Retrospective Evaluation of Blended User Modeling For Adaptive Educational Systems

Michael Yudelson, Sergey Sosnovsky

University of Pittsburgh, School of Information Sciences, 135 N. Bellefield Ave.,
15260 Pittsburgh, PA, USA
{mvy3, sas15}@pitt.edu

Abstract. In this paper, we are presenting a retrospective approach to evaluating user models by utilizing previously collected learning logs rather than setting up a new experiment. This approach is applied in a novel way to modeling heterogeneous types of user activity – problem solving, and browsing annotated examples. We are blending the two types of activity in the user model in an attempt to increase the accuracy of the composite model. Obtained results suggest that such blending, in fact, does make a difference both for users individually and on a global scale.

Keywords: user modeling, evaluation, model blending, adaptive educational systems.

1 Introduction

The best way to determine the quality of an adaptive system is through a carefully planned empirical evaluation with human subjects. The evaluation design can vary from a short-term controlled experiment to a longitudinal study, but before the system is put into use its value is rather unknown. The system under evaluation is usually considered as a black box, that influences the depended variable as a whole. However, it is not always clear what do we really measure when evaluating the quality of an adaptive system. The effect or the value of adaptation observed in such experiments can be attributed to several things: the accuracy of user modeling, the effectiveness of adaptation strategies, or the quality of the content.

One of the known alternatives to the holistic view on adaptive system studies is layered evaluation [1, 2]. It implies that the user modeling component and the adaptation component of an adaptive system are assessed independently. The evaluation of a user modeling component is based on its accuracy, or predictive validity, which defines how well the model represents the actual state of the user and how reliably it can predict user's next action [3]. In the context of adaptive education, it can be interpreted as the model's ability to predict the result of the student's next attempt to apply a concept or answer a problem.

An interesting opportunity that this approach opens for experimenters is the implementation of several modeling algorithms operating on the stored log of users' activity and comparative evaluation of these algorithms based on their predictive validity. Such retrospective analysis allows the reuse of once collected data for multiple evaluation experiments based on “what-if” scenarios aimed at pre-selection of an optimal user modeling approach [13]. Naturally the optimality of such pre-selection is limited to the user modeling layer. The presence of adaptation that is based on the values supplied by the user model, would add an additional factor. An overall cross-layer empirical evaluation would be necessary to make a final assessment.

In this paper, we apply retrospective evaluation to choose the best value for a singe parameter in the modeling formula. The data set is the log of students’ learning activity with two types of education content. The user modeling algorithm used this log to populate overlay models of students’ knowledge. However, different types of activity were processed independently to compute parallel student models on two different cognitive levels: comprehension level (corresponding to example-browsing activity) and application level (problem-solving activity). Our main goal is to find whether a blending of the user models that correspond to the two cognitive layers can result in a better composite model with higher predictive validity.

The rest of the paper is organized as follows. Section 2 talks about the original approach to building user models from cognitively heterogeneous educational activity. Section 3 discusses user modeling without blending. Section 4 proposes a modification to the modeling approach and introduces blended user modeling. Section 5 outlines the hypotheses and goals of this experiment, which is presented in Section 6. Finally, section 7 concludes the paper with an extended discussion of the obtained results.

2 Modeling From Heterogeneous Student Activity

Many e-learning environments provide students with various types of educational content (learning problems, examples, tutorials, interactive simulations, etc.) that contribute to different levels of material understanding. Several adaptive systems integrate or provide means for integrating such components (e.g. [4, 5]). One of the problems for these systems is to incorporate evidence coming from heterogeneous sources into a student model that would help to deliver viable adaptation. Our previous solution was not to fuse these activities, essentially, maintaining a set of parallel models of student knowledge, each populated by a specific kind of learning activity. The levels of student modeling where taken from the Bloom’s taxonomy of educational objectives [6]. For example, reading a textbook would contribute to the “knowledge” level of the Bloom’s taxonomy; exploring examples – “comprehension” level; answering problems and quizzes – “application” level, etc. However this approach does not take into account the transfer between the categories of the Bloom’s taxonomy: mastering a lower level of activity should also influence the higher level(s).

