Rules for HLG cluster:
Rule1: Direct Arc Click frequency = Highest
Rule5: Domain Split frequency = Highest **and** Auto AC frequency = Lowest
└ **Rule8:** Domain Split frequency = Highest **and** Auto AC frequency = Lowest **and** Fine Step Pause Average = Highest **and** Reset frequency = Lowest
Rules for LLG cluster:
Rule1: Direct Arc Click Pause Average = Lowest
Rule3: Direct Arc Click frequency = Lowest

When delivering the implemented interventions to a user, there can be more than one rule satisfied at a certain time leading to multiple items being recommended to that user. If the items are semantically correlated (as determined by the system designer), there is an opportunity to combine two items into one hint. For instance, based on the light blue items in Table 1, a hint can recommend using Direct Arc Click instead of Auto AC, because Direct Arc Click is a finer-grained version of Auto AC, with added user involvement (semantically correlated items have the same color in Table 1). However, non-related items will need separate hint messages and we decided to deliver only one hint at a time to prevent users from possibly getting overwhelmed. Therefore, in each step we choose the intervention item with highest score, calculated similar to above but only for the satisfied rules that recommend that item.

Adaptation rules can be categorized into two main groups, (1) Preventive interventions that discourage bad behavior as detected by the rules for LLG cluster, e.g.: "IF user is classified as a LLG and is using Direct Arc Click very infrequently (less than a threshold), then give a hint to promote this action"; and (2) Prescriptive interventions that encourage the effective behaviors described by the rules for HLG cluster. In this case, we want these rules to be satisfied. This means that if a student labeled as LLG shows any behavior in contrast with these rules then the corresponding intervention will be delivered to her, e.g.: "IF the user label is LLG, then if *Direct Arc Click frequency* is lower than x and *Auto AC frequency* is higher than y then "prompt user to use Direct Arc Click instead of Auto AC".

The advantage of preventive interventions is that we already know these behaviors result in bad performance so we can confidently prevent users from following such patterns. Prescriptive interventions are less reliable because it is not clear if/how behaviors that were effective for some learners could be beneficial for others.

5 Designing adaptive interventions

There are different forms of adaptive interventions that can be used to implement a specific adaptation goal (in our case, helping students use and learn most effectively from the CSP applet). Similar to most of the educational environments that provide adaptive support, we provide explicit advice via textual hints, and provide this advice incrementally. However, our focus on the interface actions when extracting the user interaction behaviors enables us to make interface changes as another way of delivering interventions. Thus, we provide a first level of advice with a textual hint that suggests or discourages a target behavior, followed when needed by a textual hint that reiterates the same advice, accompanied by a related interface adaptation (e.g., highlighting or deactivating relevant interface items).

Delivering adaptive interventions also require deciding whether the interventions should be subtle or forceful. Subtle interventions are in the form of suggestions that can be easily ignored by the user (e.g. a text message shown in a hint box at the corner of the screen). Forceful interventions make the user follow the related advice by reducing or eliminating user's options for the next action (e.g. deactivating all the items on the toolbar to force the user to pause before taking next action).

The current adaptive version of the CSP applet uses the subtle approach. The main drawback of this approach is that the recommendations may not be attended to by users or the user might decide not to follow them. However, this approach has the very desirable advantage of being less intrusive than the forceful approach. Therefore, from a usability point of view, it makes sense to try and see whether subtle adaptive interventions can already significantly improve the effectiveness of the CSP applet.

The detailed procedure of delivering the subtle incremental interventions described above is as follows: (1) for each intervention there is a text message presented in format of a hint that appears in a hint box at the upper left corner of the applet's main panel (level-1 hint). The hint box will blink once, each time a new message is displayed. (2) After receiving the hint, the student is given a time window to change her behavior. (3) If after the time window, the preconditions for that intervention are still satisfied the intervention will be provided again. In this case in addition to a text message, corresponding interface element(s) for that intervention will be highlighted until the user chooses her next action (level-2 hint). Figure 3 shows a level-2 intervention suggesting a decrease in use of *Auto AC* vs. an increase in use of *Direct Arc Click*. In addition to a text message the arcs that can be clicked are also highlighted.

