Representation, Selection, and Scheduling of Training in a Lifelong Learning Context

Alan Carlin, Danielle Ward, Jared Freeman

Aptima, Inc.

Woburn, MA

acarlin@aptima.com, dward@aptima.com, freeman@aptima.com

ABSTRACT

Modern job training often takes place away from the classroom. Students' goals include meeting training requirements, augmenting professional knowledge, and reaching career objectives. However, training opportunities are often limited by a student's responsibilities and current tasking. Constraints also include the student's location and schedule. In this work, we report on (1) a framework to formalize the representation, selection, and scheduling of training in a lifelong learning context. (2) An AI-scheduling algorithm to best choose training courses, venues, and schedules for the student. First, the framework includes training needs, including job requirements and career objectives. Second, the framework includes training opportunities, including information about training type, training location, competencies trained, and course schedules (times and dates). Third, the framework includes information about the trainee, including the trainee's previous knowledge, skills, and experiences, and previous assessments. Once the framework is defined, sequential optimization algorithms can provide advice on training schedule. The algorithms consider hard constraints and soft constraints. Hard constraints include scheduling constraints and location constraints. Soft constraints include venue preferences, training environment preferences, and training type preferences. These constraints are balanced against student career objectives and goals. The selection is sequential, meaning that the algorithm "looks ahead" to the future, it is able to select a course because it is a prerequisite to another, later course. It is our hope that by carefully formalizing lifelong learning as an AI-scheduling problem, students will be able to achieve career goals and seize opportunities that balance against their existing responsibilities.

ABOUT THE AUTHORS

Dr. Alan S. Carlin is a Senior Research Engineer at Aptima, Inc. His interests focus on problems of artificial intelligence and machine learning. These include problems of decision making under uncertainty, communication between members of a team, meta-reasoning for decision-makers, and causal modeling. His publications include works on Decentralized Partially Observable Markov Decision Processes, communication under uncertainty, and distributed meta-reasoning in uncertain environments.

Dr. Danielle (Dumond) Ward is a Senior Research Engineer in Aptima, Inc.'s Analytics, Modeling & Simulation Division. Her background lies in machine learning, decision making, control engineering, probabilistic modeling, and robotics. Dr. Ward's interests include using mathematical models to predict and control how different agents within a system will behave and interact. She leads the Organizational Research and Optimization capability at Aptima.

Dr. Jared Freeman is Chief Scientist at Aptima, Inc. As a Cognitive Scientist, Dr. Freeman investigates human problem solving and decision making in real-world settings, and defines methods of monitoring and managing these processes using modeling and training applications. Dr. Freeman has served as Principal Investigator (P.I.) for projects to analyze and develop systems to train and assess imagery analysts, information operations specialists, technicians for intelligence equipment, Air Force and Navy air crew, Marines and Soldiers in operations in the Global War on Terror, and interagency personnel engaged in Maritime Domain Awareness missions.

Representation, Selection, and Scheduling of Training in a Lifelong Learning Context

Alan Carlin, Danielle Ward, Jared Freeman

Aptima, Inc.

Woburn, MA

acarlin@aptima.com, dward@aptima.com, freeman@aptima.com

Introduction

Recent literature has shown that career success is not just a function of educational credentials, but rather of a set of skills (Hanuchek et al. 2013). The resulting differential between skills, when holding educational credentials constant, has been referred to as the skills premium (Autor, 2014). With an emphasis on continuing skills rather than an initial education, the model of receiving an initial 2 or 4 year education, without follow-up, is largely outdated. A lifelong learning paradigm has taken its place. Concepts such as Continuing Education and On the Job training have started to become the norm, and supplemental materials such as Massive Open Online Courses (MOOCs) have become more popular, with a diverse set of courses available (Schneider, 2013).

Lifelong learners are in a quandary, however. To set learning goals well, they must understand the skill requirements of jobs and professions. To acquire those skills, they must survey and select from a plethora of educational materials. Finally, lifelong learners may have time and location constraints, due to both work schedules and other responsibilities that complicate the logistics of learning. These are significant challenges. Learners rarely have the information and the capability to construct both near and far term plans for lifelong learning.

