Stimulating collaborative activity in online social learning environments with Markov decision processes

Matthew Yee-King
Computing, Goldsmiths
University of London
m.yee-king@gold.ac.uk

Mark d'Inverno
Computing, Goldsmiths
University of London
dinverno@gold.ac.uk

ABSTRACT

Our work is motivated by a belief that social learning, where a community of students interact with eachother to co-create and share knowledge, is key to our students developing 21st century skills. However, convincing students to engage in and value this kind of activity is challenging. In this paper, we employ a technique from AI research called a Markov Decision Process (MDP) to model social learning activity then to suggest interventions that might increase the activity. We describe the model and its validation in simulation and draw conclusions about the effectiveness of this approach in general. The main contributions of the paper is to (i) show how it is possible to model education data as an MDP (ii) show that the resulting decision policy succeeds in guiding the system towards goal states in simulation.

Keywords

Social learning; Education system modelling, MDP, MOOC

Categories and Subject Descriptors

K.3.1 [**Collaborative learning**]: K.3.2 Computer science education G.3 Markov processes

1. INTRODUCTION

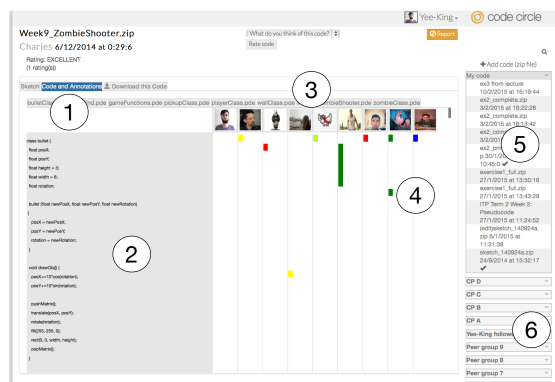
In this paper, we use a Markov Decision Process (MDP) to model social learning activity in terms of content consumption and content creation. This allows us to derive an 'action policy' which can potentially inform tutors and students what type of content to create and when to create it in order to maximise the levels of consumption of content in a social learning system. MDPs [2] are a commonly used method for sequential decision making under uncertainty, and they have been used in education technology e.g. [1]. The work presented here represents a novel application of MDP in a social learning context[1].

[1] A full version of the paper can be found at http://dx.doi.org/10.13140/RG.2.1.3592.0242

1.1 The case study and data set

The data used for the analysis presented here was collected during a 10 week case study involving 174 students on an introductory undergraduate programming course who were learning how to program using the Processing IDE. The students were using our social learning environment [3], as shown in Figure 1, which allow in-browser execution of programs as well as sharing, commenting and replying to comments on specific sections of code.

Figure 1: **The code discussion UI. 1) mode buttons: view running program, view code, download code, 2) the code viewer 3) the people who have commented on this code 4) a comment about a section of the code 5) my uploaded content 6) my communities.**



2. THE MODEL

MDP problems are formulated in terms of states, actions, state transitions, reward functions and action policies. The action policy dictates what is the best action to take in a given state in order to maximise future reward, where reward is defined in terms of the value of each state.

We begin by slicing the dataset into time windows and counting the number of activity types per window, split into content consumption and content creation activities. We define state as a 5 dimensional vector describing levels of 5 types of content consumption activity, namely read code, login, open thread, preview comment (pre-comm) and run code. The size of the state space is reduced by converting the raw