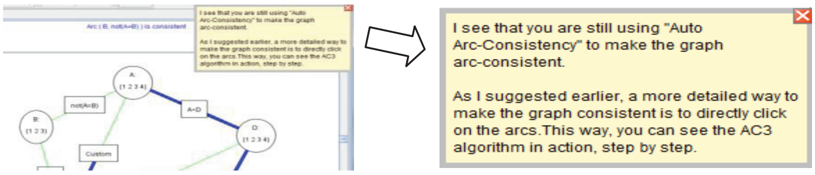

Fig. 3. A hint suggesting the use of Direct Arc Click action with the interface highlights (left); and the content of the hint box (right).

6 Evaluation

We ran a pilot study in a Wizard-of-Oz setting (i.e., experimenter would trigger the interventions based on a set adaptation rules) to evaluate the intervention mechanism described above for three factors: visibility, intrusiveness, and follow rate of the interventions. The data was collected from 6 computer science students. Each participant: (1) studied a textbook chapter on the AC-3 algorithm; (2) wrote a pre-test on the concepts covered in the chapter; (3) used the CSP applet to study two CSPs, while her gaze was tracked with a Tobii T120 eye-tracker; and (4) took a post-test analogous to the pre-test [9]. At the end of the experiment, a qualitative evaluation of interventions was done using a post-hoc questionnaire and a follow-up interview.

Figure 4 summarizes the opinion of our 6 participants about the text hint messages collected by the post-hoc questionnaire. The participants did not find the hint messages intrusive or annoying. They found the messages easy to notice and useful in the process of interaction. Moreover, most of the participants reported following the instructions provided in the hints. The rest of this section will present quantitative results derived from action logs and eye gaze data collected during the interaction.

Regarding visibility of the hints, out of 27 hints provided in total, 25 of them were attended to by the participants. One of two omitted hints was a level-1 hint given to participant 4 (P4), while she did not notice this hint, the subsequent level-2 of the same hint (with interface highlights) managed to grab her attention. The second case was a level-2 hint given to P6, where he decided not to follow a level-1 hint prior to this hint and was given a level 2 hint. In this case, the highlighting reminded him of the recommended action (Direct Arc Click) from the level-1 hint, thus he followed the hint without having to look at the hint box. These two cases, highlight the importance of the 2-level hinting strategy reinforced by interface changes.

Figure 5 illustrates the number of hints shown, attended to and followed by each participant. Out of 27 hints given, 20 were followed by the participants (74% follow rate). Students, who show many detrimental behaviors, will get more hints. Such students are the target group that we want to help learn better from their interaction with the CSP applet. Therefore, P2 and P4 are of especial interest. Both of these participants reported finding the interventions relevant and useful. However, P4 did not follow every hint, and generally only followed the recommendations when repeated in the form of a level-2 hint. This is reflected in her self assessment of how often she followed the hints as well (Table 2).

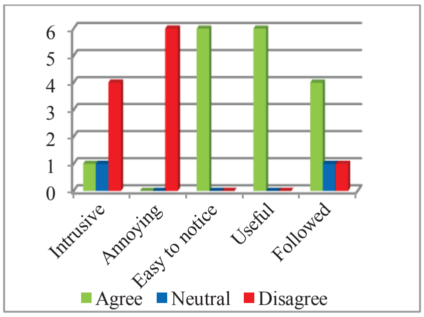

Fig. 4. Reception of the text hints by participants

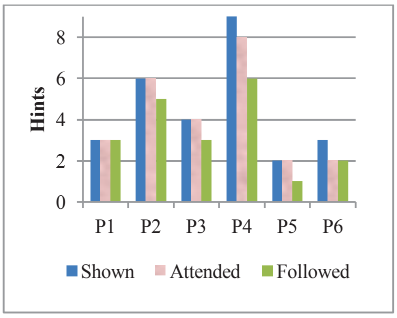

Fig. 5. Number of hints shown, attended and followed for each participant

We also analyzed the average reading time of the hint messages for each participant, overall and for the hints they dismissed/followed (Table 2). We can observe an individual element in reading time between participants which can be further investigated as a guide for user adaptive reaction time for hints. Another trend is that users who received more hints also spent less time reading them. This is expected as these users are the ones with sub-optimal interaction behaviours and this again shows the importance of the 2-level progressive hinting strategy which gets more intrusive the second time a hint is provided.

