Discovery of Student Strategies using Hidden Markov Model Clustering

Benjamin Shih
Machine Learning Department
Carnegie-Mellon University
Pittsburgh, PA 15213
shih@cmu.edu

Kenneth R. Koedinger
HCI Institute
Carnegie-Mellon University
Pittsburgh, PA 15213
koedinger@cmu.edu

Richard Scheines
Department of Philosophy
Carnegie-Mellon University
Pittsburgh, PA 15213
scheines@cmu.edu

Abstract

Students interacting with educational software generate data on their use of software assistance and on the correctness of their answers. This data comes in the form of a time series, with each interaction as a separate data point. This data poses a number of unique issues. In educational research, results should be interpretable by domain experts, which strongly biases learning towards simpler models. Educational data also has a temporal dimension that is generally not fully utilized. Finally, when educational data is analyzed using machine learning techniques, the algorithm is generally off-the-shelf with little consideration for the unique properties of educational data. We focus on the problem of analyzing student interactions with software tutors. Our objective is to discover different strategies that students employ and to use those strategies to predict learning outcomes. For this, we utilize hidden Markov model (HMM) clustering. Unlike some other approaches, HMMs incorporate the time dimension into the model. By learning many HMMs rather than just one, the result will include smaller, more interpretable models. Finally, as part of this process, we can examine different model selection criteria with respect to the models predictions of student learning outcomes. This allows further insight into the properties of model selection criteria on educational data sets, beyond the usual cross-validation or test analysis. We discover that the algorithm is effective across multiple measures and that the adjusted-$\mathbf{R}^2$ is an effective model selection metric.

1 Introduction

Educational software is an increasingly importanlisi

attempt. By not incorporating the entirety of the data, particularly the ordering of actions, such analyses fail to realize the data's full potential.

In a computer tutoring system, the log data may be treated as a time-series with variable intervals of observation. If the performance on each task is conditionally independent given the student, i.e. solving a math problem step does not require successful solutions to prior steps, then each task can be treated as a separate sequence of observations. Thus, each step or sequence can be considered a segment of a time-series. For example, if a student requests help at the beginning of the step and then attempts solutions until they solve the step, that is likely a different strategy than if a student attempts to solve the step and, upon failing, requests help.

In this paper, the concept of student strategies is instantiated by hidden Markov models (HMMs). HMMs are graphical models which treat observed data as an ordered sequences of symbols. HMMs will be discussed in more detail in the Background section; however, the primary observation is, by learning many different HMMs from educational data, each HMM can be treated as a model of a different student strategy. Prior work in educational data mining has largely focused on learning single, complicated models that describe all possible student behaviors. The advantages of collections of HMMs are four-fold: they have disjoint observations, the observations are ordered, they are much easier to interpret, and they provide extremely accurate predictions. Further, the algorithm we propose offers several advantages over standard HMM clustering algorithms: it has adaptive parameters, biases strongly towards smaller models, and can incorporate external measures.

The remainder of this paper is divided into several sections. The Background section covers the relevant machine learning literature. The Method section describes a number of unique properties to our method, including data preprocessing. The Data section describes the two data se2(t)0.964541(i)0.964.89017(i)0.9

Student	Step	Action	Duration
S01	ARCS-3 ARC-EG-MEASURE	Attempt	11.446
S01	ARCS-3 ARC-EG-MEASURE	Attempt	4.847
S01	ARCS-3 ARC-EG-MEASURE	Attempt	19.588
S01	ARCS-3 ARC-EG-MEASURE	Attempt	6.179
S01	ARCS-3 ARC-EG-MEASURE	Attempt	10.535

Table 1: Example Tutor Step

using spectral clustering instead of partition-based clustering. [6] Jebara et. al.'s work on spectral clustering with HMMs is especially important as a potential avenue for future work. [6]

Some of the prior work on E-M HMM clustering uses fixed values for $\mathbf{K}$ and for the number of states ($\mathbf{N}$) per initial model[7]. Other examples use fixed initial values for $\mathbf{K}$, but allow the merging or splitting of clusters. For example, Schliep uses "model surgery", which merges and splits clusters based on the total size of each cluster[11]. However, it is unclear which merge/split criteria are optimal. We will instead use *HMM-Cluster* as a subroutine for another algorithm, and so will limit it to fixed values of $\mathbf{K}$ and $\mathbf{N}$.

3 Data

We consider two data sets extracted from log files of the Geometry Cognitive Tutor. In the tutor, students are presented with a geometry problem and several empty text fields. A step in the problem requires filling in a text field. The fields are arranged systematically on each problem page and might, for example, ask for the values of angles in a polygon or for the intermediate values required to calculate the circumference of a circle.

