Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society Conference

Exploring the Effects of Errors in Assessment and Time Requirements of Learning Objects in a Peer-Based Intelligent Tutoring System

John Champaign and Robin Cohen
University of Waterloo
200 University Avenue West
Waterloo, Ontario, Canada N2L 3G1
jchampai@cs.uwaterloo.ca,rcohen@ai.uwaterloo.ca

Abstract

We revisit a framework for designing peer-based intelligent tutoring systems motivated by McCalla's ecological approach, where learning is facilitated by the previous experiences of peers with a corpus of learning objects. Prior research demonstrated the value of a proposed algorithm for modeling student learning and for selecting the most beneficial learning objects to present to new students. In this paper, we first adjust the validation of this approach to demonstrate its ability to cope with errors in assessing the learning of student peers. We then deepen the representation of learning objects to reflect the expected time to completion and demonstrate how this may lead to more effective selection of learning objects for students, and thus more effective learning. As part of our exploration of these new adjustments, we offer insights into how the size of learning object repositories may affect student learning, suggesting future extensions for the model and its validation.

Introduction

In this paper, we provide a model for determining which learning objects to present to students in an intelligent tutoring system[1], against the backdrop of a repository of learning objects and a history of the previous experiences of peers with these objects. This approach uses techniques inspired by collaborative filtering (Breese, Heckerman, and Kadie 1998), identifying which users in a system are similar to each other, to then preferentially recommend what has been most useful to similar students. Our work is also motivated by McCalla's ecological approach to e-learning systems (McCalla 2004) described as "attaching models of learners to the learning objects they interact with, and then mining these models for patterns that are useful for various purposes". This approach demands that, in real-time, the learning of the students adjusts and informs the process of selecting the appropriate content for each new student.

In previous work (Champaign and Cohen 2010) two primary elements were in focus: i) specifying the algorithm that determines which peers and which learning objects are most important for each new student, based on a modeling of similarity matching and of the benefit derived from learning objects by previous peers ii) presenting a validation of the approach that simulates student learning, leveraging an assessment in terms of letter grades (through pre- and post-tests) as well as a modeling of the target knowledge levels for each learning object.

In this paper, we first of all demonstrate the robustness of the algorithm (item i) above) by introducing error into the assessment used as part of the validation. Through this extension, we are able to show that the average knowledge level attained by the students continues to reflect appropriate learning, because the ongoing collaborative recommendation of learning objects helps to compensate for errors that are introduced.

We also explore a richer modeling of learning objects in terms of their expected time requirements. From here, we return to populate our simulations with learning objects of varying temporal demands, continuing to operate with possible errors in assessment as well. We also extend the size of the student population to be much larger, in our experiments. We are able to show that our revised algorithm continues to provide high levels of knowledge to students, on average. We conclude with a detailed discussion of the value of this extended model, in comparison with related work, leading to some proposed directions for future research.

Background

The previous algorithm for determining which learning objects to assign to students (Champaign and Cohen 2010) is presented in Algorithm 1. It assumes that we are tracking a set of values, *v[j,l]*, representing the benefit of the interaction for user *j* with learning object *l*. *v[j,l]* is determined by assessing the student before and after the interaction, and the difference in knowledge is the benefit. For each learning object, the previous interactions of students with that object (in terms of their initial and final assessments) is also recorded.[2] A student's knowledge is assessed by mapping it to 18 concrete levels: A+, A, A-, ... F+, F, F-, each representing $\frac{1}{18}$th

[1] A learning object might be a lesson on fractions or a video showing a sorting algorithm, for example.

[2] The algorithm would be run after an initial phase where students are learning through the use of a set of learning objects. These students' experiences would then form the basis for instructing the subsequent students.