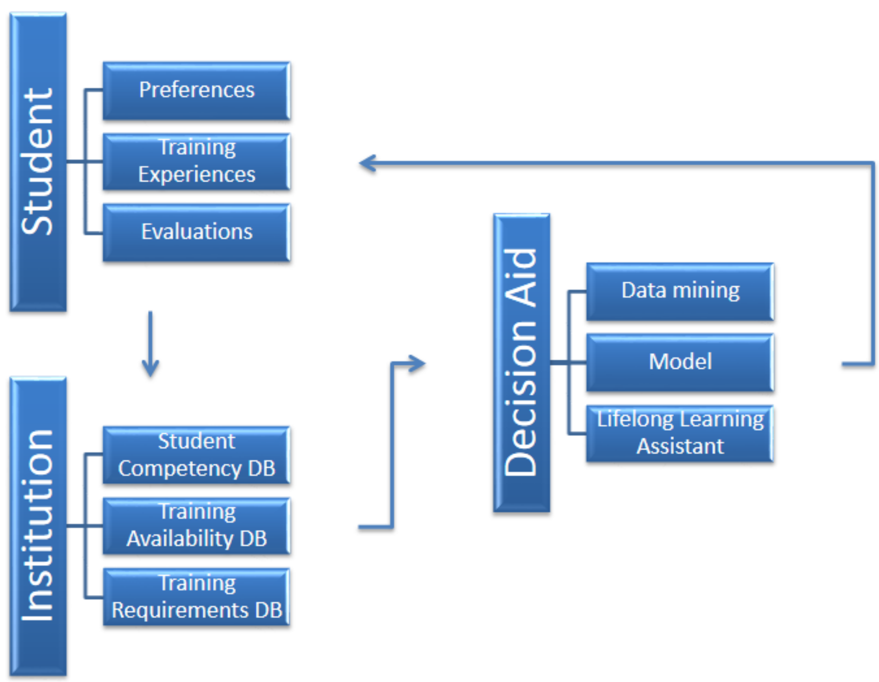

Figure 1. Components of the lifelong learning process discussed in this paper.

Figure 1 shows various components of lifelong learning, and the interactions among the components. The student is engaged in the process of lifelong learning. Each evaluation populates databases that administrators (of MOOCs, or at colleges, or at On the Job training sites) access. Educational Data Mining (EDM) tools, are used to make sense of the data, by interpreting the fundamental skills taught in the education process, as well as the effect of courses on those skills, and the fidelity of measurement tests (e.g., pre-tests, post-tests, course materials, letter grades) in measuring student progress. Another resource available to administrators is the availability of venues for future training. In contrast, the student typically holds little of this data, and it is often beyond their capability to reason over it completely and well.

In the future, software should assist this process, and we refer to this software as the Lifelong Learning Assistant in Figure 1. A data mining component and a model drive the Assistant. This paper focuses the Decision Aid containing those three components. We also argue that this component of the solution should be personalized. Different learners, with different goals and different levels of availability, at different levels of skill, require different learning plans. In this paper, we outline several facets, and the relationship among them, that can aid lifelong learning. These facets include:

- A model of skill acquisition that can be used by the lifelong learner.
- An optimization algorithm that can help the lifelong learner to plan.
- A data mining tool that can help educators better understand and adapt their content to the needs of the students.

The rest of the paper proceeds as follows. First, we describe our model of the lifelong learning process. This model is domain agnostic. Next, we describe how the model can be parameterized; that is, we discuss how the variables in the model can be assigned to fit a given learning domain. Finally, we discuss how a data mining tool can help with this assignment.

Model of Learning Process

The process of lifelong learning can be represented, at least in part, as a *sequential optimization problem under uncertainty*. Optimization problems are a class of problems involving the selection of the optimal configuration of a set of variables, given a set of constraints and a rule for assigning each configuration a value. In the case of learning, a set of variables is created to represent training options, and the value of each variable configuration represents the expected impact of that training configuration on the learner.

A specific class of models representing sequential optimization under uncertainty is the POMDP model (Smallwood & Sondik, 1973). The POMDP (Partially Observable Markov Decision Process) framework is sequential in nature, that is, optimization takes place over a *sequence* of time steps, and at each time step a single decision is made. Furthermore, at each time step the configuration of the environment contains some *uncertainty*. (We say that the environment is partially observable). The goal is to optimize the sequence of decisions. Two aspects of the POMDP model make it suitable for lifelong learning. First, selection is both adaptive and personalized. POMDP solutions continuously adjust their assessment of the learner, and select the next component of the curriculum based on the learner results as they are obtained. Second, the model assigns higher value to solutions that plan ahead. Rather than just selecting the next course or experience, a POMDP solution is a plan, which includes a plan for future steps of the curriculum as well as the current one. This allows the plan to training in skills that are not of immediate benefit, but which will support future training.

