CS4710: Artificial Intelligence
Reasoning Under Uncertainty

Part 3: Example application of a hidden markov model called *Bayesian Model Tracing*

**material from: Brett Van De Sande. "Properties of the Bayesian Knowledge Tracing Model". Journal of Educational Data Mining, Volume 5, No 2, August 2013*

Special Topic:
Bayesian Knowledge Tracing

What is the key goal of BKT?

- A concrete example of using an uncertainty model to do something useful
- Suppose we are building an education system that teaches math to elementary or middle school students
- We want to measure how well a student knows a specific skill/knowledge component at a specific time
  - *So we know whether to give them more practice (individualized learning)*
  - *So we know what topic to move them onto next (individualized progression of topics)*
  - *Etc…*

What is the key goal of BKT?

- Why not just give a student a bunch of problems and calculate *NumCorrect / NumQuestions*

- Some serious problems with this:
  - If student gets many problems wrong early, difficult to “dig out of the hole”
  - Doesn’t take into account:
    - Students might know a skill but make a mistake
    - Students might not know a skill but guess correctly
    - Doesn’t model a student’s knowledge changing over time
      - i.e., later correct answers should be weighted more because more likely students has learned skill through practice

First: skills should be tightly defined

- The goal is not to measure *overall* skill for a broadly-defined construct
  - Such as arithmetic

- But to measure a specific skill or knowledge component
  - Such as addition of two-digit numbers where no carrying is needed
  - Generally more specific is better, but don't want to get TOO specific
    - E.g., ability to add 144 and 126

What is the typical use of BKT?

- Assess a student's knowledge of a skill
  - Note that this idea would extend to other types of problems too!
- Based on a sequence of items that are dichotomously scored
  - E.g. the student can get a score of 0 or 1 on each item (right or wrong)
    - Only true for some domains (math, etc.)
- Where each item corresponds to a single skill
- Where the student can learn on each item, due to help, feedback, scaffolding, etc.
  - i.e., a human or system provides hints, gives a short lesson, etc.

Key assumptions

- Each item must involve a single latent trait or skill
- Each skill has four parameters
- From these parameters, and the pattern of successes and failures the student has had on each relevant skill so far, we can compute latent knowledge P(Ln) and the probability P(CORR) that the learner will get the item correct

Key Assumptions

- Two-state learning model
  - Each skill is either learned or unlearned
- In problem-solving, the student can learn a skill at each opportunity to apply the skill
- A student does not forget a skill, once he or she knows it

Model Performance Assumptions

- If the student knows a skill, there is still some chance the student will slip and make a mistake.

- If the student does not know a skill, there is still some chance the student will guess correctly.

Corbett and Anderson's Model

Two Learning Parameters

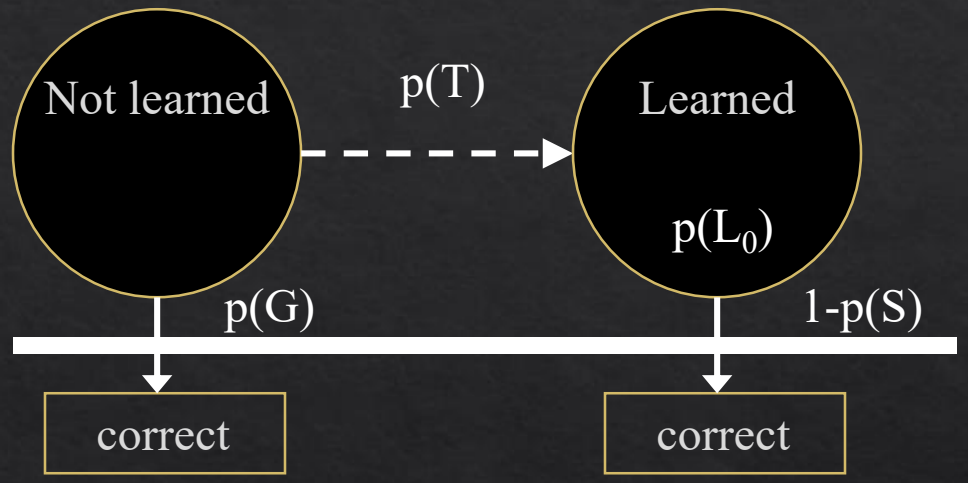
$p(L_0)$ Probability the skill is already known before the first opportunity to use the skill in problem solving.

p(T) Probability the skill will be learned at each opportunity to use the skill.