Over the last several years, we have accumulated a rich collection of user activity logs from student of several undergraduate and graduate level courses using a set of our systems in a number of learning domains. A tangible portion of the logs covers problem solving and browsing annotated examples that correspond to the application and comprehension levels of Bloom’s taxonomy. A question for this study is whether modeling the transfer between different cognitive levels of the user model (in this case, the comprehension and application levels) can be quantitatively detected, i.e. whether this transfer would improve the accuracy of our user models. We try to explore this effect by combining or *blending* different tiers of the user model retrospectively and re-evaluating each blend by computing the prediction validity of the composite user model.

3 User Modeling Without Blending

There is an abundance of approaches to user modeling. A great number of them follow the overlay paradigm, when a user model is calculated with respect to a set of concepts, skills, or preferences. The user modeling component processes evidence of a user's interaction with a content item and updates relevant portions of the overlay vector, spanning the domain. One such approach has been implemented in the user modeling server CUMULATE [7].

CUMULATE builds several types of user models resulting from different types of user activity. The ones that are of interest to our discussion here are: the model of example browsing (the comprehension level of Bloom's taxonomy), and the model of problem solving (the application level of Bloom's taxonomy). For each of the models, CUMULATE uses a different technique to compute knowledge levels. In the case of example browsing, CUMULATE tracks percent of example lines explored. When that percentage reaches 80%, all of the concepts relevant to this example are considered known (on the comprehension level).

Modeling problem solving in CUMULATE is done in a more complicated way. Each of the concepts with which a problem is indexed, has a weight. This weight is produced during indexing and denotes the importance of that concept in mastering the problem. Concept weights are used in distributing the total amount of updates a user model receives. CUMULATE also has a safety mechanism discouraging users from over-practicing one particular exercise. This over-practicing gradually decreases the knowledge updates when users solve one particular problem correctly more that one time. Thus users are motivated to attempt solve a diverse set of problems in order for their user models to grow. Refer to equations (1) and (2) for details.

$$k_{n+1} = k_n + res \cdot (1 - k_n)^2 \cdot \begin{cases} k_n \le .5 & {}^{w}\!/_{2} \\ k_n > .5 & w \end{cases} \qquad (1)$$

$$w = \sqrt[4]{\frac{w_{c,p}}{\sum_i w_{c_i p} \cdot \left({}_{succ}att_p + 1 \right)}} \qquad (2)$$

The initial value of a concepts knowledge k_0 is 0. With every correct solution of the problem (where *res*=1 in (1)), all of the related concepts receive an update. This update is directly related to: a) the amount of knowledge this concept can grow by squared ($(1 \; - k_n)^2$ in (1)), and b) to a special weighting factor (2). This weighting factor is composed of weights ratio and over-practicing penalty. Weights ratio is the weight of the currently updated concept in the problem ($w_{c,p}$ in (2)) over the sum of all weights involved in the problem ($\Sigma_{\mathrm{i}}\, w_{c,p}$). The problem's over-practicing penalty is one over number of successful solutions to this specific problem by a particular user plus one ($_{succ}att_p$ + 1). When the prior knowledge level is below 50% the weighting factor is halved (1). This is done to prevent initial leaps in knowledge level.

4 Blending Problem-Solving And Example Exploration

Over several years we collected user activity and modeling user knowledge in CUMULATE. We noticed that, while practicing problem solving does provide a faster way to acquire knowledge, users do spend significant time reviewing annotated examples. This suggests that examples are in fact an important part of learning and that there may be a better way to incorporate example browsing into computing the user model than the one we have described in the previous section.

Intuitively there should be some form of transfer between comprehension and application tiers of the user model. There might not be direct impact, of course, as problem solving requires deeper understanding of the domain than mere clicking and looking could hope to achieve. However, a limited influence of example browsing is not at all impossible.

