Emotion-Adaptive Intelligent Tutoring Systems: Perception, Prediction, and Control

Jacob Whitehill (jrwhitehill@wpi.edu)
Department of Computer Science
Worcester Polytechnic Institute
My background

My background

- 2012-2014: Co-founder of computer vision startup Emotient.
My background

• 2012-2014: Co-founder of computer vision startup Emotient.
Research group at WPI

• I’m a new faculty member in WPI’s CS Department.

• Research in applied machine learning & computer vision:
  • Affective computing
  • Human behavior recognition & optimization (education, computer games)
  • Crowdsourcing
Computer Expression Recognition Toolbox (CERT)

• Computer Expression Recognition Toolbox (“CERT”)
  • 6 basic emotions
  • 3-d pose (yaw, pitch, roll)
  • 20 facial “action units” (Ekman and Friesen 1978)
Computer Expression Recognition Toolbox (CERT)

Littlewort, Whitehill, et al. 2011
Emotion-Adaptive Intelligent tutoring systems (ITS)

• Advances in automatic facial expression recognition can enable a new generation of emotion-adaptive intelligent tutoring systems (ITS).
Emotion-Adaptive Intelligent tutoring systems (ITS)

• Advances in automatic facial expression recognition can enable a new generation of emotion-adaptive intelligent tutoring systems (ITS).

• Such ITS must be able to:

  • **Perceive** their students’ emotions.
  
  • Decide in real time how to **act** accordingly.
Closed-loop control system

- Teacher’s **actions** can impact student’s **state**: $A_1, \ldots, A_t$
Closed-loop control system

• Teacher’s **actions** can impact student’s **state**:  
  
  • Assigning a practice problem on topic X can help student increase student’s skill on X.
Closed-loop control system

- Teacher’s **actions** can impact student’s **state**:
  - Giving problems that are too difficult can cause student to become frustrated.
Closed-loop control system

- Student can give **feedback** to the teacher:

  ![Diagram](image)

  - Feedback: $O_1, \ldots, O_t$
  - Actions: $A_1, \ldots, A_t$
  - Knowledge state
  - Affective state

  Teacher

  Student
Closed-loop control system

• Student can give **feedback** to the teacher:

  • Answers to practice problems.
Closed-loop control system

• Student can give **feedback** to the teacher:

• Facial expression indicative of emotional state.
Closed-loop control system

- Implicitly, the ITS must solve a **decision problem**: what action $A_{t+1}$ to take next, given history of feedback & prior actions?

\[
A_{t+1} = f(A_1, \ldots, A_t, O_1, \ldots, O_t)
\]
Closed-loop control system

- Implicitly, the ITS must solve a **decision problem**: what action $A_{t+1}$ to take next, given history of feedback & prior actions?

**Decision function:**

$$A_{t+1} = f(A_1, \ldots, A_t, O_1, \ldots, O_t)$$

1. Automatic measurement of student engagement

**Knowledge state**

**Affective state**

**Teacher**

**Student**

- **Feedback**

$O_1, \ldots, O_t$
Closed-loop control system

- Implicitly, the ITS must solve a decision problem: what action $A_{t+1}$ to take next, given history of feedback & prior actions?

Teacher

Student

Feedback

Prior actions

Feedback

Decision function: $A_{t+1} = f(A_1, \ldots, A_t, O_1, \ldots, O_t)$

2. Approximately optimal teaching of approximately optimal learners
Automatic measurement of student engagement
Student engagement

• Starting around 1980s, “engagement” has emerged as a key metric of teaching success (e.g., Larson & Richards 1991).

• At HILT 2014 conference, “engagement” was most frequently chosen word.
Conventional methods of measuring student engagement

- Surveys and observational checklists:
  - Primacy/recency effects
  - Reluctance to give honest responses
  - Lack of temporal resolution

- Physiological sensors (EEG, GSR):
  - Obtrusive; requires physical contact
Measuring student engagement

• Our goal: develop automatic detector of student engagement using web-camera & computer vision.

