Laying a Foundation for the Graphical Course Map

Linda DuHadway and Thomas C. Henderson
University of Utah

UUCS-16-001

School of Computing
University of Utah
Salt Lake City, UT 84112 USA

21 January 2016

Abstract

The transition from standard classroom courses to on-line delivery systems has been stymied due to both technical and human interface issues. One of the limitations is the text-based, linear presentation of course materials available in the learning management system. We have developed a graph-based approach to presenting the learning materials of a course using a system called **ENABLE** [6, 7] with three major goals: (1) facilitate restructuring a set of synchronous classroom materials into a dynamic on-line system, (2) provide algorithms to analyze and enhance student performance as well as provide insights to the instructor concerning the efficacy of the learning items and their organization, and (3) identify ways to use data from an existing linear, temporal based course presentation to train predictive models for a course that allows individual flexibility in the ordering of the material. This work demonstrates the possibility of presenting course materials in a graphical way that expresses important relations and provides support for manipulating the order of those materials. Making a fundamental change in how course materials are presented and interfaced with may potentially make educational opportunities available to a broader spectrum of people with diverse abilities and circumstances. The graphical course map can be pivotal in attaining this transition.

1 Introduction

The availability and accessibility of education over the Web has increased but barriers remain [1, 8, 11, 13]. Current learning management systems (LMS) display learning materials in a textual, linear format primarily based on chronology. This presentation of material provides a limited view of the course and reaches only some of the potential users of online educational tools. Expanding the delivery of learning material to include a graphical course map can increase the information that is available and make that information accessible to a larger number of diverse consumers. Making educational information available to a broader spectrum of users has the potential to include more people in the educational process and improve their opportunities for inclusion and success.

As more and more educational opportunities are being made available on the Web there is a greater need for tools that can present those materials in a more accessible way. An online course is not limited to a linear, chronological organization that has been the preferred presentation of the traditional classroom. Our research focuses on the possibilities of presenting learning materials in a graphical course map. This has led to many discoveries about the opportunities for enhancing the information available to students and educators. The development of a variety of course maps has identified new ways to organize and present learning materials and restructure their delivery to exploit the flexibility of the online setting.

Improving accommodation for people with different abilities and a wide range of circumstances can be augmented by removing the temporal limitations of the traditional text-based, linear presentation of course materials. To facilitate a different presentation of the learning material there is a need to focus on more functional relations between learning items and presenting those relations in a graphical way. The **ENABLE** system provides interactive tools that allow an instructor to discover important relations between the learning items and manipulate a variety of graphical course maps that maintain those relations.

Consider a standard classroom course consisting of ten learning items. Figure 1 shows an example ten item course presented graphically. The course map on the left directly reflects the linear, temporal based ordering found in the LMS. The course map on the right is reorganized according to the *prerequisite* relation. See [6, 7] for a description of algorithms to achieve these results.

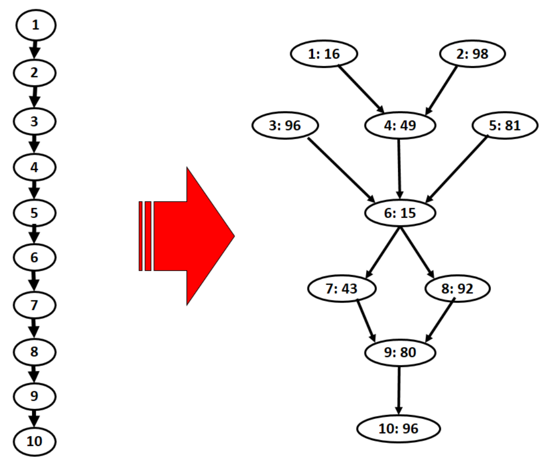

Figure 1: Example of Converting a Standard Synchronous Class to an On-line Class (the First Number is the Learning Item Number, the Second is the Difficulty).

2 Discovering Existing Data

A beginning point for initiating change is to discover where one currently is. The first stage of this work was to discover what data is currently available and how that data might be used to provide meaningful information for building course maps. Three existing computer science courses were analyzed.

2.1 Gathering Data from the LMS

The three sample courses are available in Canvas, a commercially available LMS. Canvas has a well-documented application program interface (API). This API allows programmatic access to much of the data that resides in Canvas. This API is used to gather existing data from the sample courses.

A course may contain many different materials that are used for instruction. For our purposes, these various materials are referred to as *learning items*. Each learning item has its own characteristics such as title, due date, content, delivery method, and whether or not it is graded. The Canvas API is used to acquire details about individual learning items.

Some of the data can be used to identify relations between learning items. The primary relation available in the LMS is the temporal *precedes* relation. This relation expresses how learning items are related over time. The due date data extracted from Canvas is used to establish this temporal ordering. The *precedes* relation expresses that one learning item comes before another in time. For example, Activity 5 *precedes* Homework 3.

Another relation that can be identified directly from the data available in the LMS is the *includes* relation. This expresses the relation between a unit and a learning item. For example, Unit 3 *includes* Homework 7. A unit is a way of grouping learning items. This grouping may be based on a temporal factor such as a group of items that all happen in a given week. It may be based on topics: all items in a unit that cover the same topic. It may be based on an external resource such as a textbook: all items included in the unit refer to a specific chapter. These units are identified by the module tool in Canvas. The module tool provides a mechanism for instructors to organize learning items into units. The *includes* relation provides information about the existing organization of the course.

2.2 Engaging the Instructor

There are additional relations that cannot be directly acquired through the LMS. Some relations can be attained through graph transformations. In addition, **ENABLE** provides a web interface that allows the instructor to provide the expert knowledge required to establish some more meaningful relations. Including the instructor in this process improves the quality of the resulting course maps. Additionally, it was found that engaging the instructor in this process also provides benefits for the instructor. As the instructor interacts with the details of the course in new ways, insight is gained about the current organization of the course and an expanded view of the possibilities for change provided.

