Learning Curves for Problems with Multiple Knowledge Components

Brett van de Sande
Pearson Eduction
brett.vandesande@pearson.com

ABSTRACT

Learning curves have proven to be a useful tool for understanding how a student learns a given skill as they progress through a curriculum. A learning curve for a given Knowledge Component (KC) is a plot of some measure of competence as a function of the number of opportunities the student has had to apply that KC. Consider the case where each problem-solving step is recorded by, for instance, by an intelligent tutoring system. In this case, one normally assigns a unique KC to each problem-solving step and the construction of the associated learning curves is straightforward. On the other hand, many online homework systems only evaluate the student's final answer to a problem. In that case, the student has generally applied a number of KCs to find the answer and their performance on the problem is some composite of their mastery of all of the requisite KCs. In this paper, we propose a simple method for generating learning curves for multiple-KC problems that is independent of any particular theory of learning. In the case where there is only one KC per problem, the method reduces to the ordinary learning curves. We demonstrate this method using a set of artificially generated student data.

Author Keywords
Learning Curves, Knowledge Components

ACM Classification Keywords
I.2.6 Learning: Knowledge acquisition

INTRODUCTION

The increased use of online homework systems and intelligent tutor systems (ITS) means that ever-increasing amounts of student log data is available for analysis. This data can be used to answer two important questions: what skills are students learning and how quickly are they learning them? To be more precise, we can equate skills with Knowledge components (KCs): small bits of information needed to solve a problem [11, 3]. KCs generally have some sort of pre-requisite relations: For example, you cannot apply the area of a circle formula $A = \pi r^2$ unless you first know the definition of "radius of a circle." However, aside from prerequisites, a KC can, by definition, be mastered independently from other KCs. This definition assumes that KCs are *context independent*. That is, the student's ability to apply that KC correctly or quickly does not depend on the particular problem the student is solving or the other KCs needed to solve that problem.

Since KCs are *defined* to have these properties, then it remains to be seen whether, and in what cases, they are a useful description of skill acquisition. One way to determine how well the KC picture is working is to examine the associated learning curves. If the curves are smooth, increasing/decreasing monotonically (depending on the measure of competence), and independent of context, then the KC picture is working.

Learning curves are a plot of some measure of mastery of a skill as a function of the number of opportunities that the student has had to apply that skill. Possible measures of mastery include:

- number of errors made before correctly applying the KC,
- time taken to correctly apply a KC,
- "assistance score," number of errors plus number of requests for help before completing a step, and
- "correctness", whether the student applied the KC correctly without any preceding errors or requests for help.

In the following, we will use "correctness" as our measure of competence for a given skill.

In a typical Intelligent Tutoring System (ITS), the student enters each problem-solving step into the tutor system. It is natural, in that case, to associate one KC with each student input and it is relatively straightforward to construct the associated learning curves. However, many online homework systems only require the student to enter their final answer to a problems into the system. In this case, a single input is the entire problem and it is natural to associate multiple KCs to each student input.

If multiple KCs are associated with a single input, then the construction of learning curves is more difficult. If the student gets the problem wrong, which KC is responsible? This is sometimes called the "assignment of blame problem" [7,

Table 1. List of definitions and quantities

k, l, m**:** label representing a KC.

t, u, v**:** label representing opportunity number for some KC.

p**:** label representing an exercise.

s**:** the student.

$P_{t,k}$ is a model parameter representing the probability that a student will apply KC k correctly on opportunity t. $P_{t,k} \in [0, 1]$.

$\xi_{s,p}$ is the model-given probability that student s will get problem p correct.

$C_{t,k}$ is the number of students in the dataset who correctly applied KC k on opportunity t.

$I(\boldsymbol{t}, \boldsymbol{k})$ is the number of students who got a an exercise containing KCs $\boldsymbol{k} = \{k_1, k_2, \ldots\}$ incorrect where $\boldsymbol{t} = (t_1, t_2, \ldots)$ is a vector of corresponding opportunities. This exercise represents opportunity t_a for the student to apply KC k_a.