Algorithm 1 Collaborative Learning Algorithm

```
Input the current-student-assessment (CSA)
for each learning object (LO): do
    Initialize currentBenefit to zero
    Initialize sumOfBenefits to zero
    Input previous interactions between students and LO
    for each previous interaction on LO: do
        similarity = calculateSimilarity(CSA, interaction-initial-assessment (IIA))
        benefit = calculateBenefit(IIA, interaction-final-assessment)
        sumOfBenefits = sumOfBenefits + similarity * benefit
    end for
    currentBenefit = sumOfBenefits / numberOfPreviousInteraction
    if bestObject.benefit < currentBenefit then
        bestObject = currentObject
    end if
end for
if bestObject.benefit < 0 then
    bestObject = randomObject
end if
```

of the range of knowledge.

The anticipated benefit of a specific learning object *l*, for the active user, *a*, under consideration would be: [3]

$$p[a,l] = \kappa \sum_{j=1}^{n} w(a,j)v(j,l) \quad (1)$$

w(a,j) reflects the similarity $\in$ (0,1] between each user *j* and the active user, *a*, and κ is a normalizing factor. $\frac{1}{|n|}$ was used as the value for κ in this work where *n* is the number of previous users who have interacted with learning object *l*. *w(a,j)* was set as $\frac{1}{1+difference}$ where difference is calculated by comparing the initial assessment of j and the current-student-assessment of the active student *a*, and assigning an absolute value on the difference of the letter grades assigned. This permits us to obtain a similarity value between 0 and 1, with 1 representing identical assessments and is shown as the calculateSimilarity function in Algorithm 1. *v(j,l)* is also computed using a difference, not an absolute difference but an actual difference (between the initial and final assessments). For example, *v(j,l)* where *j* is initially assessed as A+ and finally assessed at B- would be -5. This is shown as the calculateBenefit function in Algorithm 1. The user *a* is ultimately assigned the learning object *l* that maximizes *p[a,l]*.

In order to simulate the learning achieved by students, the following approach was used. Let *LOK[l,k]* represent some learning object *l*'s target instruction level of knowledge *k*, such that *LOK[l,k]* $\in$ [0,1]. For example, the target instruction level might be 0.68 for a 90 minute lab on recursion, since students have completed previous learning but are still gaining an understanding of nuances.

[3]Adapted from (Breese, Heckerman, and Kadie 1998)

Learning objects also have an impact, which can be positive or negative[4]. Let *I[l,k]* $\in \mathcal{R}$, represent the impact of learning from learning object *l* on the knowledge *k*, that is, in the optimal case how much the learning object can adjust a student's knowledge *k*. The impact can be thought of as, for a student at the target level, what is the expected learning benefit of the object. This is information used by our approach to simulate the learning that is occurring.

Let *UK[j,k]* represent user *j*'s knowledge of $k \in$ K, such that *UK[j,k]* $\in$ [0,1]. An example from computer science would be a knowledge of recursion recorded to be at 0.33. This would be interpreted as the student has an understanding of 33% of the course content dealing with recursion.

After an interaction with an object *l*, a user *j*'s knowledge of *k* is changed by:

$$\Delta UK[j,k] = \frac{I[l,k]}{1 + (UK[j,k] - LOK[l,k])^2} \quad (2)$$

This has the implication that the impact of a lesson is at a maximum when the student's knowledge level matches the target level of the learning object. As the two values differ, the impact of the lesson exponentially decreases.

Based on this change, the user's knowledge in that area is updated as:

$$UK'[j,k] = UK[j,k] + \Delta UK[j,k] \quad (3)$$

The user's average knowledge can then be calculated as:

$$\overline{UK}[j] = \frac{1}{|K|} \sum_{k \in K} UK[j,k] \quad (4)$$

In order to plot learning curves, the average knowledge ($\in [0,1]$) of all students is plotted against their progress in the course of study. Algorithms perform well when the average knowledge attained by students is high. Previously, a set of algorithms to select learning objects for students were run, to demonstrate the value of the proposed approach. **Random Association** associates each student with a randomly assigned learning object; **Greedy God** chooses the best possible interaction for each student for each trial. There two curves are the benchmarks (low performance and "the ideal"). Three variations of Algorithm 1 were then run. **Raw Ecological** has each student matched with the learning object best predicted to benefit her knowledge; **Pilot Group** has a subset of the students (10%) assigned, as a pilot group, systematically to learning objects - these interactions are used to reason about the best sequence for the remaining 90% of the students; **Simulated Annealing** is such that during the first 1/2 of the trials there is an inverse chance, based on the progress of the trails, that each student would be randomly associated with a lesson; otherwise, the ecological approach was applied. Encouraging results for all three variations of Algorithm 1 were found.