With input from the instructor about the topics that are the focus of the course, text analysis techniques are used on the content available in the LMS and provided by the instructor to establish the *occurs in* relations. This relation expresses that a topic *occurs in* a learning item. For example, publishing *occurs in* Activity 13. With these relations, topical connections between learning items can be identified.

The topical connections between learning items combined with the *precedes* relation provide valuable information to identify the *prerequisite* relation. The *prerequisite* relation expresses that there is value in doing one learning item before another. It may be that the following learning material is difficult to understand if the previous learning item has not yet been completed. The *precedes* relation must exist for a *prerequisite* relation to exist. By definition the item that is a prerequisite to another item must precede it. However, there are many cases when the *precedes* relation exists and there is no *prerequisite* relation. When two items have common topic associations it is more likely there is a *prerequisite* relation between them, although this is not always the case. Common topics is just an indicator of the greater possibility of the *prerequisite* relation.

ENABLE provides an interface for the instructor or other expert to identify *prerequisite* relations. For each learning item the interface provides a list of all the preceding learning items and identifies whether they include common topics. The instructor can select any of the preceding learning items to be included as a prerequisite.

3 Building the Course Map

A graph of the learning items and the relations is used to create the course map. A *course map* is a graph, $\mathcal{M} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of learning items, topic and unit nodes, and $\mathcal{E}$ is the set of *precedes*, *topically precedes*, *prerequisite*, *occurs in* and *includes* edges (relations). Then the *class map* is $\mathcal{C} = (\mathcal{L}, \mathcal{R})$, where $\mathcal{L} \subset \mathcal{N}$ is the set of learning item nodes and $\mathcal{R} \subset \mathcal{E}$ is the set of edges. A *path* of length k is any legal sequence $P = \{n_1, n_2, \ldots, n_{k+1}\}$, where $n_i \in \mathcal{L}$ and $\neg\exists i, j \ni n_j$ *prerequisite* n_i and $i < j$. Let P^S be the set of nodes in the path P.

Notice that the limiting relation in the path is only the *prerequisite* relation. This creates possibilities for a wide variety of paths through the learning items. This kind of flexibility does not usually exist in the traditional class setting. In an online educational setting the temporal limitations of the *precedes* relations need not be enforced. Making this shift to allow varied paths through the course material changes how both the educator and the student view a course. Tools described here provide the mechanisms to support such variation.

3.1 Course Map of Current Organization

In the existing course both the *precedes* and *includes* relations can be extracted from the LMS. This provides enough data to produce the course map shown in Figure 2. In this very simple course map we begin to see the additional information that can be expressed when the learning materials are presented in a graphical way. The location of each learning item on the x-axis is relative to the day the learning item is due. This provides a way to visually see how much temporal distance is between learning items. For example, CA5 *precedes* HW4 and HW4 *precedes* CA6 but there is a clear difference in the amount of time that separates them. CA5 and HW4 are due at about the same time while there is a period of multiple days between when HW4 and CA6 are due. This relative positioning is one way that information can be visually encoded.

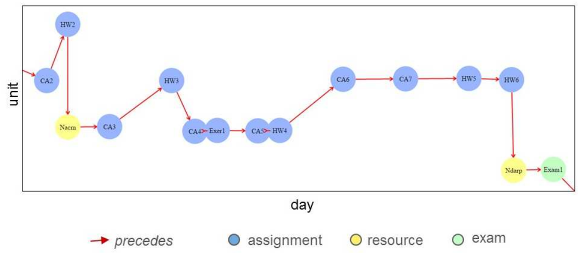

Figure 2: Section of a Course Map of the Sample CS0 Course with Learning Item Nodes,
xpressed in the Vertical Location of the

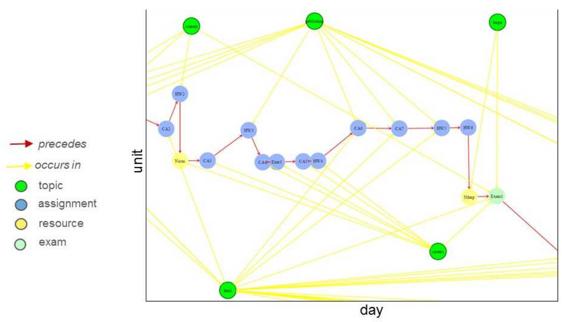

Figure 3: Section of Course Map Showing the Topical *Occurs In* Relations in Bipartite Graph.

3.2 Adding Topical Relations

The course map can be expanded to include the topically based *occurs in* relation. The first approach represents the topics as nodes in the graph. These nodes have a unique color to distinguish them from the learning item nodes. Then a bipartite graph is created with directed edges from the set of topic nodes to the set of learning item nodes.

As it is possible for each learning item to have an *occurs in* relation with multiple topics, this adds more than N edges, where N is the number of learning items in the course. To reduce the number of edge crossings introduced by the addition of this large number of edges, the topic nodes are located above and below the learning item nodes. This allows the edges representing the *precedes* relations to be clearly visible. It also maintains the relative temporal spacing between the learning items (see Figure 3).

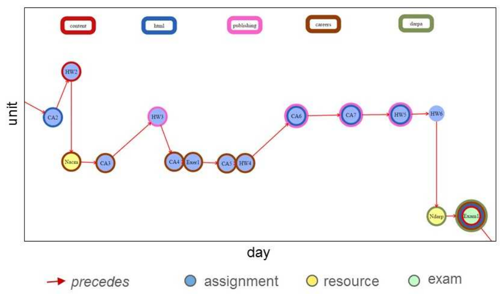

Figure 4: Section of Course Map Showing the Topical *Occurs In* Relations Drawn as Colored Rings.