The POMDP is a statistical Bayesian approach to decision/theoretic planning under uncertainty (Smallwood & Sondik, 1973). A POMDP extends the classic Markov Decision Process (Puterman, 1994), and is used in Operations Research and Computer Science domains such as assisted living (Hoey, Poupart, von Bertoldi, Craig, Boutilier, & Mihailidis, 2010), patient management (Hauskrecht & Fraser, 1998), and spoken dialog systems (Williams, 2005). In prior work, we have developed the model for intelligent training systems and intelligent tutoring systems (Levchuk, Shebilske, & Freeman, in press; Freeman, Stacy, MacMillan, Carlin, & Levchuk, 2009). POMDPs are used to solve problems in which there are observable variables (e.g., measurements from the system such as course grades) and non-observable variables (e.g., current underlying skills of the learner). Our particular approach to mathematically modeling the learner is to combine multiple sources of observable information and hypotheses about non-observable information to form an optimized plan called the POMDP policy that transitions the learner through a sequence of scenes in a scenario. Thus, for any given learner for which we have collected data, our mathematical modeling approach can determine events or exercises as training content. As an example, we provide a glimpse of the mathematics that underlie this modeling approach. A Partially Observable Markov Decision Process (POMDP) can be defined with the tuple: M = $(S, A, P, \Omega, O, R, \gamma)$ such that :

- S is a set of states.

  - In learning, state represents the level of skill with respect to various Knowledges, Skills, and Experiences (KSE's) of the learner.
- A is a set of actions.
  - Actions represent training options (courses, training simulations, etc.).
- P is the state transition probability table: $P\ (s'|s,a)$,
  - This represents the anticipated effectiveness of training.
- Ω is the set of observations.
  - This represents any measure or assessment that can be stored in an LMS (Learning Management System) or LRS (Learning Record Store)
- O is the observation probability function: $O(o\ \in\ \Omega|s', a, s')$,
  - This represents the probability of the measure, given underlying skill level. Fundamentally, the observation function allows the model to quantify measurement error.
- R is the reward function: $R(s,a)$, the immediate reward for being in state s and taking action a.
  - Reward is achieved by meeting training objectives.
- $\gamma \in (0,1)$ is a discount factor between 0 and 1.
  - This term is included in the model for mathematical completeness. It can be safely ignored in context of this paper.

The figure below shows the concept of state in the model visually. The left shows that learner state is represented as a point in an n-dimensional space, where n is the number of skills to be learned. The figure shows a 2-dimensional space, so 2 skills are applicable in the simple figure. The point represents the expertise of the learner on the 2 skills. In the POMDP model, state is not known, rather there is measurement with uncertainty. This is shown by the dotted circle around the state, the instructor is not aware of the exact location of the point, only that the learner lies in the area of the circle. (Note: The uncertainty in the figure is somewhat simplified for visual representation. In fact, algorithms perform Bayesian reasoning over the transition and observation probability tables to derive its beliefs about learner state, and we compute a derived probability for each state). Our model of action effects is analogous to Vygogsky's (1978) concept of the Zone of Proximal Development (ZPD), which defines the training experiences that a student in a given state can master in a well-defined training environment. On the right of the figure, learning actions (courses, training events, etc.) are shown as squares. Each action targets a particular learner level of expertise. If the learner state lies outside the actions square (thus, outside of the ZPD), the action is unlikely to improve the learner. The goal of the model is to select actions that will provide a path to expertise, moving the learner from the lower left corner of expertise to the upper right, as the figure.