[4]The negative impact was introduced to simulate the possibility of misinformation from a poor quality learning object or a learning object that does a good job teaching one concept, while undermining the understanding of another concept.

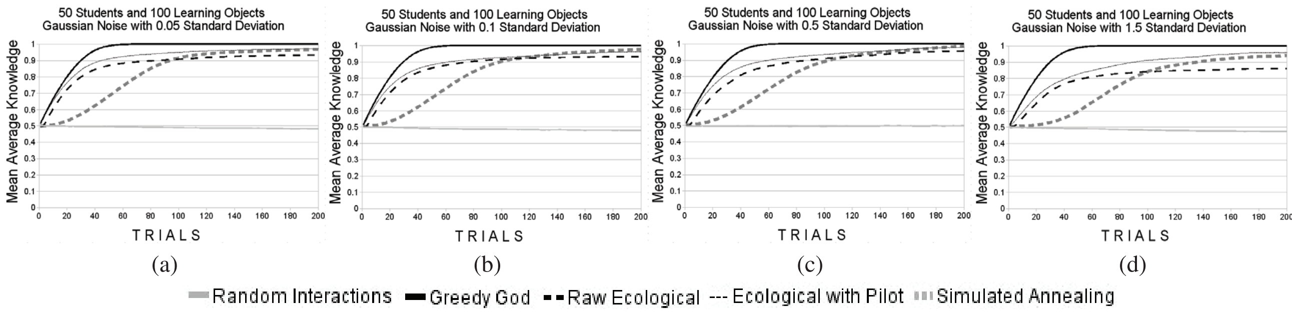

Figure 1: Comparison for Varying Standard Deviation of Error in Assessment

Error

Algorithm 1 proposes the selection of learning objects for students based on their similarity to peers and the benefits these peers have obtained from existing learning objects. Both similarity and benefit are determined in terms of the assessment levels of the students (obtained by some kind of pre- and post-test of each student, mapped to a level of a letter grade).

Since assessment in the real world is both imprecise and occasionally inaccurate[5], there is merit in exploring how well the algorithm would perform when validated in a simulated environment where the assessments include an element of error. Our interest is in how well the algorithm makes recommendations using this noisy data[6].

ERROR: approach

Our original hypothesis was that we expected errors in assessment, of the form of introduced noise, to degrade the learning curves observed. That is, we expected that as the noise increased, the slope of each learning curve embodying Algorithm 1 would decrease and they would gradually move away from the ideal greedy god curve and towards the random baseline. It was expected that rather than converging on perfect knowledge (the 1 value on the y axis), the curves would converge on a lower value (that would drop ever lower as the noise increased). This would happen as the number of interactions between students and learning objects increased (modeled as trials in the x axis).

If the algorithm were robust in the face of error (that is, if the learning curves stay closer to the ideal case), this tells us that our approach can handle errors in assessment and continue to provide worthwhile recommendations to students, even in the face of assessment errors. Poor performance, where the slope of the curves would drop quickly towards the random baseline, would tell us that this approach is highly dependant on good assessments (and would thus constrain the environment where it would be appropriate to use this approach).

In order to produce the error, we modified that assessment function in our simulation. Rather than mapping a knowledge level (continuous values in the range [0,1]) to a discrete level {A+, A, ... F, F-}, we first added a random number, using a Gaussian distribution, with a mean of zero and a standard deviation of 0.05, 0.1, and 0.5[7]. These experiments, taken as a whole, should provide us with an understanding of how increasing levels of noise in the assessment affects the effectiveness of curriculum sequencing performed by this approach. The greedy god and random baselines remained unchanged, since neither relied on assessment.