It is possible to represent the topical relations without adding edges. One way this is done is by adding colored rings to the learning item nodes. Each topic is assigned a specific color. A legend is added to the course map that shows which color goes with which topic. Whenever there is an *occurs in* relation between a topic and a learning item, a ring of the specific color associated with the topic is added to the learning item. Multiple rings can be added to a single learning item node. Adding multiple rings increases the size of the learning item node. This visually expresses the quantity of topics occurring in a single learning item (see Figure 4).

3.3 Adding Prerequisite Relations

The *precedes* relation is the predominant relation expressed in the LMS. It is a weak relation tying learning items together only by the order in which they come in a specific version of a course. In some cases this order is critical, and it is important to do one learning item before the other as the second item is dependent on knowledge acquired from the previous item. In other cases the items are unrelated and the temporal relation is a restraint. To provide more variation in possible organizations and therefore meet the educational needs of a wider group of individuals it is valuable to remove such restraints.

We define a restraint as an unnecessary constraint between two items. Thus, restraints are removed in order to open up more possibilities for the relations between learning items. When removing restraints it is important to maintain both the semantics of the course representation and the integrity of the graph structure.

The *prerequisite* relation expresses a *precedes* relation that has a specific benefit. This *prerequisite* relation identifies the educational value in doing one learning item before another.

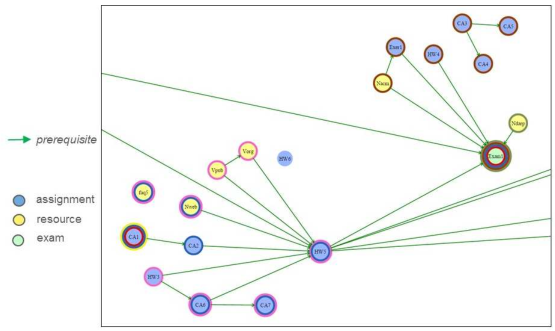

Figure 5: Section of Course Map Showing Prerequisite Relations with the Topical *Occurs In* Relations Drawn as Colored Rings and the Learning Items Grouped by the Unit Based *includes* Relation.

Once the prerequisites have been identified the course map is redrawn using these *prerequisite* relations to connect the learning items (see Figure 5). In this particular course map topically based *occurs in* relations are represented with colored rings. The learning items are grouped based on the unit *includes* relation. No *precedes* relations are included. They have been replaced by the *prerequisite* relations.

The *prerequisite* relations help students identify the meaningful connections between learning items. These connections are often familiar to the educator but mechanisms to impart that awareness to students are missing in the current presentation of learning materials in the LMS.

3.4 Manipulating the Course Map

Now that the temporal restraints of the course have been removed, the opportunities for restructuring the course have increased. The **ENABLE** system can display a graph based on the *occurs in*, *includes*, and *prerequisite* relations without the chronological restraints.

The course map display is designed in such a way that the nodes can be moved about. As a node is moved, any connecting edges move with it. Keeping these connections intact during moving preserves the integrity of the graph structure and maintains the relations between learning items. This manual manipulation of the course map provides a way to see the course with many different layouts. The learning items can be organized by topic, by exam, by learning item type, by prerequisite chains, etc. This provides the instructor, and potentially students, the opportunity to explore and discover possible paths through the course material.

4 Facilitating Change

This work has shown that the data available in the LMS can be used to generate a graphical representation of a course. By gathering some additional information from an instructor or other course expert, the graphical representation can be expanded to provide additional information and more meaningful relations. This graphical course map provides a new way to see the course materials and how they are related to each other. This more informative presentation is valuable. However, the graphical course map is more powerful than the information it can contain. It has the potential to be a mechanism for fundamental change in how education is delivered.

Traditionally courses have run over a specific time frame and are delivered in the same order and timing to all students regardless of ability and circumstance. This works for many students and educators but not all. To expand the reach of education to people outside the traditional classroom, the attachment to a linear, temporally based approach to education can be shifted. Currently some see the online course as a way to include people with severe disability in the world of education [10], but many online courses unnecessarily bring the limitations of time and order with them. To decrease this limitation and expand the educational opportunities provided on the Web, a fundamental change needs to occur. Educators and students will need to view the linear, time oriented presentation of a course as an unnecessary limitation and expand their thinking to include alternative approaches.

The graphical course map is a possible mechanism to support this change in perspective. Simply presenting the learning items in a graphical way allows the educator to see the course differently. Secondly, restraints can be reduced by removing the connections that are temporally based and adding the prerequisite relations that identify the connections that are so beneficial to the educational process. With these meaningful connections in place the course map can be manipulated to illustrate new ways to organize the material. Since it maintains the prerequisite connections while the learning items are manipulated, the important relations stay intact ensuring the educational integrity of the course materials remains.

There will be a need for additional tools to support a more flexible delivery of course materials such as individual due dates and varying completion time frames. Expanding our thinking and supporting fundamental change can lead to a more universal design that has the potential to meet the needs of a broader spectrum of people and circumstances.

5 Testing the Possibilities

Allowing students to move through the learning items in different orders introduces an entirely different component to a course. To explore the theoretic impact of such a change artificial student agents, probability models, and calibration techniques were implemented.