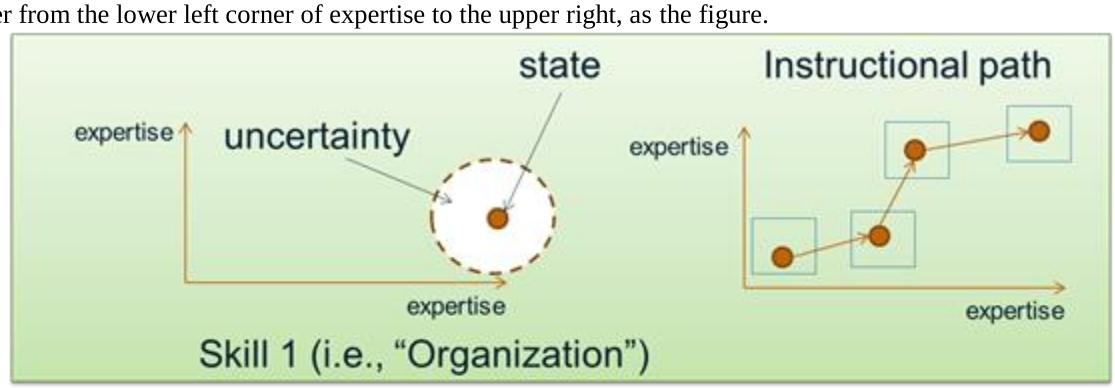

Figure 2. (left) State of a POMDP is partially observable. We can estimate it (dotted line) based on measures such as course grades, but usually it can not be deduced fully. (right): Even given the uncertainty, we can plan a path to expertise.

The **state** space is made of applicable skills of the learner (due to customer constraints we do not identify a particular domain for this paper, rather we keep the discussion domain agnostic). For each construct, or competency numbered i we create a set S_i consisting of the number of possible levels for the competency (e.g., 1 = "novice", 3 = "intermediate", 5 = "expert"), then we create the state space $S = S_1 \times \ldots \times S_n$ where each S_i represents a number of possible trainee levels on competency. A POMDP solver tracks the distribution of possible states that the trainee may be in. This is formally referred to as a *belief state*, $b \in P(S)$.

The set of *actions* is the available training content or modules accessible to the instructor. Each training module will be tagged for its difficulty level in each of the competencies. This will be used to help define the *state transition probability* table *P* (shown as the Transition Model in Figure 3). For example, difficult learning modules will have a positive effect on advanced trainees, whereas easy learning modules will have a positive effect on novice trainees. The state transition module will be determined a priori and adjusted by learning the transition probabilities as training data becomes available.

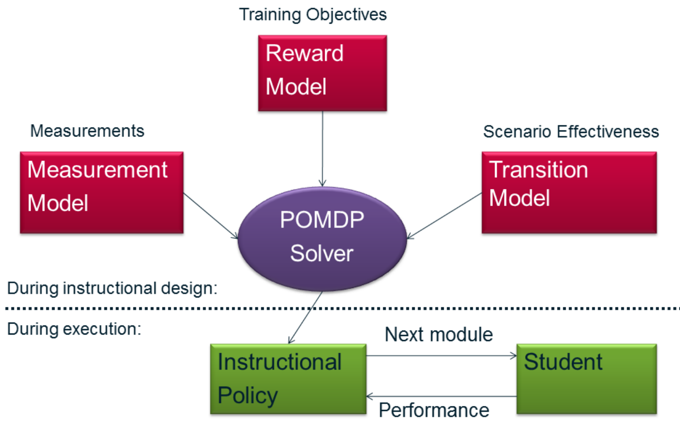

Figure 3. POMDP framework. Models of transition (predicted course effect), measurement (predicted fidelity of course grades), and rewards (learning goals) are inputs into the model. The output is a "policy" that helps the student select a course or module.

After each course, an assessment or course grade is typically received, defining *the observation probability function*. The probability function specifies the accuracy of assessment. Define the set of possible observations *Ω*. The observation can be the whole course grade, or where applicable, it can represent measures taken within a course. For example, at the level of a quiz the set *Ω* is the set of possible quiz scores. As a second example, at the level of a question, the set *Ω* may be correct or incorrect. Define the observation function as a probability function. $(o \in O \mid s \in S, a \in A, s' \in S)$, specifying the chance of a given outcome the previous a trainee state and learning item. Through use of this probability table, learner state can be inferred.
The purpose of the instructional model is to select actions at each step that maximize trainee capabilities. This will be enforced through the *reward function*. Note that this function rewards the instructional model for obtaining the training objectives, and not the trainee. The instructional model will receive higher reward for moving trainees into states of high competency. As an initial implementation, we will reward each skill equally, but in future implementations subject matter experts may determine that some skills should be rewarded more than others.