We have modified equation (1) to reflect the possible comprehension-to-application level transfer. Refer to equation (3). The only difference is a *B* weight. This weight is 1 for problem solving, making equation (3) identical to equation (1). In the case of example browsing, *B* would constitute a blending coefficient: value from 0 to 1. 0 – meaning no blending whatsoever – without considering example browsing, and 1 – meaning example browsing is as important as problem solving. Other than the B weight, the updates to the knowledge level of the concepts are done in the same manner on the unified problem- and example-related user model.

$$k_{n+1} = k_n + res \cdot B \cdot (1 - k_n)^2 \cdot \begin{cases} k_n \le .5 & w/2 \\ k_n > .5 & w \end{cases}, \qquad (3)$$

After some experimentation, we found that in addition to blending coefficient we should take into account the amount to which the example was explored. Truly, we cannot equally consider user activity in case the example is fully explored and when only say 1 out of ten lines were reviewed. To take that into account, for examples-related activity modeling we have decided to define *B* in equation (3) as a product of blending coefficient and percentage of example lines explored.

5 Hypotheses And Goals

Our hypotheses regarding blending comprehension and application layers of user mode are the following.

1. In general, blending example activity (evidence of concepts' comprehension) and problem solving (evidence of concepts' application) increases the accuracy of user modeling.
2. Different users benefit from different blends.

The goals that we are trying to reach in this study are.

1. Find a universally optimal blend of comprehension and application levels in the user model, if such exists.
2. If possible, determine and describe groups of users that can benefit from different blending conditions.

6 Experiment

6.1 Experimental Setup

To evaluate our hypotheses and meet our goals regarding blending layers of the user model belonging to different levels of Bloom's taxonomy, we have set up a computational experiment. We used student activity logs that were collected during Fall 2007 and Spring 2008 semesters from 4 database design courses offered at both the University of Pittsburgh (1 graduate and 2 undergraduate courses), and Dublin City University (1 undergraduate). All 4 courses, although slightly different in structure, were roughly identical with respect to the content. Each course consisted of a set of topics. Every topic had a set of SQL writing problems provided by SQL KnoT system [8] and a set of annotated SQL code examples supplied by the system WebEx [9]. Both SQL KnoT and WebEx were introduced to students roughly in the beginning of each of the semesters. The use of these systems was optional and did not impact the students' grades. Overall, there were 48 problems and 64 examples available to the students.

The number of students, as well as their level of participation, varied across semesters and is summarized in Table 1 along with basic usage statistics.

Table 1. Basic user participation statistics across semesters and courses.

School	Semester	Level	No. of users	Avg. problem attempts	Avg. example views	Avg. distinct problems	Avg. distinct examples
U. of Pitt	Fall 2007	U*	27	156.40	189.00	29.96	32.07
U. of Pitt	Fall 2007	G	20	61.70	104.70	29.95	29.10
U. of Pitt	Spring 2008	U	15	26.94	46.65	16.35	10.29
DCU	Spring 2008	U	52	81.68	257.25	22.82	38.63

* U – undergraduate, G – graduate

All student activity with both problems (SQL KnoT) and examples (WebEx) has been logged by the CUMULATE user modeling server. Each problem and example has been indexed with a set of metadata concepts with the help of a semi-automatic grammar parser. The concepts came from an SQL ontology, developed by domain experts. The indexes were double-checked afterwards.

6.2 Experimental Procedures

For each of the semester logs, we have (re)-computed several blended user models. First of all, a 0-blend was computed; here, no example activity was taken into account – only problem solving activity was modeled. 0.1, 0.2, ... 0.9, and 1.0 blends corresponded to user models where updates resulting from example activity were weighted from 0.1 to 1.0 with 0.1 steps. This gave us 4 semesters * 11 blends = 44 clusters of user models or 114 users * 11 blends = 1254 user models. A classical accuracy measure (correct predictions over all predictions) was computed for each user model.