5.1 Artificial Student Agents

An artificial student agent is an algorithm that is provided with a set of learning item options (nodes it may spend time on), feedback of performance (i.e., grades or scores), and which returns an action in an iterative manner. In order to analyze the relation between the learning item organization and student performances, detailed student models (based on actual student data) were developed. These artificial student agents can traverse the course map in a variety of node sequences. Different limitations were placed on individual student agents based on their own characteristics. Specifically the student agents were assigned a value for each of four characteristics: intelligence, work ethic, background, and distractibility. However, the only limitation imposed by the course map is prerequisite relations. A learning item $l \in \mathcal{L}$ can only be attempted after all the prerequisite learning items have been visited. The agent determines how well they do on each learning item they visit including the option to apply no effort and receive a zero score. After all the prerequisite learning items have been visited, the learning item $l \in \mathcal{L}$ can be visited.

A trace of the order the learning items are visited is recorded as each agent moves through the learning items. These learning agents demonstrated a large variety in the order in which the learning items can be attempted. For more information about these learning agents refer to the author's previous work in [7].

For use with the estimation method described in Section 5.2, learning agents were created that implement the concept of mastery. A learning item, $l \in \mathcal{L}$, also has an associated difficulty level, $d(l) \in \Re^{\geq 0}$, where $\Re^{\geq 0}$ is the non-negative real numbers. The class graph imposes traversal constraints on a student; namely, every prerequisite of a node l must be mastered to an acceptable level before node l can be mastered.

At each time step, the agent specifies how much time of the total alloted is to be spent on each accessible learning item; this constitutes and *action*. Agents may implement different learning tactics and their respective learning performance traces may then be compared. For example, an agent that spends equal time on newly available nodes or equal time on all previous nodes if there are no new nodes, is specified as:

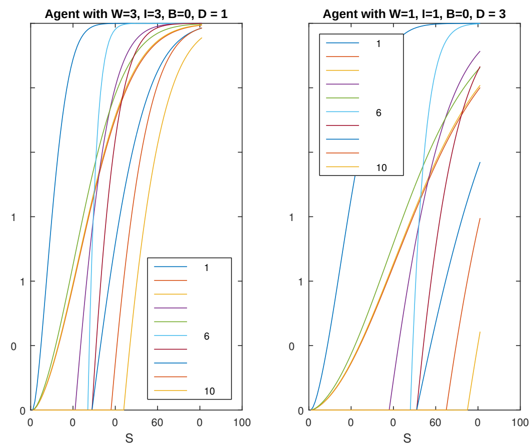

Figure 6: Learning Curves for Agents with Different Abilities.

```
On input: open (accessible) nodes
Local: visited nodes
On output: relative percent of time on all nodes

new_nodes <-- open and not visited

if no new_nodes
  if open_nodes not empty
    action <-- open/|open|
  end
else
  action <-- new_nodes/|new_nodes|
end
```

To demonstrate the performance of an artificial student agent on the class map, $\mathcal{C}$ (shown in Figure 1), suppose the agent has characteristics $W = 3$, $I = 3$, $B = 0$, and $D = 1$. Then, a learning curve plot for $Agent_1$ on the 10-node class graph is shown in Figure 6 (left), while an $Agent_2$ with $W = 1$, $I = 1$, $B = 0$ and $D = 3$ is shown on the right side of the Figure 6 (right). $Agent_1$ has achieved almost perfect mastery of all ten learning items by step 80, whereas $Agent_2$ has only mastered a few items in the same time.

Note that learning curves are also a function of the learning tactics of the agent. Suppose that $Agent_1$ modifies its approach so as to focus on individual items until they are mastered before moving on to the next available item. The resulting learning curve is shown in Figure 7 which illustrates that items are mastered sequentially and takes longer to learn all ten items than the equal time strategy. [Note that this may also provide evidence that a

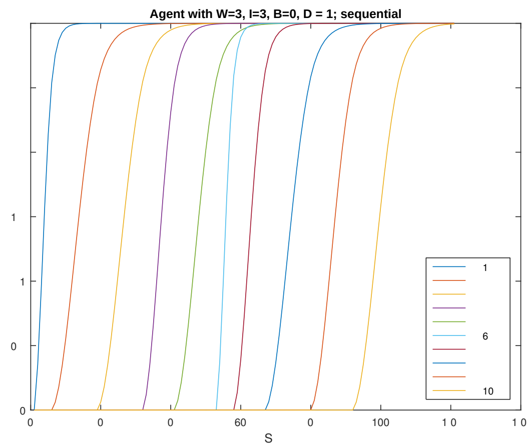

Figure 7: Learning Curves for Agent with Two Different Learning Tactics.

linear organization of the course material slows learning!]

5.2 A Learning Model

In [7], we defined the notion of *mastery* of a learning item as a random variable ranging from 0 to 1 and demonstrated the use of a linear learning model combined with a Kalman Filter to obtain an optimal estimate of student mastery of the learning items in a course graph based on combining the model prediction with a measurement (i.e., a grade) correction.

In the present work we propose a more refined nonlinear student learning model which includes a parameter – the *learning coefficient* – and compare three ways to estimate it: (1) direct inverse, (2) iterative least squares, and (3) the Extended Kalman Filter. This is called either *model parameter calibration* or *parameter estimation*.

The estimation method is based on the use of a class graph which describes the organization of the learning material, a set of artificial student agents with an associated learning model, and a mechanism for the class graph traversal. A wide variety of user models have been proposed for interactive learning environments; e.g., see [2, 3, 4, 9]. We have opted to use a more basic and general model of learning as described in [15]:

$$x_i^{t+1} = M - (M - x_i^t)e^{-k_i s_i^t} + \epsilon \tag{1}$$

where x_i^t is the mastery level of learning item i at time t, M is the maximal mastery level (which we set to 3 in our experiments), k_i is the learning coefficient for the student on learning item i, s_i^t is the cumulative time spent on learning item i, and $\epsilon \sim \mathcal{N}(0, \sigma^2)$. σ^2 is the variance in the learning model process. The learning coefficient k_i is a function of the agent and the learning item:

$$k_i = \frac{\frac{W+I+B}{D}}{\alpha_i} \quad (2)$$

where α_i is the difficulty of learning item i (and is in the range [0,100] in our experiments).