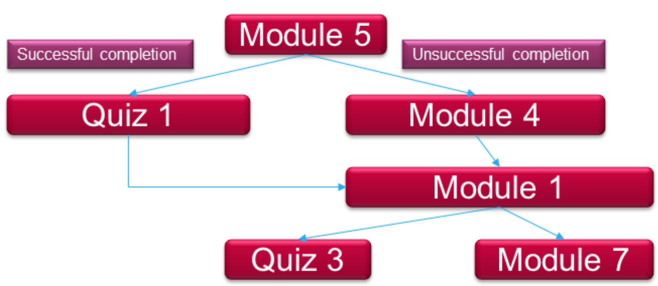

Figure 4. A simple example of a POMDP policy within the assistant depicted in Figure 1. Blocks of instruction can be at the granularity of modules within a course (as shown in the figure), or alternatively blocks can represent the courses themselves.

The goal of a POMDP planner is to provide a plan for instruction. This is a mapping from each possible history of learner measurements to the next module of the curriculum. A simple example of a POMDP policy is shown in the Figure 4. This example policy starts at the top of the tree and proceeds downwards. Each arrow represents a possible observation. The trainee is presented with Learning Module 5. If the trainee completes it successfully,

Quiz 1 is given. If not, Module 4 is invoked. As opposed to this simple hand-drawn policy, automatically generated policies which we will develop may have hundreds of branches (observations) for each node. Our approach does not have an opinion on the granularity of the modules selected. In Figure 4, the granularity is at the level of modules and quizzes within a course, but in a lifelong learning context it may be more applicable for the boxes to represent the courses themselves, and the branches to represent course grades.
A POMDP solver tries to maximize the total reward:

$$E\left(\sum_{t=0}^{T} r_t\right)$$

accumulated over a horizon T of interest: where r_t is a reward at time t. If the horizon is infinite ($T=\infty$), it can be shown that the value of the policy is: $V_\pi(s)=\sum_{t\in S} R(s,\pi(s),t)+V_\pi(t)\cdot P(s,\pi(s),t)$. That is, the value of a state is the direct reward for being in that state, plus the expected value of the next state. This expected value is found by multiplying the value of the next state by its probability, given the selected action. By the mathematical properties of the function, it can be shown that when it converges, the optimal policy is the optimal policy for the entire future of the student planned out, and not just the next action.

The corresponding optimal policy itself is: $\pi^*(s)=\arg\max_a\left[\sum_{t\in S} R(s,a,t)+V^*(t)\cdot P(s,a,t)\right]$.
More simply put, a policy denotes what action should be taken (e.g., which course should be collected next) given learner state or an instructor belief about trainee state. Belief about learner state is being updated with each measurement of performance on each course.

Finding Parameters of the POMDP Lifelong Learning Model

The above section provided the general POMDP model, but not its parameters. Several parameters must be filled in order to apply the domain of choice. The following questions must be answered.

- **State Space**: What are the competencies to be tracked?
- **Action Space**: What learning content is available?
- **Transition**: What is the predicted effect of the learning content on the competencies?
- **Observations**: What measures, records, etc., are made available during the learning process?
- **Reward**: What are the learner's goals?

<u>Parameterization of State Space:</u> Recall that learner "state" is a state of skill of the learner with respect to various competencies. The following are examples of what those competencies are, for the case of training.

- Mission Essential Competencies (MECs). These are defined as individual, team, or inter-team competencies required for successful mission completion under adverse conditions in a non-permissive environment.
- Knowledges and Skills. Knowledge is information or facts that can be accessed quickly under stress, while skills are compiled actions that can be carried out free of error under stress.
- Experiences. An experience is defined as a developmental event during training and/or career necessary to learn a Knowledge or Skill, or practice a MEC under operational conditions.

Alliger, Beard, Bennett, and Colegrove (2012) describes the intended purpose for Mission Essential Competencies[SM] and how the individual MEC components are developed and used.
Other components that can be used for the state space include:

- Identification of job tasks that a learner has executed or mastered over their career.
- Past Training Events and associated training performance data. This type of data can be stored in a government database, for government positions, in a school database, for academic purposes, or in a MOOC database.

We note that different domains will have different sources of information available, and thus will entail different methods of determining the competencies.