Table 2. Hint rate, self rated following of hints, and average reading time for each participant

	P1	P2	P3	P4	P5	P6
Followed Hints - Self-rated (1-5)	4	4	4	2	4	3
Avg. Reading Time (ms)	2814	1642	1547	925	2639.5	9460
Avg. Reading Time: Followed (ms)	2814	1530.6	1663	937.5	3464	8975
Avg. Reading Time: Dismissed (ms)	-	2199	1199	887.5	1815	9945
# Hints given	3	6	4	9	2	3

7 Conclusion and future work

In this paper, we presented the final step of the process for adding adaptive interventions to an OELE called AIspace CSP applet. This process started with mining behavior patterns in the form of association rules from a dataset of collected user interface actions [7]. Then, continued with extracting adaptation rules from the discovered behaviors [8]. The final step was to deliver the adaptive interventions defined based on the adaptation rules via an intervention mechanism. We identified the *form* and *forcefulness* of delivering the interventions as two aspects of this step and described our 2-level subtle method of delivering interventions using both text messages and interface changes. The very encouraging initial results of our pilot study regarding reception of the interventions by the users, shows a great potential for the Adaptive version of the CSP applet which provides personalized support. A second pilot study is scheduled to test the user model and the improvements made to the applet based on our findings in the first pilot study. We plan to run a full scale study afterwards.

References

1. Ploetzner, R., Lippitsch, S., Galmbacher, M., Heuer, D., Scherrer, S.: Students' difficulties in learning from dynamic visualisations and how they may be overcome. Computers in Human Behavior. 25, 56–65 (2009).
2. Shute, V.J.: A comparison of learning environments: All that glitters. Computers as cognitive tools. pp. 47–73. Hillsdale, NJ, England: Lawrence Erlbaum Associates, Inc (1993).
3. De Jong, T.: Technological Advances in Inquiry Learning. Science. 312, 532–533 (2006).
4. Amershi, S., Carenini, G., Conati, C., Mackworth, A.K., Poole, D.: Pedagogy and usability in interactive algorithm visualizations: Designing and evaluating CIspace. Interacting with Computers. 20, 64–96 (2008).
5. Cocea, M., Gutierrez-Santos, S., Magoulas, G.D.: Challenges for intelligent support in exploratory learning: the case of ShapeBuilder. Proceedings of the International Workshop on Intelligent Support for Exploratory Environments at ECTEL 2008. , Maastricht, The Netherlands (2008).
6. Romero, C., Ventura, S.: Educational Data Mining: A Review of the State of the Art. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews. 40, 601–618 (2010).
7. Kardan, S., Conati, C.: A Framework for Capturing Distinguishing User Interaction Behaviours in Novel Interfaces. In: Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero, C., and Stamper, J. (eds.) Proceedings of the 4th International Conference on Educational Data Mining. pp. 159–168. , Eindhoven, the Netherlands (2011).
8. Kardan, S., Conati, C.: Providing Adaptive Support in an Exploratory Learning Environment by Mining User Interaction Data. Proceedings of the 5th International Workshop on Intelligent Support for Exploratory Environments (ISEE 2012), in conjunction with the 11th International Conference on Intelligent Tutoring Systems (ITS 2012). , Chania - Greece (2012).
9. Kardan, S., Conati, C.: Exploring Gaze Data for Determining User Learning with an Interactive Simulation. In: Masthoff, J., Mobasher, B., Desmarais, M., and Nkambou, R. (eds.) User Modeling, Adaptation, and Personalization. pp. 126–138. Springer Berlin / Heidelberg (2012).