5.2.1 Learning Model Parameter Calibration

Given the **ENABLE** framework it is possible to use the information from a student's interaction with the class map to estimate the particular learning coefficient for each learning item. This allows active modification of the graph traversal in order to facilitate learning by the student. It also makes it possible to estimate the difficulty of each learning item (by using ratios of learning coefficients) so that the instructor is better informed about the nature of the presentation of the learning items.

5.2.2 Direct Inverse Method

Given the learning model in Equation 1, then for every step at a learning item in which learning takes place (i.e., $x_i^{t+1} > x_i^t$), and which had time allocated to the item (i.e., $s_i^{t+1} > s_i^t$), then k_i can be found as:

$$k_i = \frac{-ln(\frac{-(x_i^{t+1}-M)}{(M-x_i^t)})}{s_i^t} \quad (3)$$

Since there is noise in the learning process, the following algorithm is applied:

```
for every learning item i
  for every step t that meets conditions
    calculate k_i,t
  end
  k_i estimate is median of k_i,t
end
```

For $\sigma^2 = 0.001$, with $Agent_1$ and class map $\mathcal{C}$, the learning curves for one trial are shown in Figure 8. The actual learning coefficients are [0.1250, 0.0204, 0.0208, 0.0408, 0.0247,

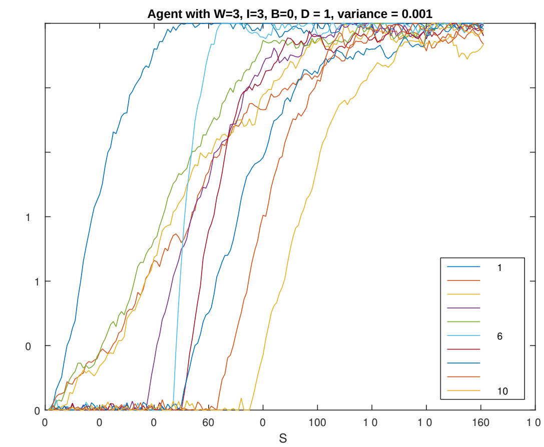

Figure 8: Learning Curves for Trial (with noise).

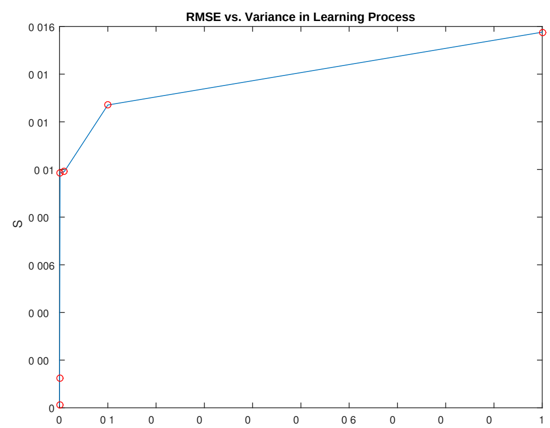

Figure 9: RMSE Values for Learning Coefficient Estimate for the Inverse Method.

0.1333, 0.0465, 0.0217, 0.0250, 0.0208] and the estimates found using the inverse method are: [0.1257, 0.0205, 0.0210, 0.0409, 0.0252, 0.1340, 0.0466, 0.0217, 0.0249, 0.0208] and the RMSE is 0.0123. Figure 9 shows the RMSE on this for σ^2 ranging from 10^{-5} to 1, with 10 trial samples per variance value.

5.2.3 Least Squares Method

Least squares is a standard method for the determination of a best solution to an over-constrained problem (see [5] for an introduction). We follow here the method described in

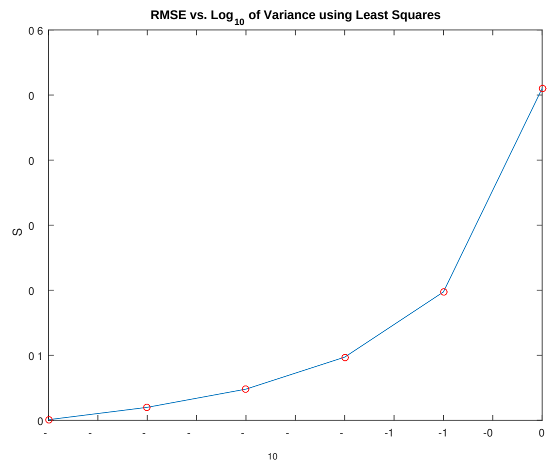

Figure 10: RMSE Values for Learning Coefficient Estimate for the Least Squares Method.

[12]. The least squares estimate is arrived at by iterating:

$$k_i^{t+1} = k_i^t + (J^T J)^{-1} J^T (Y - V|_{k_i^t}) \tag{4}$$

where k_i^t is the estimate of the actual learning coefficient k^* at step t, J is the Jacobian of the learning process model, Y is the observed mastery values from the trace of a student traversal of the class graph, and $V|_{k_i^t}$ is the predicted mastery values for a student traversal of the class graph using the current learning coefficient estimate. Note that for the full graph, k^* is a vector, and $|k^*| = n$, where n is the number of nodes in the class graph.

Since the process model is:

$$f(k, s, x) = M - (M - x)e^{-ks} \tag{5}$$

then the Jacobian is:

$$J = \frac{\partial f}{\partial k} = s(M - x)e^{-ks} \tag{6}$$

The least squares iteration is continued until convergence criteria are satisfied. Figure 10 shows the RMSE values achieved by the least squares method.