Two Performance Parameters

p(G) Probability the student will guess correctly if the skill is not known.

p(S) Probability the student will slip (make a mistake) if the skill is known.



Solving the Hidden Markov Model

- We can define the following probabilities that we are interested in:

$$P(L_j) = P(L_{j-1}) + P(T)\,(1 - P(L_{j-1}))$$

Probability the skill is learned at the jth step

$$P(C_j) = P(G)\,(1 - P(L_j)) + (1 - P(S))\,P(L_j)$$

Probability the student gets the jth problem correct

Solving the Hidden Markov Model

- **Goal:** We want to solve for $\mathbf{P(C_j)}$, but only in terms of the variables from the model
- So only other probabilities in formula can be P(S), P(G), $P(L_0)$, and P(T)

$$P(L_j) = P(L_{j-1}) + P(T)\,(1 - P(L_{j-1}))$$

Probability the skill is learned at the jth step

$$P(C_j) = P(G)\,(1 - P(L_j)) + (1 - P(S))\,P(L_j)$$

Probability the student gets the jth problem correct

Solving the Hidden Markov Model

$$P(L_j) = P(L_{j-1}) + P(T)\,(1 - P(L_{j-1}))$$

$$1 - P(L_j) = (1 - P(T))\,(1 - P(L_{j-1}))$$

Previous equation re-written in different form

Solving the Hidden Markov Model

$$P(L_j) = P(L_{j-1}) + P(T)\,(1 - P(L_{j-1}))$$

$$1 - P(L_j) = (1 - P(T))\,(1 - P(L_{j-1}))$$

Previous equation re-written in different form

$$1 - P(L_j) = (1 - P(T))^j\,(1 - P(L_0))$$

By solving the previous recursion relation (subst. method)

Solving the Hidden Markov Model

$$P(L_j) = P(L_{j-1}) + P(T)\,(1 - P(L_{j-1}))$$

$$1 - P(L_j) = (1 - P(T))\,(1 - P(L_{j-1}))$$

Previous equation re-written in different form

$$1 - P(L_j) = (1 - P(T))^j\,(1 - P(L_0))$$

By solving the previous recursion relation (subst. method)

$$P(C_j) = 1 - P(S) - (1 - P(S) - P(G))\,(1 - P(L_0))\,(1 - P(T))^j$$

Substituting the previous equation into the $P(C_j)$ equation
From previous slide

Solving the Hidden Markov Model

Our final solution is:

$$P(C_j) = 1 - P(S) - Ae^{-\beta j}$$

$$A = (1 - P(S) - P(G))\,(1 - P(L_0))$$

$$\beta = -\log(1 - P(T))$$

Solving the Hidden Markov Model

Our final solution is:

$$P(C_j) = 1 - P(S) - Ae^{-\beta j}$$

Strengths:

This is an EXACT solution to the hidden markov model we looked at.

Theoretically, can plug in j and know how likely it is a student will get a problem correct.
**Seems kind of weird doesn't it?*

Weakness:

I need to know $P(L_0)$, P(G), and P(S) to use this, but how do I get these values? One way is to fit the model to student data to find values that guess performance well.

Is very retroactive (need a student to do work before knowing the correct parameters to the model)

Several different parameter values will lead to the exact same model *(Identifiability Problem)*

Solving the Hidden Markov Model

Our final solution is:

$$P(C_j) = 1 - P(S) - Ae^{-\beta j}$$

How is this used in practice:

A student works on some problems. Data is collected.

Data is used to fit the model parameters above

Model is used to figure out how much practice (or more practice) student needs on a given topic

Another Approach: Knowledge Tracing Algorithm

Knowledge Tracing Algorithm

- Whenever the student has an opportunity to use a skill, the probability that the student knows the skill is updated using formulas derived from Bayes' Theorem.
- Same model as before (reminder on next slide).
- Able to calculate probabilities of interest in real time (while a student is working)