ERROR: results and discussion

We did not see evidence of what we originally expected with the 3 graphs created with standard deviation (0.05, 0.1, 0.5). Instead, all 3 curves looked quite similar to the learning curves obtained using this approach on data without noise added to it as presented in (Champaign and Cohen 2010). In Figure 1 (a)(b)(c), all three variations of the algorithm are performing well, in getting close to attaining the ideal average level of knowledge for students (i.e. the greedy god) by the end of the 200 trials. Note, as expected, simulated annealing takes longer to converge, as it is coping with random information at the beginning.

An initial concern was that our experiment might somehow be accidentally determining the appropriate learning objects without relying on the assessment. To test this concern, we replaced the assessment function with a function that randomly provides one of the 18 discrete levels (instead of an assessment, it provides a random grade). Since the three variations on this approach (ecological, ecological with pilot and simulated annealing) all rely on assessments to function, our expectation was that this change would produce 3 curves that were degraded to the performance of the random baseline. This is the result we saw.

[5] Even with a ideal assessment tool, there will still be situations where student mistakenly give incorrect information that they understand (known as a slip) or accidentally give the right answer to something they don't understand (known as a guess).

[6] It is important to note that there are different approaches to model a "bad assessment". By randomly adding noise, we're modeling an assessment that has variability in every assessment. This does not model an assessment with a systemic bias, for example, one that always evaluates C+ students as D students.

[7] The idea is that if a student could be modelled with an erroneous assessment level (e.g. B vs. A) then with greater standard deviation, the likelihood of an erroneous label increases. Note values closer to the true value will still be the most likely to be assigned.

In all, these results tell us that this approach is, in fact, highly robust with noisy data. As long as there is a tendency for an assessment to be closer to a correct value than an incorrect value, this approach will steadily improve the curriculum sequence suggestions as more data is obtained. Realistic amounts of noise, which are expected with any assessment, would seem to be acceptable to the functioning of this approach.

What is happening with our approach is the following. Suppose the error in assessment led to an inappropriate learning object being proposed for a new student (e.g. the previous student was assessed as deriving benefit from that object where, in fact, he had not). The simulation would model this new student's interaction with the learning object. Now, this should reflect a poor increase in knowledge (i.e. the student's knowledge level is not attuned to the target level of knowledge (Equation 2)). When the new student's assessment is modeled, therefore, this will be attached as one of the experiences with that learning object. As a result, this learning object would now be less likely to be assigned to new peers.

Part of the power of this peer-based algorithm is the ability to correct mistakes. If a recommendation is made because of an inaccurate assessment (that is, an interaction that was actually harmful is instead recorded as being useful) this will lead to further recommendations of that object to similar students. However, when those similar students use the object, they will in turn be assessed. As this approach considers all previous interactions, with more students interacting with the object, a larger history of interactions will accumulate. The average of these assessments will approach the true value, even if some of those assessments are distorted by noise. As this happens, the system is less likely to recommend the bad object, and will increasingly direct students to a better choice, leading to a self-correcting system.

While our initial feeling was that a standard deviation of 0.5 was a large amount of noise, we then ran an experiment with a standard deviation of 1.5 (see Figure 1 (d)). The consequence of this is that we're adding noise which is very likely to move data points anywhere in the range (with a standard deviation of 1.5, there is roughly a 25% chance of a perfect knowledge of 1 being mapped to a F-). With this massive amount of noise being added, we then saw the degradation we had initially expected (with the ecological condition converging on 0.8 instead of 1.0).

Variable Time of Instruction

In the previous approach outlined the Background section, which learning object should be assigned to a particular student is dependent on similarity of peers and the previous learning benefit obtained by those peers, alone. We explored a new extension, where we incorporate reasoning about the length of time it takes to complete an interaction with a learning object as well.