Prior to proceeding with testing of our hypotheses, we filtered user models. The filtering condition was that the user had to attempt to solve at least 33% of the problems (15 out of 48) and view at least 33% of examples (22 out of 64). The reason behind this threshold was that, in order to improve problem solving model by blending it with example browsing model, both have to be well populated. Namely, the user had to work with both examples and problems to a significant extent.

After the filtering, the number of users in each semester/class dropped to the values shown in Table 2. Thus, the initial number of 114 users was reduced to 56 users.

Table 2. Number of qualified users after applying filtering.

School	Semester	Level	No. of users	No. of qualified users
U. of Pitt	Fall 2007	U*	27	14
U. of Pitt	Fall 2007	G	20	10
U. of Pitt	Spring 2008	U	15	3
DCU	Spring 2008	U	52	29

* U – undergraduate, G – graduate

6.3 Results

To get a general idea about the usefulness of blended models for each user, we have selected the best non-0% blend (10% to 100%) and ran a left-tailed paired *t*-test. Individual best blends turned out to be significantly better then 0% blends with t = -5.38, p-value<.001. The average edge of each student's best blend over 0% blend was .015 or 1.5% in terms of accuracy. Mean standard deviation of blended model accuracies across users was .0113 or 1.13%. The minimum standard deviation was 0% and the maximum was 10%.

To select a universal useful blend we ran 10 left-tailed paired *t*-tests, in each case comparing 0% blend to one of 10 non-0% blends. Here, 40% and 50% blends turned out to be the most potent ones and the only ones with significant edge over 0%-blend (both with t = -2.05 and p-value = .023). The average advantage of 40% and 50% blends over 0% blend dropped to .56%. As we can see, "universal" blends lose to individually tailored blends.

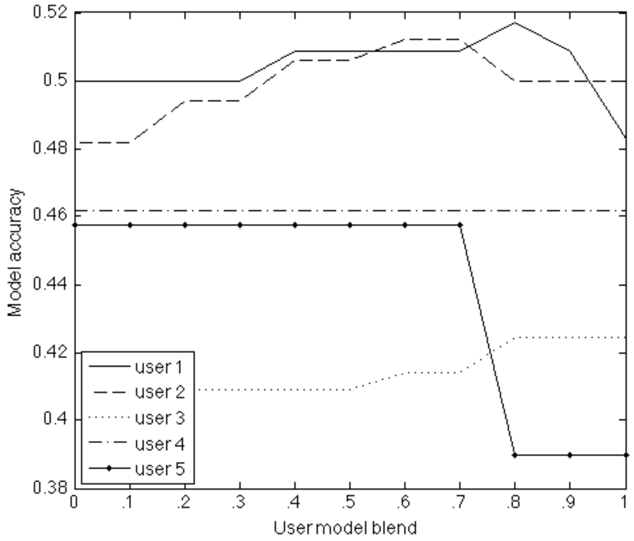

Fig. 1 Examples of blend effect on user model accuracy.

Before further exploring individual user differences with respect to blends, let us refer to Fig. 1, where 5 sample users are represented with a graph of blending percentage vs. accuracy. Here we can see that the model of user 4 is not sensitive to blending whatsoever: the accuracy does not change with respect to blends. In the case of user 5, blending has no effect till 70% blend after which accuracy drops. Blending does help to improve user models for users 1,2, and 3.

One feature of the blended models apparent in Fig. 1 is that different users have different numbers of points of maximum accuracy. Graph of user 4 is flat, giving us 11 points of maximum (or no maximum at all). User 5 has 7 points of maximum, and users 1, 2, and 3 have 1, 2, and 3 points of maximum accuracy respectively. Fig. 2 shows the distribution of the number of maximum accuracy points for blended models of all 56 users.

Instantly, we can notice a group of "no difference" consisting of 15 users for which blending doesn't improve the user model. The rest of the range of the number of maximum blends can be subdivided into the "low" group (1 maximum) of 2 users, the "medium" group (2-4 maximums) of 22 users, and the "high" group (5-9 maximums) of 17 users.