• Specifically, estimate from video how “engaged” the student appears to an average observer.

• Unobtrusive

• High temporal resolution.

Whitehill, Serpell, et al. 2014
Cognitive skills training

• We developed custom cognitive skills training software for the iPad:
  • 3 games to target logical reasoning and memory.
  • Record student’s game actions synchronized with video of his/her face.
Context:
cognitive skills training
Our approach

• First step: define what “engagement” means.

• Based on video data, we defined four engagement levels:
  • 1 = Not engaged at all.
  • 2 = Nominally engaged.
  • 3 = Engaged.
  • 4 = Very engaged.
Examples

Not engaged at all

Labels rated Engagement=1

Whitehill, Serpell, et al. 2014
Examples

Nominally engaged

Labels rated Engagement=2

Whitehill, Serpell, et al. 2014
Examples

Engaged

Labels rated Engagement=3

Whitehill, Serpell, et al. 2014
Examples

Very engaged

Labels rated Engagement=4

Whitehill, Serpell, et al. 2014
We labeled ~20,000 face images from ~30 participants in a cognitive skills training task.

Given images+labels, we can use machine learning to create an automatic engagement detector.
Recognition architecture

• Using similar computational architecture as CERT, we trained an engagement detector from scratch:
  • Capture aspects of the face image not estimated by CERT, e.g., hand-on-face.
  • Tailored to the demographics of our study.
Recognition architecture

Input image

48x48 face image

Neural network

Gabor filters

Estimate of “engagement”
System accuracy

• Accuracy in distinguishing $E=4$ from $E=1$: 93%
System accuracy

- Accuracy on fine-grained classification tasks (e.g., “Is this student in engagement level 2?”):
  - Automated detector: 73%
  - Human-human: 72%

Whitehill, Serpell, et al. 2014
Correlations with task performance

• We also investigated whether engagement is correlated with students’ test performance.

• “Engagement” was positively correlated with both pre-test and post-test:
  • Pre-test: 0.57
  Post-test: 0.47

• Correlation of pre-test with post-test: 0.44
Correlations with task performance

• In other words:

• To predict a student’s post-test score, the average engagement level is a better predictor than the pre-test performance.
Smile v. Test performance

• Using CERT, we investigated how Smile intensity correlates with Posttest-minus-Pretest performance.
Smile v. Test performance

• Average smile intensity was negatively correlated (R = -0.3353, p < 0.05) with Posttest-minus-Pretest.

  • I.e., subjects who smiled less learned more.

• This is consistent with the subjective finding of the experimenters that smiles tended to express embarrassment more than achievement.

• Hoque and Picard (2011) found that smile often occurred in natural frustration.
Smile
No smile
Is the face important?

- How is the student doing?
- Is the task too easy? Too hard?
- Is the student trying to succeed?
Is the face important?
Approximately optimal teaching of approximately optimal learners (AOTAOL)
Automated teaching systems

• In intelligent tutoring systems, a big challenge is to design the **decision function** that maps from prior actions & observations to the next action:

\[ A_{t+1} = f(A_1, \ldots, A_t, O_1, \ldots, O_t) \]

• The number of possible decision functions is doubly-exponential in \( t \).
Automated teaching systems

• Note that $O_t$ can consist of both “cognitive” and “affective” observations.

• **Cognitive**: binary (correct/incorrect) responses to test questions.

• **Affective**: real-valued estimates of student engagement at 30fps.

• How to use these intelligently within an ITS?
Designing the decision function

• Approach 1 — *rule based*:

  • Manually create rules, e.g.,

    “IF student answers incorrectly 3x in a row, THEN give a hint.”
Designing the decision function

• Approach 1 — **rule based**:

  • Manually create rules, e.g.,

    “IF student answers incorrectly 3x in a row, THEN give a hint.”