Table 1: **An excerpt from the action policy, showing its proposed content creation actions for the most commonly observed content consumption states**

read code	login	open thread	pre-comm	run code		comment	reply	share	grade-comm	grade code
0	0	0	0	0	→	0	0	0	0	1
1	0	0	0	0	→	0	0	0	0	1
2	2	2	2	2	→	2	0	0	0	0
1	0	0	1	1	→	2	0	0	0	0
2	1	2	2	2	→	0	0	2	0	0
State						Action				

log counts (e.g. number of times 'read code' happened) into 3 bands indicating low (0), medium (1) or high (2) activity relative to other time slices. For example, a state of 01012 would indicate low read code (0), medium login (1), low open thread (0), medium preview comment (1) and high run code (2). Some states are shown in Table 1. We define action as a 5 dimensional vector describing the levels of 5 content creation activities, namely comment, reply, share, grade comment (grade-comm) and grade code. As for state, they are reduced to low, medium or high relative to other time slices. For example, an action of 01012 would mean low comments, medium replies, low shares, medium comment grading, and high code grading (where we refer to the amount of grading, not the grades themselves). We can then gather observations of state-action-state tuples in the dataset, i.e. we observed this action being taken in this state and it was followed by this state. This is converted to a state transition matrix, an example entry in which is: "20020x00000" : "02002" : 0.5,"02222" : 0.5. In this example, state 20020 and action 00000 are observed to be followed by states 02002 and 02222 in an equal number of cases.

The next part of the MDP formulation is the reward function which involves assigning a value to every possible state plus a reward and cost for every possible state-action pair. State values are essentially sums of the elements of the state vector (state 00120 is worth 3). The values of state-action pairs are a sum of the values of all states observed to follow that state-action pair weighted by the number of observations of each follow on state. The cost for an action is calculated based on the frequency of that action in the observed set of actions, where we assume that infrequent actions are costly.

3. SIMULATION

Having derived an action policy, we will now evaluate it by runing it in simulation against a state transition matrix derived from a different period of the case study than the period the action policy was trained on. In this case, the training and test sets contained data derived from the same student cohort, just gathered during different time periods. The aim of this simulation is to examine the ability of the policy to generalise (to the same students in a different time window), and therefore to assess the potential usefulness of this system in a real world context.

Figure 2: **Test performance of the action policy in simulation vs. real world performance with varying time slice length. With 1 hour time slices, the MDP provides 1.25 times more value. Error bars are based on standard deviation over 100 simulation runs.**

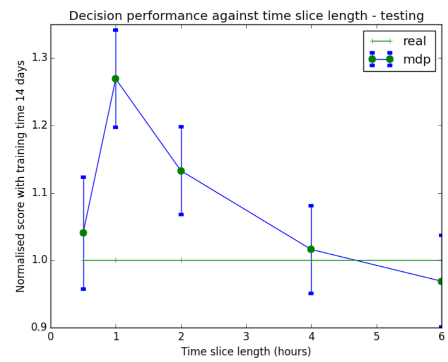


Figure 2 shows the results of running the action policy in simulation, where the training was carried out with varying time slice lengths. For each time slice length, the simulation was run 100 times to establish the typical range of performance. The one hour time slice provides the best performance, where the accumulated state values over the simulation were 1.25 times the value accumulated in the real case study data. It should be noted that the deteriorating performance as time slice length increases is likely to be caused by the smaller number of samples: there are less 6 hour slices than there are 1 hour slices. This means the transition matrix becomes very sparse, resulting in very limited simulation detail. This positive result demonstrates that the MDP approach could be a viable method to model and advise about online educational systems based around content consumption and creation.

4. CONCLUSION

We have described how social learning activity data can be formulated into an MDP and that this formulation allows the derivation of an action policy that can be used to decide what kind of content to create and when to create it, in order to maximise content consumption activity. We have also presented a preliminary validation of the action policy in a simulation based on real data, showing that the action policy selects actions that lead to higher levels of content consumption.

5. REFERENCES

[1] G. Durand, F. Laplante, and R. Kop. A Learning Design Recommendation System Based on Markov Decision Processes. In *17th ACM SIGKDD conference on knowledge discovery and data mining*, 2011.

[2] T. Snijders. Methods for longitudinal social network data: Review and Markov process models, 1995.

[3] M. Yee-King, M. Krivenski, H. Brenton, and M. D'Inverno. Designing educational social machines for effective feedback. In *8th International Conference on e-learning*, Lisbon, 2014. IADIS.