$\mathcal{T}_{s,p}$ is the set of KC, opportunity pairs such that problem p is opportunity t for student s to apply KC k.

6, 5]. In the following, a simple method is proposed which addresses the assignment of blame problem while making a minimum of theoretical assumptions, allowing one to construct learning curves for exercises with multiple KCs. Our strategy is to introduce a model where every point on each learning curve is identified as a model parameter. These model parameters, and their associated errors, are then determined by a maximum likelihood fit to student log data. In the case of a single KC per problem/step, this reduces to the usual learning curves.

LEARNING CURVE MODEL

A number of studies have addressed the multiple-KC problem in the context of some model of learning, such as Bayesian Knowledge Tracing or Performance Factor Analysis [2, 4]. In the present work, our goal is simply to construct learning curves using a minimum number of model assumptions. Note that conventional learning curves themselves make two major assumptions:

1. They average over students. This corresponds to a model that does not have any student-specific parameters.

2. They ignore the problem context. This corresponds to a model that does not have any problem-specific parameters.

In fact, the construction of a learning curve is equivalent to fitting the student log data to a model containing a parameter representing each KC and step. In other words, if I define $P_{t,k}$ as the probability that a student will correctly apply KC k at opportunity t, and determine $P_{t,k}$ by fitting to the student log data, then plotting of $P_{t,k}$ versus t is a learning curve for KC k.

This gives us a way forward in the multiple-KC case. We define a model having parameters $\{P_{t,k}\}$. The associated log-likelihood is

$$\log(\mathcal{L}) = \sum_{s,p \in \mathcal{C}_s} \log(\xi_{s,p}) + \sum_{s,p \in \mathcal{I}_s} \log(1 - \xi_{s,p}) \qquad (1)$$

where s is the student, p is the problem, $\mathcal{C}_s$ is the set of problems s got correct, and $\mathcal{I}_s$ is the set of problems s got incorrect. Also, $\xi_{s,p}$ is the model-given probability that student s will get problem p correct.

We will assume that the student must apply *all* of the associated KCs to solve a given exercise correctly. This is sometimes called a "conjunctive model" and is a good approach for typical K-12 math exercises [8]. This means that the total probability of success is the product of the KC probabilities:

$$\xi_{s,p} = \prod_{t,k \in \mathcal{T}_{s,p}} P_{t,k} \qquad (2)$$

where $\mathcal{T}_{s,p}$ is the set of KCs and opportunities such that problem p is opportunity t for student s to apply KC k.

To construct $\mathcal{T}_{s,p}$, one needs a list of KCs associated with each exercise p, sometimes referred to as the "Q-matrix" [10]. In this discussion, we will assume that the Q-matrix is known, perhaps determined by the problem author or a domain expert.

Numerical Calculation

The likelihood given by Eqn. (1) is rather inconvenient for large numerical calculations. Instead, we will introduce variables that aggregate over student and exercise. Define $C_{t,k}$ to be the number of students in the dataset who correctly applied KC k on opportunity t. Likewise, define $I(\boldsymbol{t}, \boldsymbol{k})$ to be the number of students who got a an exercise containing KCs $\boldsymbol{k} = \{k_1, k_2, \ldots\}$ incorrect where $\boldsymbol{t}$ is a vector of associated opportunities. This exercise represents opportunity t_a for the student to apply KC k_a. Then, the log-likelihood can be written as

$$\log(\mathcal{L}) = \sum_{t,k} C_{t,k} \log(P_{t,k}) + \sum_{\boldsymbol{t},\boldsymbol{k}} I(\boldsymbol{t}, \boldsymbol{k}) \log(1 - \Gamma(\boldsymbol{t}, \boldsymbol{k})) \qquad (3)$$

where $\Gamma(\boldsymbol{t}, \boldsymbol{k})$ is the probability that a student with opportunity vector $\boldsymbol{t}$ will have success on a problem containing KCs $\boldsymbol{k} = \{k_1, k_2, \ldots\}$. Following Eqn. (2), $\Gamma(\boldsymbol{t}, \boldsymbol{k})$ is a product over the associated probabilities:

$$\Gamma(\boldsymbol{t}, \boldsymbol{k}) = \prod_a P_{t_a, k_a} \,. \qquad (4)$$

Note that the first term of Eqn. (3) has a much simpler form than the second term. This is due to our use of a conjunctive model. If a student gets an exercise "correct" then we know without ambiguity that they applied all of the associated KCs correctly. However, if they get a problem wrong, then it is not clear which KC is to blame and the associated probabilities must be considered jointly.

Let $\{\widehat{P}_{t,k}\}$ be the model parameters at the maximum likelihood point. $\{\widehat{P}_{t,k}\}$ can be found numerically by maximizing the log-likelihood, Eqn. (3) subject to the constraints

Table 2. KC content of the artificial homework set. Students completed the first eight problems in the given order and the remaining problems in random order; they completed between 15 and 20 problems total.

1	2	3	4	5	6	7	8	9	10
A	A	A	A	B	B	B	B	A	B
11	12	13	14	15	16	17	18	19	20
A	B	AB	AB	AB	AB	AB	AB	AB	AB

$0 \leq P_{t,k} \leq 1$. For convenience, the *Mathematica* function **FindMaximum**, was used to calculate the maximum of $\log(\mathcal{L})$. However, any optimization algorithm that enforces constraints and uses information about the gradient of the function should work as well.

Error analysis

It is important to calculate the standard errors associated with the model parameters. Unlike the single KC per problem case, the model parameters may be strongly correlated and the errors can have unexpected values. In addition, the error analysis can elucidate any cases where the model parameter cannot be determined from the data (we will discuss this further in the conclusion).

Before finding the errors, we need to examine the the maxiumum likelihood point and identify any parameters that lie on the boundaries $\widehat{P}_{t,k} = 0$ or 1. The likelihood function $\mathcal{L}$ is not stationary in these parameters at the maximum likelihood point, so the error analysis cannot be applied to them; they should be not be included in the Hessian matrix below, Eqn (5). In practice, this should not a significant issue, since $\widehat{P}_{t,k} = 0$ or 1 typically occurs when there are just a few student problem-solving instances for a given t and k.

For a maximum likelihood fit, the standard errors associated with the model parameters can determined using the following procedure [1, 9]. First, we find the Hessian matrix associated with $P_{t,k} = \widehat{P}_{t,k}$. The matrix elements of the Hessian are given by

$$\left.\frac{\partial^2 \log(\mathcal{L})}{\partial P_{t,k} \partial P_{u,l}}\right|_{P_{v,m}=\widehat{P}_{v,m}} = -\frac{1}{\widehat{P}_{t,k}\widehat{P}_{u,l}} \sum_{\boldsymbol{t},\boldsymbol{k}} \left.\frac{I(\boldsymbol{t},\boldsymbol{k})\,\Gamma(\boldsymbol{t},\boldsymbol{k})}{(1-\Gamma(\boldsymbol{t},\boldsymbol{k}))^2}\right|_{P_{v,m}=\widehat{P}_{v,m}} . \quad (5)$$

To find the standard error associated with each of the model parameters $\widehat{P}_{t,k}$, we invert the negative of the Hessian matrix and take the square root of the diagonal elements. If this process fails (the Hessian matrix is singular), it is a signal that some of the model parameters cannot be uniquely determined from the given log data. Similarly, if the Hessian matrix is nearly singular, then the associated standard errors will be very large. This will single out any model parameters that cannot be determined from the data.