  • But this approach can become intractable as size & complexity of the curriculum, and number of sensors (camera, EEG), increase.
Designing the decision function

- Approach 2 — **model-based optimization**

1. Create a predictive model of how student changes state (**dynamics model**) given the teacher’s action:

   \[ P(s_{t+1} \mid s_t, a_t) \]
Designing the decision function

• Approach 2 — model-based optimization:

2. Create a model of how student expresses her state (observation model):

\[ P(o_t \mid s_t, a_t) \]
Designing the decision function

• Approach 2 — model-based optimization:

3. Decide which states are more preferable via preference function $r$.

• Examples:

• We prefer student to know a skill over not knowing the skill.

• We prefer student to be “engaged” over “extremely frustrated”.
Designing the decision function

• Approach 2 — *model-based optimization*:

4. Find the decision function \( f \) that maximizes the expected long-term goodness of student states:

\[
    f = \arg \max_{f'} \mathbb{E} \left[ \sum_{t=1}^{T} r(S_t) \mid f' \right]
\]
Designing the decision function

• Approach 2 — **model-based optimization**:

  • The state $S_t$ is not directly observed; this is a Partially Observable Markov Decision Process (POMDP).

  • We can estimate and update $S_t$ at each timestep:

    $\frac{P(s_{t+1} | o_{1:t}, a_{1:t})}{Z}$

    $\propto \int P(s_{t+1} | s_t, a_t)P(o_t | s_t, a_t)P(s_t | o_{1:t-1}, a_{1:t-1}) ds_t$

    dynamics  observation likelihood  previous state estimate
Designing the decision function

- Approach 2 — **model-based optimization**:

4. Find the decision function $f$ that maximizes the expected long-term goodness of student states:

$$f = \arg \max_{f'} \mathbb{E} \left[ \sum_{t=1}^{T} r(S_t) \mid f' \right]$$

- Exact optimization of POMDPs is intractable.

- But there are a variety of reinforcement learning methods to solve them approximately.
Designing the decision function

• Approach 2 — **model-based optimization**:

  • Analogy: To pilot an airplane, having a model of how the plane responds to the pilot’s actions is very useful.

  • Don’t need to randomly “twiddle” with the yoke & throttles until we stumble onto the right inputs.

  • Don’t need to manually decide how to respond to every possible flight scenario.
Model-based optimal teaching

- Model-based control theory has been used for creating automated teaching systems for >50 years (Suppes 1964; Matheson 1964; Atkinson 1980; Smallwood 1971; ...)

Whitehill & Movellan (in review)
Model-based optimal teaching

• Model-based control theory has been used for creating automated teaching systems for >50 years (Suppes 1964; Matheson 1964; Atkinson 1980; Smallwood 1971; …)

• But most attempts have used very simplistic (2-state) models of the learner (Bower 1961).
Model-based optimal teaching

• Model-based control theory has been used for creating automated teaching systems for >50 years (Suppes 1964; Matheson 1964; Atkinson 1980; Smallwood 1971; …)

• But most attempts have used very simplistic (2-state) models of the learner (Bower 1961).

![Diagram of model-based optimal teaching](image)
Model-based optimal teaching

- Model-based control theory has been used for creating automated teaching systems for >50 years (Suppes 1964; Matheson 1964; Atkinson 1980; Smallwood 1971; ...)

- But most attempts have used very simplistic (2-state) models of the learner (Bower 1961).
  - Can’t model fine gradations in knowledge.
  - Can’t model student’s affective state.
Approximately optimal teaching of approximately optimal learners (AOTAOL)

- **Purpose:**
  - Develop a *more powerful* student model *without needing huge amounts of data* to estimate parameters.
  - Develop a control-based architecture for optimizing the decision function of an ITS.
Approximately optimal teaching of approximately optimal learners (AOTAOL)

• Key:

• When creating the student dynamics model, assume the learner is approximately rational in how she uses information from teacher to update her belief about curriculum.

• Builds on work by Rafferty, et al. (2011).
Concept learning

- Application domain of this project is concept learning.

