Optimal Teaching Machines

Jacob Whitehill

Computer Science & Engineering, and
Machine Perception Laboratory

UCSD
Introduction:
Human tutors and computer tutors
Human tutors

- 1-to-1 human tutoring is one of the most effective methods of instruction.

- Bloom (1984) found a $2\sigma$ gain in achievement scores of tutored students compared to students taught in traditional 1-to-many fashion.

- High cost, limited availability.

- Desirable to create computer tutors!
Automated tutors

- Some automated tutoring systems have achieved impressive learning gains (e.g., Koedinger & Anderson).
  - Geometry, algebra, LISP
- Still, it is unclear whether existing approaches to automated teaching will generalize to new domains.
- Also, contemporary computer tutors use very impoverished sensors:
  - Keyboard & mouse.
Automated tutors

- Machine perception is now equipped to provide computer tutors with rich sensor information.
- Open question: how can rich sensor information at varying time-scales be used intelligently?
- Heuristic approaches will not scale up.
- We need principled frameworks to integrate perception and action.
Optimal teaching machines

- We believe machine learning and stochastic optimal control theory may provide the necessary tools.

- This talk/workshop explores two approaches to automated teaching:
  - Approach 1: Create/learn a model of the student, and compute the optimal teaching policy using Partially Observable Markov Decision Processes.
  - Approach 2: Learn how to teach directly from records of how a human teacher acts in response to certain stimuli.
Approach 1: Optimal Teaching using POMDPs
POMDP Approach to Teaching

- POMDP = Partially Observable Markov Decision Process.
- Fundamental assumption: there is some underlying student state which the teacher cannot directly observe.
Simple example

- Suppose the teacher wishes to teach the student some skill.
- The teacher will have three actions to choose from:
  - Teach: transmit knowledge of the skill to the student.
  - Query: ask the student to demonstrate the skill.
  - Stop: terminate the session.
- Reward structure:
  - The teacher has a sense of utility and costs:
    - Teaching and querying incur a cost.
    - Bringing the student to the “learned” state earns a reward.
All-or-nothing model (Bower 1961)

- States:
  - unlearned
  - learned

- Observations:
  - correct
  - incorrect

- Probabilities:
  - $p$
  - $1-p$
  - $g$
  - $1-g$
  - 1

Thursday, August 20, 2009
Simple example solution

- Parameter values:
  - Reward for teaching: -1
  - Reward for querying: -0.5
  - Reward for stopping when learned: 10
  - Reward for stopping when unlearned: 0
  - Transition probability: 0.2
  - Guessing probability: 0.5
  - Time horizon: infinite
How should the teacher teach?

- We assume the teacher cannot directly observe the student’s state.
- We wish to devise a policy, i.e., a control law that tells the teacher what to do based on:
  - Which actions he/she has performed already.
  - What observations he/she receives from the student.
- An optimal policy maximizes the expected reward.
- This teaching problem is an example of a Partially Observable Markov Decision Process (POMDP).
Simple example

- We will present the solution to this simple example later.
- For now, we briefly introduce the POMDP framework.
POMDPs
POMDPs: Overview

- Schematic:

- Agent (Teacher)
  - $b_t$
  - Belief update
  - $b_{t+1}$

- Environment (Student)
  - $s_t$
  - State transition
  - $s_{t+1}$

- Action
- Observation
POMDPs: Definition

- Formally:
  - State space: $S$
  - Action space: $A$
  - Observation space: $O$
  - Transition probability: $P(s_{t+1} | s_t, a_t)$
  - Observation probability: $P(o_t | s_t, a_t)$
  - Reward: $R_t$, defined by function $R(s_t, a_t)$
  - Time horizon: $T$ (can be infinite)
  - Discount factor: $\gamma \in [0, 1]$
  - Prior distribution over states: $P(s_0)$
POMDPs: Policy

- The agent maintains a belief $b_t$ of the state the environment is in based on the history of actions and observations $h_t$.
- The agent needs a policy $\pi$ that decides which action $a_t$ to execute at time $t$.
- $\pi$ is a mapping from $b_t$ onto actions.
POMDPs: Policy

- Each policy $\pi$ induces an expected sum of discounted rewards over the entire time horizon.

$$\varphi(\pi) = \sum_{t=0}^{T} \gamma^t E[R_t | \pi]$$

- The optimal policy $\pi^*$ is defined:

$$\pi^* = \operatorname{arg\ max}_\pi \varphi(\pi)$$

- We can use both exact & approximate methods from the POMDP literature to compute optimal policies.
Optimal policy for simple example

![Value versus belief graph](image)

- **Value** versus **p(Learned)**
- Teach, Query, Stop regions indicated

Thursday, August 20, 2009
Optimal policy for simple example
Simple example policy

- What is notable is that reasonable - in fact, optimal - behavior emerged completely from the model.