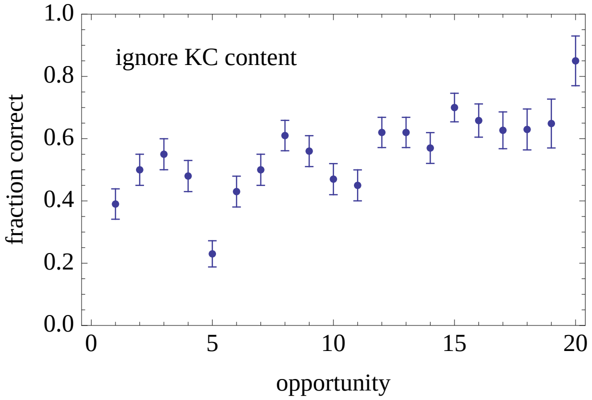

Figure 1. Learning curve for the artificial homework set where we assume each problem has the same single KC. Note the jump after opportunity 4 due to the fact that the first four and second four problems have different KCs.

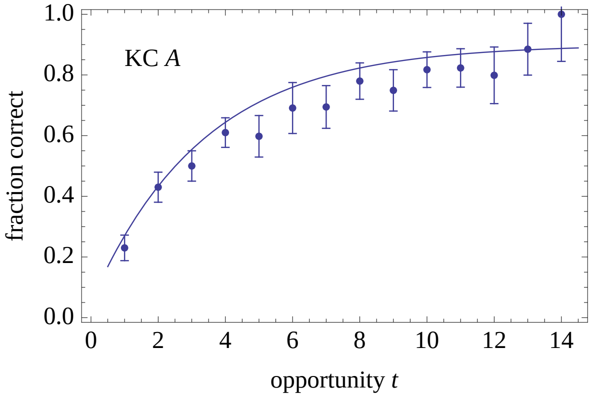

Figure 2. Learning curve for KC A. The solid line is the model used to generate the student data and the points with error bars represent the learning curve determined from the student data using our procedure. Note that the error bars for the last few opportunities are larger, due to student attrition.

APPLICATION TO STUDENT DATA

To illustrate how this model works, we will generate an artificial student performance dataset. Consider a homework assignment of 20 problems that exercise two KCs, A and B as detailed in Table 2. We assume that students progress through the first 8 problems in the given order, but solve the remaining 12 problems in random order, completing between 15 and 20 problems. We assume that student mastery for the KCs is given by the functions $P_{t,A} = 0.9 - 0.85\mathrm{e}^{-0.3t}$ and $P_{t,B} = 0.85 - 0.45\mathrm{e}^{-0.1t}$; see Figures 2 and 3. We use this model to generate a set of outcomes, $\mathcal{C}_s$, $\mathcal{I}_s$, and $\mathcal{T}_{s,p}$, for 100 students.

If we ignore the KC content of the problems, we can plot a naïve learning curve for this student data; See Fig. 1. We see a discontinuity at $t = 4$ due to the change in actual KC content of the problems. The last problems are more difficult, since they involve two skills and so the student performance on them is suppressed.

Next, we use our procedure to generate learning curves and associated errors for this dataset. The results are plotted in

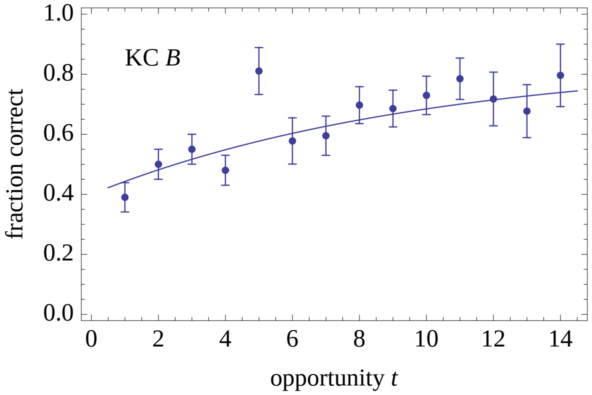

Figure 3. Learning curve for KC B. The high value at $t = 5$ is a statistical fluctuation: as we iincrease the number of students, the model parameters will converge to the solid line.

Figs. 2 and 3. As expected, they agree well with the model used to generate the student data. This shows that our method is working. Note that the error bars can vary considerably from point to point.