<u>Parameterization of Action Set:</u> This parameter does not differ between the SME-driven and data-driven cases. The action set is the set of learning options available. It can be populated from course rosters, training exercise assignments, etc.

<u>Parameterization of Transition Matrix:</u> SME's can generate parameters for the POMDP transition matrix. An accurate but infeasible method of doing this would be by filling out the Probability column in the following table:

Table 1. To fill out a transition matrix, each course must be tagged for the skills it trains, and the probability of progressing students through the skills. That is, a SME or modeler must fill out the second column and the final column of the table.

Course ID	Skill Id	Begin State	End State	Probability
Course 101	Skill A	Novice	Novice	20%
Course 101	Skill A	Novice	Intermediate	80%
Course 101	Skill B	Novice	Novice	100%
Course 101B	Skill A	Novice	Novice	100%
Course 101B	Skill B	Novice	Novice	10%
Course 101B	Skill B	Novice	Intermediate	90%
…				

In the example table, Course 101 trains novices at Skill A, while course 101B is more effective at training novices at Skill B.
However, it is impractical to ask a SME to fill out the numerous probabilities in this table, and the product might be inaccurate for a complex domain. A better mechanism, which we have used in several domains, has been to have the SME fill out an applicability and difficulty for each item instead.

Table 2. To make tagging data easier for the SME, the SME only needs to fill in the final two columns of this table. That is, how much the course applies to each skill, and its difficulty. From this information, Table 1 can be automatically constructed.

Course ID	Skill Id	Begin State	End State	Applicability	Difficulty
Course 101	Skill A	Novice	Novice	1 - Barely applicable	Easy
Course 101	Skill A	Novice	Intermediate	4 – Somewhat applicable	Easy
Course 101	Skill B	Novice	Novice	5 - Very applicable	Easy
Course 101B	Skill A	Novice	Novice	5 - Very applicable	Easy
Course 101B	Skill B	Novice	Novice	1 - Barely applicable	Easy
Course 101B	Skill B	Novice	Intermediate	5 - Very applicable	Easy
Course 102	Skill A	Intermediate	Expert	5 - Very applicable	Intermediate
…					

Then, through a rubric, the difficulty and applicability levels are converted to the transition probabilities. The only requirements for the rubric is that (1) the more applicable the course to the skill in its row, the more it trains that skill. (2) The more the scenario difficulty matches the Begin State (skill level) of the learner, the more effective it is at training that skill. This second rule corresponds to observing the Zone of Proximal Development (Vygotsky 1978).

<u>Parameterization of Observation Matrix</u>: Observations are the probability of getting course grades, given student skill level. There are two ways to do parameterize this. The first, in absence of any data, is to have a SME or instructor estimate a mean and standard deviation for the course grades (e.g., course mean = "B-", standard deviation = "1 letter grade"), and use the normal distribution equation as an observation function.

The second method, where available, is to use Item Response Theory (Lord, 1980). Our approach employs a variant of IRT that can represent multiple skills (Reckase, 2009), as well as Partial Credit. In Equation 1 below, different responses *c* represent different letter grades.

$$P_{\{c\}(\theta)} = \frac{e^{\sum_{j=0}^{c}(\theta - d_j)}}{\sum_{h=0}^{m} e^{\sum_{j=0}^{h}(\theta - d_j)}} \quad (1)$$

Where c represents the category of response, and d_j represents the difficulty of achieving that response. For ease of explanation, we can assume that difficulties are ordered so that for all $j < m$

$$d_j \leq d_{j+1} \tag{2}$$

To understand the intuition of Equation 1, first note, by inspection of the denominator's inner summation (it matches the form of the numerator), that the denominator is just a normalization term for the numerator. θ represents the student's skill level, and d_j represents the difficulty of achieving response j. Thus note that if $\theta \gg d_j$, the numerator is very large and the response is very likely. Conversely, if $\theta \ll d_j$, the numerator is very small, representing a very difficult item for the student, and the response is very unlikely to be achieved. When the difficulties are ordered so that $d_j \leq d_{j+1}$, as indicated in Equation 2, then each d_{j+1} is less likely to be achieved than d_j, and the equation can be viewed as the probability of achieving gradations of correctness such as incorrect, partially correct, almost correct, and fully correct.