- No need to heuristically devise a policy.
Vygotsky problem

- Item $i$ can only be learned if item $i-1$ is already known.
- Assume transition probability is $p$ for each item $i$ (given item $i-1$ is known).
- No forgetting.
- No “jumping” over levels.
- Uniform prior distribution over state

- Reward equals the state $i$ reached by the end of the session.
Vygotsky problem

- Action and state spaces: \{ 1, 2, ..., m \}
- Observation space: \{ correct, incorrect \}
- Each action $i$ is a composite action consisting of two phases:
  - Test the student to determine if he/she has yet mastered level $i$.
  - Attempt to advance the student from $i-1$ to $i$.
- There are only $m$ states!
- What is the optimal policy?
Vygotsky optimal policy
Vygotsky optimal policy

True state level

What is being taught

Belief vector
Vygotsky optimal policy

- Two-phased policy:
  - Perform “binary search” to find the student’s state $i$.
  - Gradually advance the student from $i$, ..., to $m$.
- This reasonable (optimal) behavior emerged from the model.
- No need to heuristically decide how to teach.
Approach 2: Learning from Human Teachers
Learning from human teachers

- One disadvantage of the POMDP approach is the need for a model of how the student learns (transitions) and behaves (observations).
- POMDP formulations may also not always be computationally tractable.
Learning from human teachers

Alternative:

- Forgo the student model, and instead create a \textit{direct mapping} from observations to actions.
- Learn the mapping from observing human teacher(s).
- This will be the approach we use during one project this afternoon.
One project this afternoon
Inverted Pendulum Game

- Demo

Time remaining (s): 4.8

Delay (s): 0.0

Press Left and Right to steer the car beneath the pendulum.
Teaching the Inverted Pendulum Game

- Suppose we wanted to “coach” a student to learn the inverted pendulum motor control skill.

- What might we do?
  - Select difficulty levels “appropriately”.
  - Provide encouragement, criticism.

- What might we pay attention to?
  - Computer screen
  - Student
Developing an automated coach

- We wish to develop an automated coach for the Inverted Pendulum Game.

- Sensors:
  - Game state (angle of pendulum at each time $t$)
  - Facial expression of student

- Actuators:
  - Verbal feedback: Congratulations, criticism.
  - Level changes (latency): 0, 1, 2, 3, or 4.
Developing an automated coach

- We will divide our group into several dyads:
  - Student - plays the game.
  - Coach - provides feedback and sets levels.
- All game data (student) and coaching data (teacher) will be recorded automatically.
- After collecting all data, we will then extract certain features and attempt to understand how the coach was implicitly making decisions.
Recorded data

- Game state - every ~1/10th of a second: theta
- Win state - end of each game: Win/Lose.
- Facial expression - several times per second (if face is found): 13 facial expression channels
- Feedback - whenever teacher gives it: Yes/No/Congrats/Criticism.
- Levels - start of each game: \{ 0, 1, 2, 3, 4 \}.
Recorded data

- All recorded data are *timestamped* to allow synchronization among them.
- We provide you with Perl & Matlab scripts to extract features more easily from raw data.
Possible features

- Suppose we wish to predict when the teacher says “Yes!” as encouragement to student.
- What might this action be based on?
  - Pendulum was very “balanced” during the last few seconds (perhaps?).
  - This could be approximately inferred by examining absolute value of deviation of theta from 0 over a time window of last 2 seconds.
Classification

- We can train a classifier to predict certain teaching actions from a set of useful features.
- This gives us an automated teacher.
- We can then test this teacher, either on an external data set, or on real students.
Warning:
Project under construction!
Do not despair:
We will work with you.
Assistance

- Your friendly colleagues (Javier and I) will be assisting you in your quest towards automated teaching.
  - Game playing/coaching experiment.
  - Data recording
  - Data analysis
I talked about two approaches to optimal teaching:

- **Approach 1:** Create a model of how the student learns and acts; use stochastic optimal control theory to find an *optimal teaching policy*.

- **Approach 2:** Learn a direct mapping from observations to actions by studying how a human teacher performs.
In both approaches, there is no need to heuristically define how to make decisions.

- POMDPs: make optimal decisions based on student model.
- Direct mapping: learn observation-action mapping from data on human teachers.