Corbett and Anderson's Model

Two Learning Parameters

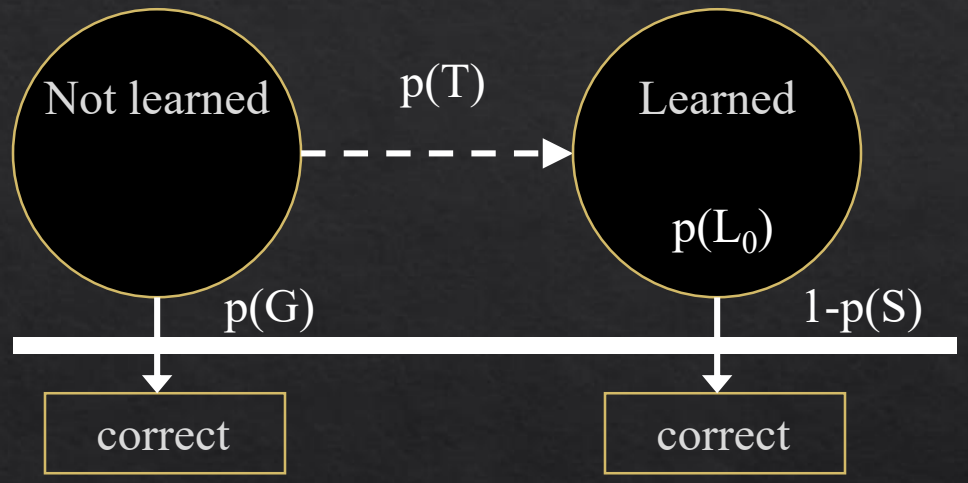
$p(L_0)$ Probability the skill is already known before the first opportunity to use the skill in problem solving.

p(T) Probability the skill will be learned at each opportunity to use the skill.

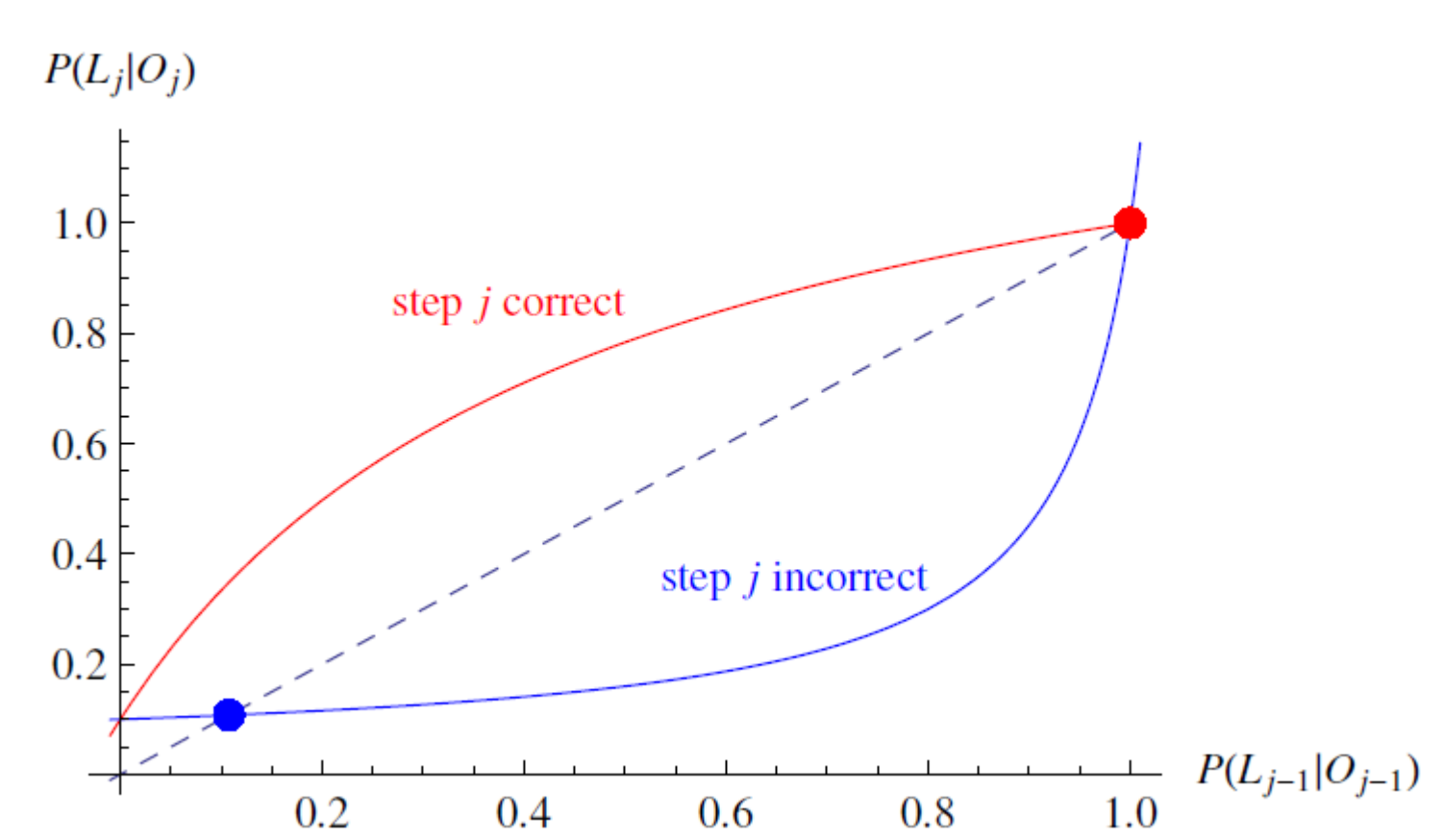
Two Performance Parameters

p(G) Probability the student will guess correctly if the skill is not known.