As a technical note, in our model, student skill levels are vectors, θ_k^t, one for each KPA k. But for ease of explanation, Equation 1 assumes a single skill. To obtain a single scalar and fit equation 1, we multiply student skill level by item applicability, that is assuming n different skills:

$$\theta = \sum_{k=1}^{n} \theta_k^t a_k^t \tag{3}$$

Parameterization of Reward Function: Rewards reflect training objectives. Recall that in the model, a reward is assigned to each state and action (s,a), where s is the state and a is the learning action such as taking a course. We decompose this into the part of the reward that depends on the learner state, and the part of the reward that depends on the learning action.

The part of the reward that depends on state, is attained for achieving a Knowledge, Skill, or Experience. In other words, under a simple rubric, having a label of "Novice" on a skill is worth 0 points, attaining a state of "Intermediate" may be worth 1 point, and being at a state of "Expert" is worth 2 points. Skills can be weighted against each other, so that skills more important to the learner are weighted higher. There are many ways to customize this rubric for individual domains, we represent this by transmission of "Preferences" from the student to the model in Figure 1. Preferences can represent a student's own prioritization of venue (location), training environment (e.g., classroom or MOOC), or training type. These are soft constraints, or penalties, that are also represented by the reward function. Suppose a learner prefers courses at night over courses in the day, but absolutely needs these courses to be in the Virginia area. The last column of the following table is computed, based on these constraints and the contents of the other columns.

Course	Time	Location	Skills trained	Reward
Course A	1900-2000 MWF	Virginia	Skill A	10
Course B	1800-1930 Tu, Th	Nevada	Skill A	$-\infty$
Course C	11am-noon MWF	Honolulu	Skill A	$-\infty$
Course D	11am-noon MWF	Virginia	Skill A	4

The table reflects that classes at night are preferred, while classes outside of the Virginia area are simply not possible.

Data Mining Tool

Another component that can help build and use the model is data mining. The algorithm presented below, can data mine all the components of the POMDP model. The algorithm assumes that data for many students and lessons are available in an LMS. It solves for unknown latent variables, including θ the skill level of each student in Equation 1 from IRT), d (the difficulty of each course in Equation 1), trans (the transition probability of courses as depicted in Table 1, and a (which of the skills each course applies to). The algorithm uses Gibbs sampling, which is a technique to find the most probable assignments of many latent variables at once.

Algorithm Initialization:

- NUM_SKILLS := Number of skills in training domain

- MAX_DIFFICULTY := Highest possible level of difficulty
- Scores := All item scores for correctness
- For each training item i
    - For skill_num = 1 to NUM_SKILLS
        - a_init(I,skill_num) := 0
    - Rand1 := random number between 1 and NUM_SKILLS
    - a_init(i,rand1) := 1
    - rand2 := random number between 1 and MAX_DIFFICULTY
    - d_init(i) := rand2
    - for before_skill = 1 to NUM_SKILLS
        - for after_skill = 1 to NUM_SKILLS
            - if after_skill == before_skill
                - trans_init(before_skill, after_skill) = .95
            - else if after_skill == before_skill + 1
                - trans_init(before_skill, after_skill) = .05
            - End if
        - End for
    - End for
- End for
- For each student s
    - For each time-step t in the student training sequence
        - Theta_init(s,t) := 1 %initialize the interpretation that the student was a novice at all times
    - End for
- End for
- a := a_init; d := d_init; theta := theta_init; trans := trans_init;

<u>Sampling Algorithm</u>:

- NUM_ITERATIONS := number of iterations to run the algorithm
- BURN_IN := NUM_ITERATIONS/2; %first several iterations are just to stabilize the algorithm
- for iter := 1 to NUM_ITERATIONS
    - d := sample_difficulty(scores, a, theta);
    - a := sample_a(scores, theta, d);
    - theta = sample_theta(scores, a, d, trans);
    - trans = sample_trans(theta);
    - if iter > BURN_IN
        - d_sum += d;
        - a_sum += a;
        - theta_sum += theta;
        - trans_sum += trans;
    - End if
- num_samples = NUM_ITERATIONS – BURN_IN
- d_avg := d_sum/num_samples
- a_avg := a_sum/num_samples
- theta_avg := theta/num_samples;
- trans_avg = trans/num_samples;
- End for