Both data sets originate in earlier experimental studies, though only the control groups for each study will be used.

In each data set, a problem is defined as a series of steps and each step as a series of transactions. A student transaction is defined by the following four-tuple: $\langle$ Student,Step,Action,Duration $\rangle$. An action can be either an "Attempt" or "Help Request". Each data set consists of a series of these transactions, categorized by step and student. An example step is shown in Table 1.

*02*This data set originates in an experiment published in 2002. [1] The control condition includes 21 students and 57204 actions divided into 3740 steps.

*06*This data set originates in an experiment published in 2006. [10] The control condition includes 16 students and 16374(FIX) actions divided into 5217(FIX) steps.

Both data sets are similar in that they cover the same geometry units and use the same general interface, though there are some differences in both domain content and interface layout. The most important differences in the data lie in the students' distribution of actions and steps. In the 06 data, students exhibit far fewer actions per step, which complicates any direct comparison between results for the two data sets.

4 Method

A student action is defined by the following four-tuple: $\langle$ Student,Step,Action,Duration$\rangle$. Once actions are conditioned on students and steps, what remains is the tuple $\langle$ Action,Duration$\rangle$. While it is technically possible to directly analyze the data in this two-dimensional, partially continuous space, the results are difficult to interpret. Instead, consider a threshold of $\boldsymbol{\tau}$ seconds which divides actions into "fast" and "slow" actions. There exists a mapping from the bivariate $\langle$Action,Duration$\rangle$ tuple to a single four-category variable, shown in Table 2.

Guessing and Trying are fairly self-explanatory: a guess is a suspected attempt to solve using the system's correctness-feedback while a try is a suspected attempt to solve using actual problem-solving techniques. A drill is rapidly requesting hints, probably without reading them, either to get

	Attempt	Help Request
Fast	*G*uess	*D*rill
Slow	*T*ry	*R*eason

Table 2: Mapping from ⟨ Action,Duration⟩ to one variable

Input: sequence set $\mathbf{Q}$,student set $\mathbf{S}$, student learning gains $\mathbf{G}$
Output: collection $\mathbf{C}$
iteration $\mathbf{t = 0}$;
models $\mathbf{K = 2}$;
states $\mathbf{N = 2}$;
collection $\mathbf{C}^0$ = New-HMMs($\mathbf{K}$,$\mathbf{N}$);
while *termination criteria not satisfied* **do**
  iteration $\mathbf{t = t + 1}$;
  relearn $\mathbf{C}^{t-1}$ = HMM-Cluster($\mathbf{Q}$,$\mathbf{K}$,$\mathbf{C}^{t-1}$);
  create partition sets $\mathbf{P}_k^s$, $\mathbf{0 \leq k < K}$, $\mathbf{0 \leq s < S}$;
  foreach *sequence* $\mathbf{q_i} \in \mathbf{Q}$ **do**
    find the best model $\mathbf{k = \arg\max_k l(q_i|M_k^{t-1})}$;
    let $\mathbf{s_i} \in \mathbf{S}$ bet the student acting in sequence $\mathbf{q_i}$;
    assign sequence $\mathbf{q_i}$ to partition $\mathbf{P}_k^s$;
  end
  significant models $\mathbf{R}$ = Regression($\mathbf{G}$,[$\mathbf{P}_0$,...,$\mathbf{P_K}$]);
  foreach $\mathbf{M}_k^{t-1} \in \mathbf{C}^{t-1}$ **do**
    if $\mathbf{M}_k^{t-1} \in \mathbf{R}$ **then**
      assign $\mathbf{M}_k^{t-1}$ to $\mathbf{C}^t$;
    end
  end
  if *model count criteria satisfied* **then**
    $\mathbf{K = K + 1}$;
  end
  if *state count criteria satisfied* **then**
    $\mathbf{N = N + 1}$;
  end
  $\mathbf{C^t = C^t} \cup$ New-HMMs($\mathbf{K} - |\mathbf{C^t}|$,$\mathbf{N}$);
end
return $\mathbf{C^t}$;