p(S) Probability the student will slip (make a mistake) if the skill is known.



Overview



<u>Goal:</u>

To predict $\mathbf{P(L_j)}$ for a student in real-time.

To do this, we want to calculate $\mathbf{P(L_j \mid O_j)}$

**The probability a student has learned a skill given the most recent problem they've solved*

$\mathbf{O_j}$ is whether or not student got the *j*th problem correct or not

Formulas

$$P(L_{j-1}|O_j) = \frac{P(L_{j-1}|O_{j-1})\,(1-P(S))}{P(L_{j-1}|O_{j-1})\,(1-P(S)) + [1-P(L_{j-1}|O_{j-1})]\,P(G)}\,,$$

$$o_j = \text{correct}$$

$$P(L_{j-1}|O_j) = \frac{P(L_{j-1}|O_{j-1})P(S)}{P(L_{j-1}|O_{j-1})P(S) + [1-P(L_{j-1}|O_{j-1})]\,(1-P(G))}\,,$$

$$o_j = \text{incorrect}$$

$$P(L_j|O_j) = P\,(L_{j-1}|O_j) + [1-P(L_{j-1}|O_j)]\,P(T)\,.$$

Knowledge Tracing

- How do we know if a knowledge tracing model is any good?
- Our primary goal is to predict ***knowledge***
- But knowledge is a latent trait
- So we instead check our knowledge predictions by checking how well the model predicts ***performance***

Fit Methods

- **Goal:** To pick the best values for P(S), P(G), P(L0), and P(T) that best predict student performance in practice

- **Algorithms:**
  - Hill-Climbing
  - Hill-Climbing (Randomized Restart)
  - Simulated Annealing (YAY!)
  - Genetic

- **This is a **local search problem**! So all local search techniques apply!*

Fit Methods

- **Goal:** To pick the best values for P(S), P(G), P(L0), and P(T) that best predict student performance in practice
- **Objective Function:** How well the model predicts given student performance (data set already in hand)
- **Generating Neighbors:** Many choices, but tweak the four parameters above a bit.
- You should be comfortable with implementing local search algorithms if you were asked to

Model Degeneracy

Conceptual Idea Behind Knowledge Tracing

- Knowing a skill generally leads to correct performance
- Correct performance implies that a student knows the relevant skill
- Hence, by looking at whether a student's performance is correct, we can infer whether they know the skill

Essentially

- A knowledge model is degenerate when it violates this idea
- When knowing a skill leads to worse performance
- When getting a skill wrong means you know it

Theoretical Degeneracy (Baker, Corbett, & Aleven, 2008)

- P(S) > 0.5
  - A student who knows a skill is more likely to get a wrong answer than a correct answer
- P(G) > 0.5
  - A student who does not know a skill is more likely to get a correct answer than a wrong answer

Empirical Degeneracy (Baker, Corbett, & Aleven, 2008)

- Actual behavior by a model that violates the link between knowledge and performance

Empirical Degeneracy: Test 1 (Concrete Version)

- If a student's first 3 actions in the tutor are correct
- The model's estimated probability that the student knows the skill
- Should be higher than before these 3 actions.

Examples

- $P(L_0)= 0.2$
- Maria gets her first three actions right
- $P(L_3)= 0.1$

- $P(L_0)= 0.2$
- Elmo gets his first ten actions right
- $P(L_{10})= 0.42$
- Elmo gets his next 300 actions right
- $P(L_{310})= 0.42$
- Elmo’s school quits using the tutor

Model Degeneracy

- What about this alternate definition of model degeneracy:
- P(G)+P(S) > 1.0 Always leads to a degenerative model
- Why might this definition make sense?

Model Degeneracy

- So, to prevent this problem, we simply add at least one of these additional constraints to our model's fit:
- $P(G) < 0.5$ AND $P(S) < 0.5$
- $P(G)+P(S) < 1.0$
- Why might this definition make sense?