Clearly, in real learning situations, learning events can take variable amounts of time. Watching a recording of a lecture might take 76 minutes, while attending a day long seminar might take 8 hours. Rather than making the simplifying assumption that each interaction with a learning object will take an equivalent length of time, we can incorporate this concept into our reasoning.

calculateBenefit in Line 8 of Algorithm 1 then needs to be modified to incorporate time. Rather than consider the benefit of the learning object, we can think of the proportionate benefit, that is, how much benefit it provides per minute of instruction (assuming a repository where each learning object's average time to completion is recorded). This can be calculated by dividing the benefit of the learning object by the length of time it takes to complete the interaction for the average student.

We are interested in ensuring that, with this more sophisticated consideration incorporated, the approach outlined in Algorithm 1 continues to provide worthwhile recommendations for curriculum sequencing.

TIME: approach

We modified the previous approach (Champaign and Cohen 2010) such that, as well as generating a random set of target instruction levels for each learning object, we also generated a random length of completion (ranging from 30 to 480 minutes). We used 50 students, 100 learning object and three runs – an error of 0.05, 0.1 and 0.5 standard deviation – each time for 20000 minutes of simulated instruction. As well as the random and greedy god baselines, we again considered the pure ecological, ecological with pilot and simulated annealling variants. These results are displayed in Figure 2 (a), (b) and (c).

It is worthwhile to note that initially we experimented with about 2400 minutes of instruction, based on this being roughly the amount of instruction in a typical university course. This was determined to be far too short a length of experiment as the learning curves reflected only the initial part of the graphs shown here. Our conclusion was that we were simply failing to see, yet, the benefit to learning that the students achieve and that either longer lesson times were needed or that it may be valuable to track students over multiple classes.

TIME: results and discussion

With the increased time provided in Figure 2 (a) (b) (c), we did indeed attain the kind of student learning that we expected (reasonably high average level of knowledge, for students). With the added sophistication of allowing learning objects to require different lengths of times to complete, this approach continues to make worthwhile curriculum recommendations to students. The fact that all three variants on the algorithm are approaching the ideal of the greedy god at the end of the trials is encouraging. As expected, the greater the standard deviation, the more challenged each algorithm is to attain appropriate student knowledge levels, but the differences between Figure 2 (a)(b)(c) are still relatively minor.

In some of our runs with varying standard deviations of error, we saw that in the early stage (up to 2000 minutes) the ecological with pilot variant would outperform the greedy god. This should be impossible, since the greedy god is an upper bound benchmark. Our theory about what is happening is that the pilot group interacts with learning objects, and

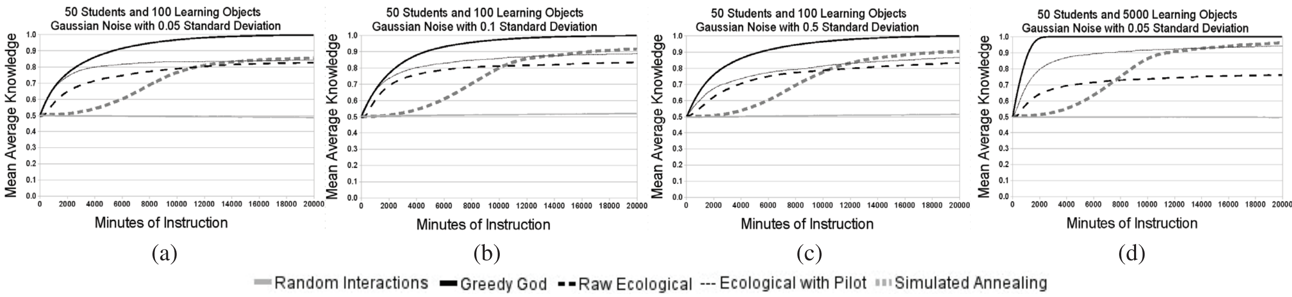

Figure 2: Comparison for Varying Standard Deviation of Error Including Time of Instruction

because interactions with short learning objects can be completed more quickly, we will naturally tend to have more data about them. This leads to a bias towards short learning objects. Since the learning isn't assessed until after the completion of a learning object, if the greedy god approach assigns a student to an (optimal) 8 hour learning object, this won't be reflected in the average knowledge for the group of students for 480 minutes. It would be expected that the greedy god will, in the long run, overtake the ecological with pilot variant, which is what happened.