- **Concept learning**: learn categories of objects by observing examples and their associated labels. (e.g., Tenenbaum 1999; Shafto & Goodman 2008).
Rosetta Stone example

- When learning the German word \textbf{Frühstück}, I may have some prior belief (e.g., uniform distribution) over which concept it might mean.
Rosetta Stone example

• Here is an image that exemplifies the word *Frühstück*: 

![Image of a boy eating breakfast](image-url)
Rosetta Stone example

• Here is an image that exemplifies the word *Frühstück*:

What concepts do you see —
boy? breakfast? table? ...

Whitehill & Movellan (in review)
Rosetta Stone example

• Given this evidence, we update our belief:
Rosetta Stone example

• Here is a second image that exemplifies **Frühstück**: 
Rosetta Stone example

• Here is a second image that exemplifies *Frühstück*:

What concepts do you see —
girl? breakfast? pink? ...

Whitehill & Movellan (in review)
Rosetta Stone example

- Given this evidence, we *update* our belief again:
Rosetta Stone concept learning task

- Rosetta Stone problem:

  - Given a target vocabulary and set of examples: how do we optimize \textit{which word we teach}, and \textit{which example we teach it with}, for each student at each timestep?

  - Goal: teach students to threshold proficiency as quickly as possible.
Formalization

Timestep 1

\[ Y_1 \rightarrow C_1 \rightarrow A_{11} \ldots A_{1n} \]

Timestep \( t \)

\[ Y_t \rightarrow C_t \rightarrow A_{t1} \ldots A_{tn} \]
Student’s task is to infer $w_1, \ldots, w_n$ based on evidence (shaded variables) provided by the teacher.
The student’s inference process (according to standard rules of probability) defines the **dynamics** of the student model.
For the **observation model**, we assume the student responds to questions about word $i$’s meaning with probability $P(w_i \mid y_1, \ldots y_t, a_{q_1}, \ldots, a_{q_t})$
For the preference function, we let $r(a)$ represent the expected amount of time it takes the student to process action $a$:

- **Teach** actions: Show image $k$ as an example (or counterexample) for word $j$.

- **Ask** actions: Ask which of two images $(k_1,k_2)$ is more likely to represent word $i$?

- **Test** actions: Ask the student to report the meanings of all words $1,\ldots,n$.

- We can estimate these time costs on people recruited on MTurk.
Optimization

• Given the dynamics and observation models, along with the preference function, we can now optimize the decision function:

\[ f = \arg \max_{f'} \mathbb{E} \left[ \sum_{t=1}^{T} r(S_t) \mid f' \right] \]

• We use policy gradient descent (Williams 1992) and a particle filter to model student’s state.
Experiment

- Using AOTAOL to create an automated teacher, we conducted experiment on Amazon Mechanical Turk (n=90):

- 10 words:

<table>
<thead>
<tr>
<th>Word</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>duzetuzi</td>
<td>man</td>
</tr>
<tr>
<td>fota</td>
<td>woman</td>
</tr>
<tr>
<td>nokidono</td>
<td>boy</td>
</tr>
<tr>
<td>mininami</td>
<td>girl</td>
</tr>
<tr>
<td>pipesu</td>
<td>dog</td>
</tr>
<tr>
<td>mekizo</td>
<td>cat</td>
</tr>
<tr>
<td>xisaxepe</td>
<td>bird</td>
</tr>
<tr>
<td>botazi</td>
<td>rabbit</td>
</tr>
<tr>
<td>koto</td>
<td>eat</td>
</tr>
<tr>
<td>notesabi</td>
<td>drink</td>
</tr>
</tbody>
</table>
Experiment

• 56 examples:
Experiment

• Complexity:

• At each timestep, the teacher must decide from among >30,000 different actions.
Experimental conditions

1. AOTAOL

2. HandCrafted1
   - Teach a randomly chosen word at each round using an appropriate image.

3. HandCrafted2
   - Teach a randomly chosen word that the student has not yet learned (based on student’s responses to test questions).
   - Both HandCrafted1 and HandCrafted2 also had parameters which were optimized in simulation to give them a fair chance.
Results of MTurk experiment

- AOTAOL teacher delivered *comparable or better* performance to 2 hand-crafted decision functions.
Results of MTurk experiment

- AOTAOL teacher was 24% faster than HandCrafted1 ($p<0.01$).
Results of MTurk experiment

- **No need** to hand-code rules to adapt to new vocabulary, examples, or prior student knowledge.
Why little difference between AOTAOL and HandCrafted2?