5.2.4 Extended Kalman Filter Method

The Kalman Filter is a state estimation technique that seeks to optimally combine a process model prediction of the state with a measurement of the state where both have an associated uncertainty. Here we use the Extended Kalman Filter since it applies to nonlinear models (see [14] for a detailed introduction to Kalman Filter methods). We apply the algorithm to each non-zero mastery level update in the student's traversal of the class graph as described earlier. The algorithm is then repeated until convergence:

1. $\bar{k}_t = a(k_{t-1})$
2. $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q_t$
3. $K_t = \bar{\Sigma}_t B_t^T (B_t \bar{\Sigma}_t B_t^T + R_t)^{-1}$
4. $k_t = \bar{k}_t + K_t(Z_t - b(\bar{k}_t))$
5. $\Sigma_t = (I - K_t B_t)\bar{\Sigma}_t$

where k_t is the learning coefficient estimate at time t, a is the process model for how k evolves, A_t is the Jacobian of the process model for k, Q_t is the covariance for the process model for k, b is the measurement model (in our case, this is the learning update function), B_t is the Jacobian of the measurement model, K_t is the Kalman Filter gain, R_t is the covariance of the measurement model, and Z_t is the observed student performance. In particular, these variables are:

$$a(k) = k$$
$$A_t(k) = 1$$
$$b(k, s, x) = M - (M - x)e^{-ks}$$
$$B(k, s, x) = s(M - x)e^{-ks}$$

and Q_t and R_t are assigned specific variances. Applying this method to the student learning traces yields the learning coefficient estimates shown in Figure 11. Figure 12 compares the three methods directly. As can be seen, the inverse method works best over all and the least squares method performs the worst, while the EKF works slightly better in lower noise than the inverse method.

5.3 Probability Models

A probability model provides a way to make predictions. Predictions can be used to inform students and educators about possible outcomes. With the data available in the existing course probability models can be generated. How accurate are they? Can data from a linear, temporal based course be used to predict outcomes for a course that allows different paths through the learning material?

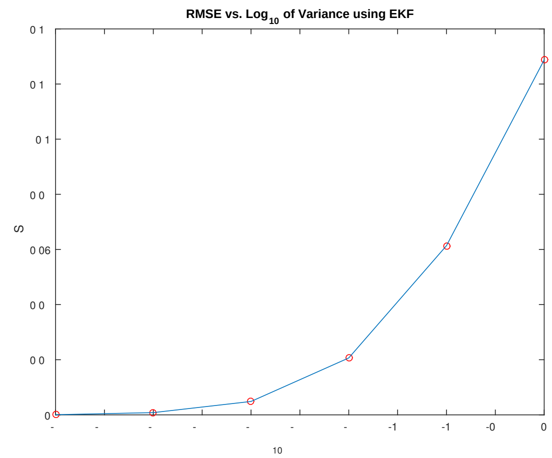

Figure 11: RMSE Values for Learning Coefficient Estimate for the Extended Kalman Filter Method.

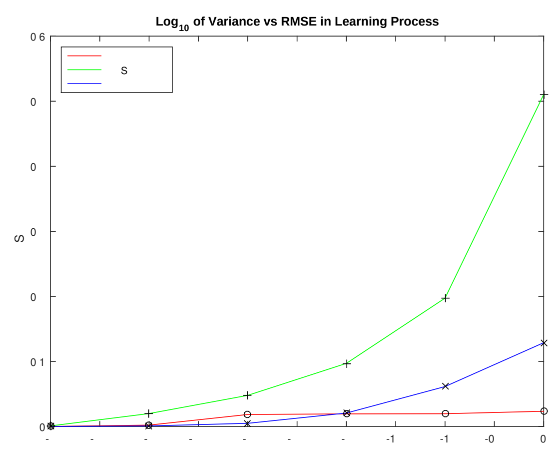

Figure 12: RMSE Values for Learning Coefficient Estimate for the Three Methods.

To answer this question several probability models that predict grades on learning items were created. In this section we discuss three of those models. The models are trained using the existing score data. Many of the models are able to predict individual scores with over 70% accuracy. They can also be sampled to produce data that has a distribution similar to the original data. These models can be restricted to only prerequisite relations in the existing data and still produce results with similar accuracy. This demonstrates that existing data from a linear, temporal based course can be used to predict outcomes for a course that allows more variation.

5.3.1 The Bayesian Network

ENABLE produces a representation of a course that includes learning items and a variety of relations between those items. Using this information a Bayesian network can be created. This network has nodes that represent the learning items and edges corresponding to the relations. To build the conditional probability tables (CPTs) scores are separated into five buckets for grades A, B, C, D/F, and zero. When analyzing scores for learning items the score of zero has special significance. It is a score that shows up on most learning items but it regularly falls outside the normal curve. The zero score most often reflects that the student did not participate in the learning item. This reflects something very different than a low score. A low score indicates that a student participated but did poorly. The existence of the zero score is an anomaly in data that is otherwise often considered a Gaussian distribution. To increase the accuracy of the predictive model this score is considered in a distinctive way.

The other anomaly occurs at the maximum points possible. Many learning items have a higher than expected value at this point. This arises from the fact that higher scores are not possible. This upper limit will disrupt the normal extension of a Gaussian distribution and congregate an increased number of scores at this maximum attainable score.

The number of parents in the CPT table vary depending on the specific model. The precedes one variation has exactly one parent for each node except the first node which will have no parent. This produces a consistent size throughout the CPTs. The first learning item will have a $1xB$ table where B is the number of buckets. For all the learning items after the first, each table will be BxB in size.

The precedes three variation has exactly three parents for each node except the first three nodes which will have zero, one, and two parents respectively. This produces a consistent size throughout the CPTs. For all the learning items after the first three, each table will be B^3xB in size.