One approach to avoiding this would be to assess the students' knowledge every minute: Rather than waiting until the completion of the interaction, assess the student's every minute and apply $\frac{1}{\text{length of lesson}}$ to their knowledge. Using such an approach, the greedy god should be the upper bound, even during the early stages.

Another unusual feature is that the learning curves approach a final knowledge less than 1, whereas in the experiments of Figure 1 they approached one. Initially this was thought to be a consequence of the introduced error; however, considering the curves from the error section above does not support this idea. It is possible that each approach is again developing a bias towards short lessons, and is therefore not taking advantage of the full range of learning objects that may help the students approach complete mastery.

One valuable extension to this work therefore would be to incorporate a requirement that students must complete a variety of learning objects of varying lengths. We believe that our experiments run with this stipulation would no longer produce the early stage anomalies.

In addition, historically learning gain has been the accepted metric for measuring a learning event for ITS researchers. One alternative which is being considered is is to use the proportional learning gains proposed by (Jackson and Graesser 2007), is defined as: $\frac{\text{post-test}-\text{pretest}}{1-\text{pretest}}$. This would be another useful alternative for avoiding a bias towards interactions where a student has a low initial score, if this formula were used instead on the right hand side of the equation in line 8 of Algorithm 1. For example, with this formula, advancing from A to A+ is a greater learning gain than advancing from B to B+.

Large Corpus

It has been suggested (McCalla 2004) that ecological approaches, rather than degrading with large amounts of information, improve. Intuitively this makes sense, with more data better recommendations should be possible. In order to investigate this, we considered a student group interacting with a large library of learning objects (5000 objects). Collecting a massive amount of educational content offers a valuable resource, but also introduces the challenge of navigating a large corpus.

When we consider a simulation with dramatically more learning objects (Figure 2(d)), we see that both the simulated annealing and the ecological with pilot learning curves become steeper. This corresponds with McCalla's prediction. The ecological with pilot group has a sustained improvement and outperforms the raw ecological more dramatically than in previous experiments. Similarly, the simulated annealing conditions performs well with a larger library. The additional exploration of the corpus available to these algorithms in their initial phases appears to be providing some valuable benefit. Note that these curves approach the ideal average knowledge of the greedy god to a greater extent with the larger repository (compared to the small repository used in Figure 2 (a)(b)(c)), which again confirms McCalla's hypothesis.

Conclusion and Discussion

In this paper, we have outlined some key extensions to an existing model for curriculum sequencing in a peer-based intelligent tutoring environment. This model is distinct in its use of the previous learning experiences of peers, rather than simply relying on peers to collaborate in real-time, to assist in student learning (as in (Cheng and Vassileva 2006)). This approach focuses on determining the most appropriate, similar peers and the most appropriate learning objects (those that provided the most benefit to those similar peers). In addition, that model was validated through the use of simulated students, modeling the knowledge levels expected for each learning object and leveraging a pre- and post-test assessment of each student, to measure whether effective learning had occurred.

The extensions that we introduce here first of all focus on our methods for validating the model. We are able to demonstrate that even in the presence of some error in assessment, the content sequencing algorithm that forms the basis of the student learning is providing very effective learning objects for our simulated students. These results not only confirm the value of this particular peer-based tutoring approach but also reinforce the appropriateness of using simulated students as an important part of validation. The simulations do rest on a specification of student assessments, but the methods that would be used to set these values can tolerate some inaccuracy.

As a result, we feel that our work continues to promote the value of simulated student validations, for other researchers as well. Our use of simulations differs from that of others such as (VanLehn, Ohlsson, and Nason 1996; Beck 2002; Matsuda et al. 2007) who all used simulated students as a technique for understanding the behaviour of real students. In particular, in (Matsuda et al. 2007) after training a simulated student using logs of interactions with real users, the simulated student they developed could explain 82% of correct problem solving steps performed by subsequent students. This approach has the benefit of creating a cognitive model through demonstration alone. Our system is different from these, in that our simulated students are used entirely to evaluate the efficacy of our techniques, and are not used as peers for human students or to predict their actions.