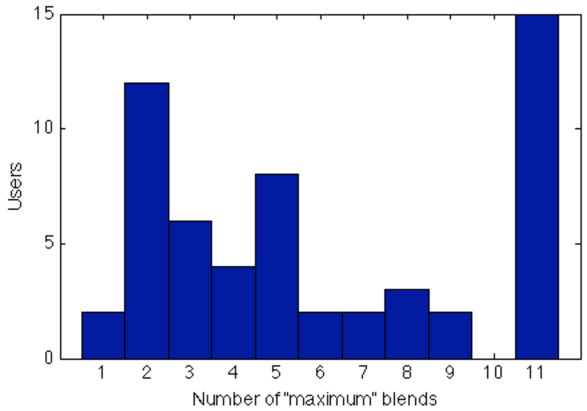

Fig. 2 Distribution of number of users for different numbers of peak (maximum) blends.

The “low” group consists of the two rare cases of a user having just one best blend. Both users prefer high blends of 80% and 100% respectively. Users in the “medium” group have an inclination towards higher blends. Since our data did not meet the requirements of the parametric test (paired *t*-test), we used its non-parametric analog Wilcoxon signed-rank test. Out of 10 tests the most potent belongs to 90% blend with *p*-value = .037.

Users of the “high” group follow the global trend. Out of 10 Wilcoxon signed-rank tests the ones corresponding to 40% and 50% blends turn out to be equally significant. Both with *p*-value = .049.

7 Discussion

We are able to see from the data that blending comprehension and application tiers of user model in fact does make a difference both for users individually and on a global scale. Namely, there is a benefit in (partially) scoring example browsing as if it was problem solving, and there is a transfer effect between cognitive layers of the model. The major downside is that, although statistically significant, the difference is quite small: on the order of few percent.

Nevertheless, there is a clear indication that, with respect to blends, users do differ in what blend works best for the higher accuracy of their model. We also believe that there is a way to pinpoint both individual and global blending effect better.

One of potential ways to improve is to contextualize the model. As described in Section 3, modeling in CUMULATE follows the *one-fits-all* schema. However, as it has been shown in [10] each item of the problem space, as well as each user, possess individual features. With respect to problems, each has its inherent complexity not always captured by the metadata index. Knowledge of concepts does not grow equally fast for all of them and does not always starts from same value (0 in our case).

Making appropriate adjustments in user modeling to accommodate these differences has a chance to improve the modeling itself and help to find an optimal blend of Bloom's user model tiers both on global and individual scale.

Another issue with an exploration of the blending effect is that we had to filter nearly 50% of the users out. Ideally, for the blending to have a tangible effect, both example browsing and problem solving behaviors have to be well established: the user has to work enough with both types of learning resources.

A prospective remedy here could be to shift from number of distinct learning resources covered to the amount of metadata overlap. Instead of counting how many examples were viewed or problems were attempted, it might be more beneficial to trace the overlap of the domain concepts that both examples and problems addressed.

One important thing to mention is that in all of the reported studies some form of adaptive navigation support was available to users and this could potentially have affected our measurements. The navigation support was expressed in the form of a descriptive icon next to the link that opened an example or a problem.

An aspect that still remains unaddressed is the temporal dimension. It might be the case that the optimal blending of the user model layers is not persistent over time. As users progress through the course, the best blend may change for them. It would be challenging to detect these changes, as users would have to stay very active for the whole duration of the course and generate enough log data to analyze. From our own experience, the proportion of such motivated users is very low in every class and often they are outstanding in various regards: both in positive and negative sense.

For our future work, we would like to apply the blending of cognitive layers of the user model in a longitudinal study. This might help us to see a clearer differentiation between blending factors and assist in making cognitive layer blending preferences explainable more transparent.

Also we would like to test our blending approach in different learning domains such as learning C or Java. In addition, we would like to test other approaches to user modeling such as knowledge tracing [11] and/or learning factor analysis [12].