CONCLUSION

The primary goal of the approach developed here is to plot learning curves for cases where there are problems (or problem steps) involving multiple KCs. In practice, we find our method to be numerically robust (no problems with local maxima).

However, there is one case where it may fail: if there is a KC that always appears along with another KC for several problems and all the students in the dataset solve nearly the same ordered sequence of problems, then there is no way distinguish between the two KCs for one or more value of t. This will result in a Hessian matrix that is not positive-definite and the matrix inversion will fail. We believe that this situation will rarely arise in practice, since most datasets involve students in multiple courses, and students are generally not forced to solve problems in a specific order.

In this work, we focused on a "conjunctive model" for combining KCs, as this is likely the most appropriate model for typical math and science exercises. Although the basic strategy we present here could be applied to other models (disjunctive, compensatory) for combining KCs, the details of the associated numerical calculation would look rather different.

Obviously, the next step is to apply this approach to real student data. This would require a set of exercises that have been tagged with multiple KCs, where the mix of KCs vary significantly from exercise to exercise. In addition, the student activity would have to fairly heterogeneous, with different students taking different paths through the exercises.

ACKNOWLEDGMENTS

This work was supported by Pearson education. I would like to thank Ilya Golden for reviewing the manuscript.

REFERENCES

1. Edwards, A. W. F. *Likelihood*. Johns Hopkins University Press, 1992.
2. Gong, Y., Beck, J., and Heffernan, N. Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting Procedures. In *Intelligent Tutoring Systems*, V. Aleven, J. Kay, and J. Mostow, Eds., vol. 6094 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2010, 35–44.
3. Koedinger, K. R., Corbett, A. T., and Perfetti, C. The Knowledge-Learning-Instruction Framework: Bridging the Science-Practice Chasm to Enhance Robust Student Learning. *Cognitive Sci. 36*, 5 (2012), 757–798.
4. Koedinger, K. R., Pavlik, P. I., Stamper, J., Nixon, T., and Ritter, S. Avoiding Problem Selection Thrashing with Conjunctive Knowledge Tracing. In *Proceedings of the 3 rd International Conference on Educational Data Mining* (2010), 91–100.
5. Nwaigwe, A., and Koedinger, K. R. The Simple Location Heuristic is Better at Predicting Students' Changes in Error Rate Over Time Compared to the Simple Temporal Heuristic. In *Proceedings of the 4th International Conference on Educational Data Mining* (Eindhoven, the Netherlands, 2011), 71–80.
6. Nwaigwe, A., Koedinger, K. R., Vanlehn, K., Hausmann, R., and Weinstein, A. Exploring alternative methods for error attribution in learning curves analysis in intelligent tutoring systems. *Frontiers in Artificial Intelligence and Applications 158* (2007), 246.
7. Ohlsson, S. Towards Intelligent Tutoring Systems that Teach Knowledge Rather than Skills: Five Research Questions. In *New Directions in Educational Technology*, E. Scanlon and T. O'Shea, Eds., no. 96 in Nato ASI Subseries F. Springer Berlin Heidelberg, Berlin, Heidelberg, 1992.
8. Pardos, Z. A., Beck, J. E., Ruiz, C., and Heffernan, N. T. The Composition Effect: Conjunctive or Compensatory? An Analysis of Multi-Skill Math Questions in ITS. In *Educational Data Mining 2008: 1st International Conference on Educational Data Mining, Proceedings*, UNC-Charlotte, Computer Science Dept. (Montreal, Canada, June 2008), 147–156.
9. Pawitan, Y. *In All Likelihood: Statistical Modelling and Inference Using Likelihood*. Oxford University Press, June 2001.
10. Tatsuoka, K. K. Rule Space: An Approach for Dealing with Misconceptions Based on Item Response Theory. *Journal of Educational Measurement 20*, 4 (Dec. 1983), 345–354.
11. VanLehn, K. The Behavior of Tutoring Systems. *Int. J. Artif. Intell. Ed. 16*, 3 (Jan. 2006), 227–265.