• There is evidence of cheating in HandCrafted1 and HandCrafted2, but not in AOTAOL.
Why little difference between AOTAOL and HandCrafted2?

- There is evidence of cheating in HandCrafted1 and HandCrafted2, but not in AOTAOL.

- 5 students in HandCrafted2 passed the test immediately after starting the task, without seeing a single word+image example!

- They may have “paired up” with another person who had already learned the vocabulary.
Why little difference between AOTAOL and HandCrafted2?

• The learners may have compensated for bad teaching policies that showed words redundantly:
Why little difference between AOTAOL and HandCrafted2?

• The learners may have *compensated* for bad teaching policies that showed words redundantly:
  
  • Using the student model, we can estimate the *information gain* of each word+image example.
Why little difference between AOTAOL and HandCrafted2?

• The learners may have *compensated* for bad teaching policies that showed words redundantly:

  • Using the student model, we can estimate the information gain of each word+image example.

  • Correlation between time spent on each example and information gain: \( r=0.31 \)
Why little difference between AOTAOL and HandCrafted2?

- The learners may have *compensated* for bad teaching policies that showed words redundantly:
  - Using the student model, we can estimate the *information gain* of each word+image example.
  - Correlation between time spent on each example and information gain: $r=0.31$
  - Avg. information gain for each teacher: AOTAOL > HandCrafted2 > HandCrafted1.
AOTAOL versus All-or-none Student Model

• Which learning model predicted students’ test scores more accurately?
AOTAOL versus All-or-none Student Model

• Which learning model predicted students’ test scores more accurately?
  
• All tests for every student:
  
  • All-or-none: $r=0.81 \ (p<0.0001)$
  
  AOTAOL: $r=0.79 \ (p<0.0001)$
AOTAOL versus All-or-none Student Model

- Which learning model predicted students’ test scores more accurately?

  - All tests for every student:
    - All-or-none: \( r=0.81 \) (\( p<0.0001 \))
    - AOTAOL: \( r=0.79 \) (\( p<0.0001 \))
  
  - All tests except the first:
    - All-or-none: \( r=-0.07 \) (\( p=0.40 \))
    - AOTAOL: \( r=0.19 \) (\( p=0.03 \))

AOTAOL versus All-or-none Student Model

• Which learning model predicted students’ test scores more accurately?

  • All tests for every student:
    - All-or-none: \( r=0.81 \) (\( p<0.0001 \))
    - AOTAOL: \( r=0.79 \) (\( p<0.0001 \))

  • All tests except the first:
    - All-or-none: \( r=-0.07 \) (\( p=0.40 \))
    - AOTAOL: \( r=0.19 \) (\( p=0.03 \))

While all-or-none model can learn coarse learning trajectories for each student, it has difficulty modeling fine-grained differences in performance across multiple students.
Affect while learning

- In pilot exploration of students’ affect, we found that students were usually engaged in the task.
Affect while learning

- There were, however, occasional moments of non-engagement.
Incorporating affective observations

• With a POMDP as the underlying framework, affective observations can be incorporated seamlessly into the optimization:

\[
P(s_{t+1} \mid o_{1:t}, a_{1:t})
\]

\[\propto \int P(s_{t+1} \mid s_t, a_t) P(o_t \mid s_t, a_t) P(s_t \mid o_{1:t-1}, a_{1:t-1}) ds_t\]

observation likelihood

• Affective observations become just another component of the observation model.
End
Learned control policy

Figure 5: Policy of the OptimizedTeacher. Each row corresponds to the policy weight vector $w$ for the action specified on the left, e.g., "Teach $j$" means teach the word indexed by $j$. Dark colors correspond to low values of the associated weight vector; light colors represent high values.