The prerequisite relations produce tables with varying dimensions. Each table will be B wide. It is the number of inputs that will vary. There will be one parent for each prerequisite. As the number of parents increases the size of the table increases exponentially. The table will have B^P rows where B is the number of buckets and P is the number of parents. This table will have $B^{(P+1)}$ entries.

5.3.2 The Linear Model

The linear probability model is created using the linear function and a variance based on the error model. The error model considers the difference between the actual values and the result computed by the linear function. These differences are used as the variance. The linear function combined with this variance is used for both prediction and sampling. The linear model in **ENABLE** uses linear regression. Linear regression works as follows: Given a random sample

$$(Y_i, X_{i1}, X_{i2}, ..., X_{ip}) \tag{7}$$

where $i = 1, \ldots, n$ and p is the number of features, the relation between the observations Y_i and the independent variables X_{ij} is formulated as

$$Y_i = W_0 + W_1 X_{i1} + \cdots + W_p X_{ip} + \varepsilon_i \qquad i = 1, \ldots, n \tag{8}$$

In the above, the $W_j's$ are the regression coefficients and $\varepsilon_i = N(0, \sigma)$ is the standard error.

The predicted values corresponding to the above model are linear functions of W_j. One function is produced for each learning item and may consider the scores of the preceding learning items. The formula includes an initial value, W_0, and a term for each feature, $W_j(X_{ij})$. Although there may be dependencies between the learning items, this is not considered in the linear model.

5.3.3 The Mixed Model

The simple linear model does not represent zero scores very well and sometimes gives near zero probabilities for possible scores that simply have not been observed. For this reason the grade probability model is built using a mixture of three distributions:

1. Gaussian model that predicts the score, assuming the student completed the item. This is the same model as described in the previous section.

2. A model that predicts a non-completion (i.e., zero) grade. This distribution assigns a likelihood of 1 to a score of 0 and gives all other scores a likelihood of 0.

3. A uniform model that predicts the same probability for each score. The likelihood of each score is simply $1/number\_of\_possible\_scores$.

The likelihood of a given score is simply the weighted sum of the three component distributions.

$$l(s) = w_g * l_g(s) + w_z * l_z(s) + w_u * l_u(s) \tag{9}$$

where w_z and w_g are determined by a logistic regression model that estimates the probability a given student will complete an assignment. w_u is determined by hand and is set to 0.02 for all reported experiments. This value can be adjusted to give greater or less weight to the uniform distribution. The logistic regression model uses the same features as the linear model described in the previous section.

Logistic regression uses the logistic function which can take an input with any value from negative infinity to infinity and produce a value between zero and one which can be interpreted as a probability. The logistic function is defined as follows:

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}} \tag{10}$$

where t is a function of a linear combination of explanatory variables and expressed as:

$$t = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n \tag{11}$$

Now the logistic function can be written as:

$$F(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + ... + \beta_n x_n)}} \tag{12}$$

$F(x)$ is the probability of the dependent variable given a linear combination of explanatory variables, x. β_j are the regression coefficients.

This approach of combining distributions is applied to sensor measurements in probabilistic robotics [14]. Its use here is a novel application of that process. This combined model is included in the analysis. See Figures 13 through 16 for an example of the component distributions and the resulting mixed distribution.

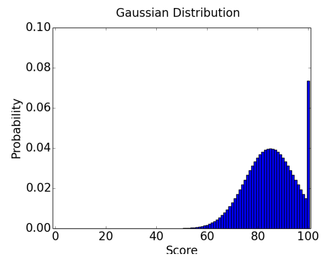

Figure 13: Histogram of Gaussian Probability Distribution.

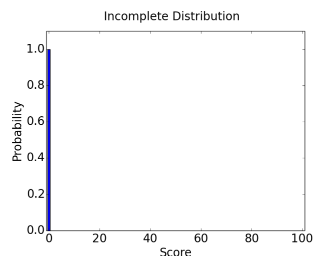

Figure 14: Histogram of Zero Probability Distribution.

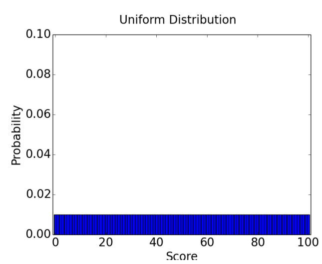

Figure 15: Histogram of Uniform Probability Distribution.

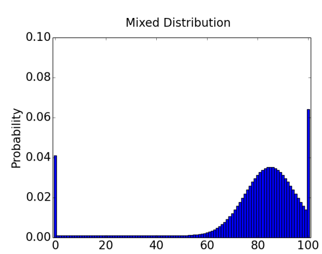

Figure 16: Histogram of Mixed Probability Distribution.

5.3.4 Comparing Score Prediction

A valuable piece of functionality that is produced by these models is the ability to predict scores. This ability can be used to make regular recommendations to students and projections for educators. The process for generating these scores is to use leave one out cross validation. This process uses the original actual data and separates it into a training set and a test set. The training set consists of the scores for all the learning items except the one that is currently being predicted. The test set are the scores that will be generated by the models for the learning item that is being predicted. In this case it is the single score left out of the training set. This is called leave one out cross validation. The scores in the training set are used to train the model. Then the trained model generates a set of scores for the single learning item being predicted.

These generated scores are compared to the original scores to identify how accurately the model predicted the score. The accuracy comparison is reported as a percentage. This is the percentage of times the score generated by the model is correct. Correctness is identified by identifying the grade category of the generated score. If the generated score belongs in the same grade category as the actual score, it is correct.

The scores are generated using the sampling process that minimizes the L1 error. Each score in the model's probability distribution is considered and the L1 error for it is computed. These computed L1 errors are compared and the score with the smallest L1 error is selected.