Acknowledgments. We would like to express special thanks and appreciation to Peter Brusilovsky for support, insights, and valuable feedback provided as this work took shape.

References

1. Paramythis, A., & Weibelzahl, S. (2005). A Decomposition Model for the Layered Evaluation of Interactive Adaptive Systems. In Ardissono, L., Brna, P., & Mitrovic, A. (Eds.), Proceedings of the 10th International Conference on User Modeling (UM2005), Edinburgh, Scotland, UK, July 24-29 (pp. 438-442) (Lecture Notes in Computer Science LNAI 3538, Springer Verlag). Berlin: Springer.
2. Brusilovsky, P., Karagiannidis, C., & Sampson, D. (2004). Layered evaluation of adaptive learning systems. International Journal of Continuing Engineering Education and Lifelong Learning, 14(4/5), 402-421.
3. Corbett, A.T., Anderson, J. R. and O'Brien, A. T.: 1993, The predictive validity of student modeling in the ACT programming tutor. In: P. Brna, S. Ohlsson and H. Pain (eds.),

Artificial Intelligence and Education, 1993: The Proceedings of AI-ED 93. Charlottesville, VA: AACE.

4. Denaux, R., Dimitrova, V., and Aroyo, L. 2005. Integrating Open User Modeling and Learning Content Management for the Semantic Web. In L. Ardissono, P. Brna & A. Mitrovic (eds.), Proceedings of 10th International Conference on User Modeling (UM'2005), Edinburgh, Scotland, UK, 23-29 July (pp. 9-18).
5. Trella, M., Carmona, C., and Conejo, R. 2005. MEDEA: an Open Service-Based Learning Platform for Developing Intelligent Educaional Systems for the Web. In Proceedings of Workshop on Adaptive Systems for Web-Based Education: Tools and Reusability at AIED'05, Amsterdam, The Netherlands (pp. 27-34).
6. Bloom, B. S. (1956). Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain. New York: David McKay Co Inc.
7. Brusilovsky, P., Sosnovsky, S. A., & Shcherbinina, O. (2005). User Modeling in a Distributed E-Learning Architecture. In: L. Ardissono, P. Brna, & A. Mitrovic (Eds.), 10th International Conference on User Modeling (UM 2005), Edinburgh, Scotland, UK, 2005 (pp. 387-391). Springer.
8. Brusilovsky, P., Sosnovsky, S. A., Lee, D. H., Yudelson, M., Zadorozhny, V., & Zhou, X. (2008). An open integrated exploratorium for database courses. In: J. Amillo, C. Laxer, E. M. Ruiz, & A. Young (Eds.), 13th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE 2008), New York, NY, USA, 2008 (pp. 22-26). ACM.
9. Brusilovsky, P. (2001). WebEx: Learning from Examples in a Programming Course. In: W. A. Lawrence-Fowler & J. Hasebrook (Eds.), World Conference on the WWW and Internet (WebNet 2001), Orlando, Florida, 2001 (pp. 124-129). AACE.
10. Corbett, A. T. & Anderson, J. R. (1992). Student Modeling and Mastery Learning in a Computer-Based Programming Tutor. In: C. Frasson, G. Gauthier, & G. I. McCalla (Eds.), 2nd International Conference On Intelligent Tutoring Systems (ITS'92), Montréal, Canada, 1992 (pp. 413-420). Springer.
11. Corbett, A. T. & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modeling and User-Adapted Interaction, 4(4), 253-278.
12. Cen, H., Koedinger, K. R., & Junker, B. (2006). Learning Factors Analysis - A General Method for Cognitive Model Evaluation and Improvement. In: M. Ikeda, K. D. Ashley, & T. Chan (Eds.), Intelligent Tutoring Systems, Jhongli, Taiwan, 2006 (pp. 164-175). Springer.
13. Yudelson, M. V., Medvedeva, O., & Crowley, R. S. (2008). A multifactor approach to student model evaluati*on. User Modeling and User-Adapted Interaction, 18(4), 349-382.