Algorithm 2: Stepwise-HMM-Cluster

τ	

τ	Models	Max States

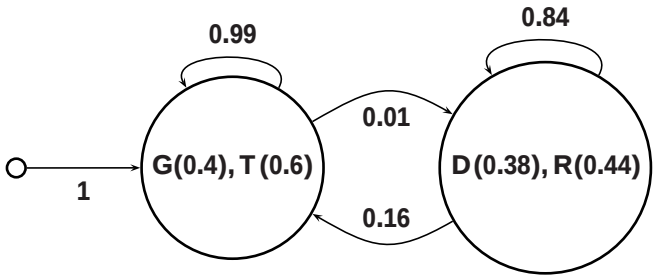

Figure 2: Dominant model for $\tau = 8$, 02 data

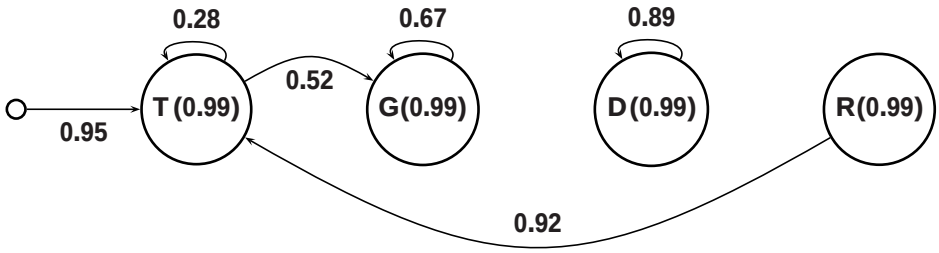

Figure 3: Try-Guess model for $\tau = 10$, 02 data

However, this raises a conflict between the results and common sense. The models shown in Figure 1 and Figure 2 have a high probability of emitting a sequence of type TGGGGG, i.e. a single Try followed by many Guesses. This is, in the educational literature, considered a very poor learning behavior. Intuitively, it represents a failed attempt to solve followed by repeated, unthinking guessing. This disagreement can be resolved by noting that no single model in any collection can be inter-

quests, usually Drill requests. Some positive-learning models involve hints as well. For example, in Figure 5, there is actually a 30% chance of generating a Reason action as the first action, before a series of attempts.

6 Conclusions

Using a traditional HMM clustering algorithm with fixed values of **K**, the number of models, and **N**, the number of states per model, it is possible to find collections of HMMs that predict learning. These models not only predict learning, but because the HMMs are relatively small, they are human-interpretable as classes of student strategies. However, this basic learning algorithm requires many random restarts, and it's unclear how to prevent the algorithm from "fishing" for results and thus overfitting.

An alternative approach is to iteratively increase the values of **K** and **N**, keeping at each iteration an optimal collection of HMMs from prior iterations. This approach, called *Stepwise-HMM-Cluster*, requires fewer clusterings to converge to a highly predictive model. Further, it avoids pre-hoc choices for **K** and **N**, biases strongly towards smaller models, provides better test-set predictions, and incorporates external measures of learning gain.

We showed that using *Stepwise-HMM-Cluster* found collections with high training-set prediction accuracy, even after adjusting for the number of models in a collection. Further, for the 02 data, withholding part of the data as a test-set still resulted in accurate predictions, on the order of a 0.5 correlation. For the other data set, a more heavily penalized selection criterion also gave similar correlations. This algorithm satisfies the primary goals of an educational data mining method: it produces interpretable models, provides good fits across data sets, and not only fits the tutor data, but predicts actual learning outcomes.

Additionally, generalization from a learning sciences perspective is not a simple matter of successful predictions on test data: it requires the production of general learning principles that can be applied independently of any given parametric model. *Stepwise-HMM-Cluster* produced such a general principle. Our results provide a strong argument that hint-scaffolding as it is presently used is not actually very effective and that most learning results from persistent attempts to solve. This suggests a new paradigm for tutoring system design that emphasizes attempts and provides hints or worked

[3] C. J. Burke and M. Rosenblatt. A markovian function of a markov chain. *The Annals of Mathematical Statistics*, 1958.

[4] Olivier Cappé, Eric Moulines, and Tobias Ryden. *Inference in Hidden Markov Models*. Springer, 2005.

[5] Pascale Fung, Grace Ngai, and Chi-Shun Cheung. Combining optimal clustering and hidden markov models for extractive summarization. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering*, pages 21–28, 2003.

[6] Tony Jebara, Yingbo Song, and Kapil Thadani. Spectral clustering and embedding with hidden markov models. In *Proceedings of the European Conference on Machine Learning*, 2007.

[7] Tim Oates, Laura Firoiu, and Paul R. Cohen. Using dynamic time warping to bootstrap hmm-based clustering of time series. *Sequence Learning*, pages 35–52, 2001.

[8] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.

[9] L. R. Rabiner, C. H. Lee, B. H. Juang, and J. G. Wilpon. Hmm clustering for connected word recognition. In *Proceedings of the IEEE ICASSP*, 1989.

[10] Ido Roll, Vincent Aleven, Bruce M. McLaren, Eunjeong Ryu, Ryan S.J.d. Baker, and Kenneth R. Koedinger. The help tutor: Does metacognitive feedback improve students' help-seeking actions, skills and learning? In