Our methods also contrast with those of others who conduct studies with actual human learners (e.g. (Jackson and Graesser 2007)). These researchers may be interested in examining the value of their approaches for much larger populations of students and thus the use of simulations may be of value. In our future research, we anticipate eventually conducting studies with (a modest number of) human users as well. The robustness of the Collaborative Learning Algorithm to errors in assessment also provides encouragement for coping with inaccuracies in assessments which would undoubtedly occur. Simulations allow us to observe the benefit of our approach in a environment with a very large number of students.

The other primary direction for our extensions was to explore the modeling of the time demands of learning objects. Our results indicate effective learning by students when decisions about what to present to these students takes the time to completion into consideration. We feel that this offers a new direction for other designers of intelligent tutoring systems, suggesting that not only the inherent value of the content but the overall demand, in time, on the student should be brought to bear when determining the student's overall curriculum. We have also noted some avenues for future work in our exploration of time demands, evolving to a requirement of both short and long duration learning objects for students, rather than simply a total time restriction.

A final insight that is gained through the process of extending our model and its validation is information about how the size of a student population affects the student learning that can be achieved, in peer-based tutoring environments. McCalla hypothesized that as the peer base grows, student learning can improve. Our experimental results show that with larger repositories there are larger learning gains. This also suggests a direction for future research in the design of intelligent tutoring systems that learn on the basis of peers, trying to measure and quantify the relative value of an increased population. Our observation is that the reputability of these peers will then be an important consideration – a larger population of less valued peers will likely in fact detract from the learning that can be achieved. We plan to explore trust-based modeling of peers in intelligent tutoring environments, using models such as (Zhang and Cohen 2008); to this end, the work of (Gorner and Cohen 2010) which uses trust modeling to investigate how to determine the ideal size of social network for providing appropriate advice may also be of some interest.

References

Beck, J. E. 2002. Directing development effort with simulated students. In *ITS '02: Proceedings of the 6th International Conference on Intelligent Tutoring Systems*, 851–860. London, UK: Springer-Verlag.

Breese, J. S.; Heckerman, D.; and Kadie, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. 43–52. Morgan Kaufmann.

Champaign, J., and Cohen, R. 2010. A model for content sequencing in intelligent tutoring systems based on the ecological approach and its validation through simulated students. In *Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference*.

Cheng, R., and Vassileva, J. 2006. Design and evaluation of an adaptive incentive mechanism for sustained educational online communities. *User Model. User-Adapt. Interact.* 16(3-4):321–348.

Gorner, J., and Cohen, R. 2010. Optimizing advisor network size in a personalized trust-modelling framework for multi-agent systems. In *Thirteenth International Workshop on Trust in Agent Societies (TRUST-2010)*.

Jackson, G. T., and Graesser, A. C. 2007. Content matters: An investigation of feedback categories within an ITS. In *Proceeding of the 2007 conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work*, 127–134. Amsterdam, The Netherlands, The Netherlands: IOS Press.

Matsuda, N.; Cohen, W. W.; Sewall, J.; Lacerda, G.; and Koedinger, K. R. 2007. Evaluating a simulated student using real students data for training and testing. In Conati, C.; McCoy, K. F.; and Paliouras, G., eds., *User Modeling*, volume 4511 of *Lecture Notes in Computer Science*, 107–116. Springer.

McCalla, G. 2004. The ecological approach to the design of e-learning environments: Purpose-based capture and use of information about learners. *Journal of Interactive Media in Education: Special Issue on the Educational Semantic Web* 7:1–23.

VanLehn, K.; Ohlsson, S.; and Nason, R. 1996. Applications of simulated students: An exploration. *Journal of Artificial Intelligence in Education* 5:135–175.

Zhang, J., and Cohen, R. 2008. Evaluating the trustworthiness of advice about seller agents in e-marketplaces: A personalized approach. *Electronic Commerce Research and Applications* 7(3):330–340.