As the algorithm shows, each variables is initialized to a default value. Then, each variable is sampled, given the values of the other variables. Although we have not delved into the internals of the sample_difficulty, sample_a, sample_theta, and sample_trans algorithms, the idea is fairly straightforward: For each unknown variable in sample_difficulty, sample_a, and sample_theta, we solve for the probability of each individual value. Then we sample from this probability distribution to generate the new value of the variable, for the next iteration. The advantage of taking the new value by sampling, rather than just assigning the most probable value to the variable, is that sampling better explores the possible space of variable settings.

sample_trans is a slightly different procedure, in that transition probabilities can be determined by counting the number of times the student transitions from one state to another. However, it is a well-known property that the procedure could fail or the overall algorithm can converge to extreme solutions if there are no samples of a given type. Therefore, we use a Dirichlet hyper-prior which acts as a pre-count (Doshi-Velez 2009).

Conclusion

In this paper we have discussed a model of instruction that can be used for lifelong learning. The model includes student learning objectives, student state which tracks those objectives, and meta-data on learning options including the effectiveness of each learning option on training, and the accuracy of grades and evaluations for each learning option. We also discuss an algorithm to help data-mine educational databases to fill out this model.
From this model, a learning plan can be constructed. Although this approach can be quite powerful, it does entail infrastructure hurdles to be overcome especially from learning institutions. Primary among these is thateducational data should be made available to data mining tools, without loss of privacy, so that course effectiveness can be evaluated and course recommendations can be accurately, efficiently delivered to students.

References

1. Alliger, G. M., Beard, R., Bennett Jr, W., Colegrove, C. M., & Garrity, M. (2007). Understanding mission essential competencies as a work analysis method. GROUP FOR ORGANIZATIONAL EFFECTIVENESS (GOE) ALBANY NY.
2. Autor, D.H. "Skills, education, and the rise of earnings inequality among the" other 99 percent"." (2014).
3. Doshi-Velez, F. (2009). The infinite partially observable Markov decision process. In Advances in neural information processing systems (pp. 477-485).
4. Freeman, J., Stacy, W., MacMillan, J., Carlin, A., & Levchuk, G. (2009, July). Capturing and building adaptive expertise in virtual worlds. *Proceedings of Human Computer Interaction International 2009*, 19-24. San Diego, CA.
5. Hanushek, E.A., Schwerdt, G., S.Wiederhold, L.Woessmann, *Returns to Skills Around the World: Evidence from PIAAC* (NBER Working Paper 19762, Cambridge, MA, 2013).
6. Hauskrecht, M. and Fraser , H. (1998). Modeling Treatment of Ischemic Heart Disease with Partially Observable Markov Decision Processes. *In Proceedings of American Medical Informatics Association Annual symposium on Computer Applications in Health Care, Orlando, Florida, pp. 538-542.*
7. Hoey, J., Poupart, P., von Bertoldi, A., Craig, T., Boutilier, C., & Mihailidis, A. (2010, May). Automated handwashing assistance for persons with dementia using video and a Partially Observable Markov Decision Process. *Computer Vision and Image Understanding (CVIU), 114*, 5.
8. Levchuk, G., Shebilske, W., and Freeman, J. (2012). A model-driven instructional strategy: The benchmarked experiential system for training (BEST). In P. J. Durlach & A. M. Lesgold (Eds.) *Adaptive Technologies for Training and Education* (pp. 303-317). Cambridge, UK: Cambridge University Press.
9. Lord, F. M. (1980). Applications of item response theory to practical testing problems. Routledge.
10. Puterman, M. L. (1994). Markov Decision Processes. J.
11. Reckase, M. (2009). *Multidimensional item response theory* (Vol. 150). New York: Springer.
12. Schneider, E. (2013, June). Welcome to the moocspace: a proposed theory and taxonomy for massive open online courses. In Proceedings of the Workshops at the 16th International Conference on Artificial Intelligence in Education (Vol. 1009, pp. 2-9).
13. Smallwood, R. D. & Sondik, E. J. (1973). The optimal control of partially observable Markov processes over a finite horizon. *Operations Research, 21*, 1071-1088.
14. Vygotsky, L. (1978). Interaction between learning and development. Readings on the development of children, 23(3), 34-41.
15. Williams, J. D., & Young, S. (2005). Scaling up POMDPs for Dialog Management: The``Summary POMDP" Method. In *Automatic Speech Recognition and Understanding*, *2005 IEEE Workshop on* (pp. 177-182). IEEE.