Table 1 shows a varying degree of accuracy in making these score predictions. The precedes one considers the score of the first preceding learning item, the precedes three considers the scores of the first three preceding items, and the prerequisites considers all the prerequisites for the specified learning item. Restricting the parents in the Bayesian network and the features in the linear and mixed linear methods to prerequisites only reduces the accuracy of the predictions by 2%-5%.

6 Conclusions

This work lays a foundation for the creation of general purpose graphical course mapping tools. It demonstrates the possibility of generating such a map using currently available data and algorithms. The manipulatable course map produced can support educators as they transform a synchronous, temporal based course presentation to one without the same

Table 1: Comparing Grade Accuracy Between Models.

Model Type	Dependencies	Grade Accuracy
Mixed linear	Precedes Three	77%
Mixed linear	Precedes One	75%
Mixed linear	Prerequisites	72%
CPT Grade Buckets	Precedes One	75%
CPT Grade Buckets	Precedes Three	73%
CPT Grade Buckets	Prerequisites	73%
Linear	Precedes Three	72%
Linear	Precedes One	72%
Linear	Prerequisites	67%

limiting temporal restraints. It allows the possibility of individual students moving through the course materials in a variety of orders and time frames.

Several automated student agents have been developed and used with learning models to consider how basing the ordering of learning materials on prerequisite relations might impact the learning process. This initial investigation found that there were many ways to navigate through the learning material of three sample courses. These paths were restricted only by the prerequisite relation.

Predictive models were produced and used to demonstrate that data from existing linear, temporal based courses could be used to train predictive models. These predictive models could be limited to prerequisite relations and still produce accuracy results that were just slightly less than when precedes restraints were included in the data. This provides a way to generate recommendation systems for students and educators using a more flexible delivery method.

Phase I of this work provides a solid foundation for the creation of graphical course mapping systems. For such a system to become widely useful, an interface is needed that incorporates the relations discovered and the recommendations available through the predictive models. Phase II is the next step of this work and includes (1) creating a rich graphical user interface that improves both the quality and quantity of student and teacher interaction with the learning material, and (2) conducting user testing at all stages of the system design, development, and testing to identify the usability and accessibility of the interface and make revisions based on the results. Such an interface could then be embedded in the LMS for student and faculty use on the Web or in mobile devices.

References

[1] I Elaine Allen and Jeff Seaman. Grade Change: Tracking Online Education in the United States, 2013. *Babson Survey Research Group and Quahog Research Group, LLC. Retrieved on*, 3(5):2014, 2014.

[2] P. Brusilovsky, S. Sosnovsky, and O. Shcherbinina. User Modeling in a Distributed E-Learning Architecture. In *User Modeling 2005. Lecture Notes in Artificial Intelligence, (Proceedings of 10th International User Modeling Conference)*, pages 24–29. Springer Verlag, 2005.

[3] C. Carmona, G. Castillo, and E. Mil'an. Designing a Dynamic Bayesian Network for Modeling Students' Learning Styles. In *Proceedings of the International Conference on Advanced Learning Technologies*, pages 346–350, Santander, Cantabria, Spain, July 2008. IEEE.

[4] C. Carmona and R. Conejo. A Learner Model in a Distributed Environment. In *Proceedings of the Third International Conference on Adaptive Hypermedia and Adaptive Web-based systems*, Eindhoven, The Netherlands, August 2004.

[5] S.C. Chapra and R.P. Canale. *Numerical Methods for Engineers*. McGraw Hill, Boston, MA, 2002.

[6] L. DuHadway and T. Henderson. Informing Change: Course Content Analysis and Organization. In *Proceedings of the Frontiers of Education Conference*, El Paso, TX, October 2015. IEEE.

[7] L. DuHadway and T. Henderson. Artificial Student Agents and Course Mastery Tracking. In *Proceedings International Conference on Agents and Artificial Intelligence*, Rome, Italy, February 2016.

[8] Catherine S Fichten, Vittoria Ferraro, Jennison V Asuncion, Caroline Chwojka, Maria Barile, Mai N Nguyen, Ryan Klomp, and Joan Wolforth. Disabilities and e-Learning Problems and Solutions: An Exploratory Study. *Journal of Educational Technology & Society*, 12(4):241–256, 2009.

[9] Nicola Henze and Wolfgang Nejdl. Student Modeling in an Active Learning Environment Using Bayesian Networks. In *In Proceedings of the Seventh International Conference on User Modeling, UM99*. Springer, 1999.

[10] Sang-Mook Lee. Universal Design in Online and Higher Education Through ICT. In *Proceedings of the 11th Web for All Conference*, W4A '14, pages 9:1–9:1, New York, NY, USA, 2014. ACM.

[11] Terry Müller. Persistence of Women in Online Degree-Completion Programs. *The International Review of Research in Open and Distributed Learning*, 9(2), 2008.

[12] M.R. Myers, A.B. Jorge, and D. Greg Walker. A Comparison of Extended Kalman Filter Approaches using Non-Linear Temperature and Ultrasound Time-of-Flight Measurement Models for Heating Source Localization of a Transient Heat Transfer Problem. In *Inverse Problems, Design and Optimization Symposium*, Joao Pessoa, Brazil, August 2010.

[13] D. Rover, Y. Astatke, S. Bakshi, and F. Vahid. An Online Revolution in Learning and Teaching. In *Proceedings of the Frontiers of Education Conference*, Oklahoma City, OK, October 2013. IEEE.

[14] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, 2005.

[15] W.I. Zangwill and P.B. Kantor. Toward a Theory of Continuous Improvement and the Learning Curve. *Management Science*, 44(7):910–920, July 1998.