Bayesian Knowledge Tracing (BKT)

- The classic approach for measuring tightly defined skill in online learning
- First proposed by Richard Atkinson
- Most thoroughly articulated and studied by Albert Corbett and John Anderson
The key goal of BKT

- Measuring how well a student knows a specific skill/knowledge component at a specific time

- Based on their past history of performance with that skill/KC
Skills should be tightly defined

- Unlike approaches such as Item Response Theory (later this week)

- 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
What is the typical use of BKT?

- Assess a student’s knowledge of skill/KC X
- 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
- Where each item corresponds to a single skill
- Where the student can learn on each item, due to help, feedback, scaffolding, etc.
Key Assumptions

- Each item must involve a single latent trait or skill
  - Different from PFA, which we’ll talk about next lecture

- Each skill has four parameters

- Only the first attempt on each item matters
  - i.e. is included in calculations
Key Assumptions

- 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 $P(L_n)$
  - 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.
Classical BKT

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.
Classical BKT

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.
Classical BKT

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.
Classical BKT

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.
Classical BKT

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.
Predicting Current Student Correctness

\[ \text{PCORR} = P(Ln) \times P(\sim S) + P(\sim Ln) \times P(G) \]
Bayesian Knowledge Tracing

- 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.
Formulas

\[ P(L_{n-1}|\text{Correct}_n) = \frac{P(L_{n-1}) \times (1 - P(S))}{P(L_{n-1}) \times (1 - P(S)) + (1 - P(L_{n-1})) \times (P(G))} \]

\[ P(L_{n-1}|\text{Incorrect}_n) = \frac{P(L_{n-1}) \times P(S)}{P(L_{n-1}) \times P(S) + (1 - P(L_{n-1})) \times (1 - P(G))} \]

\[ P(L_n|\text{Action}_n) = P(L_{n-1}|\text{Action}_n) + ((1 - P(L_{n-1}|\text{Action}_n)) \times P(T)) \]
Example

- $P(L_0) = 0.4$, $P(T) = 0.1$, $P(S) = 0.3$, $P(G) = 0.2$

<table>
<thead>
<tr>
<th>Actual</th>
<th>$P(L_{n-1})$</th>
<th>$P(L_{n-1} \mid \text{actual})$</th>
<th>$P(L_n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

- \( P(L_0) = 0.4, \ P(T) = 0.1, \ P(S) = 0.3, \ P(G) = 0.2 \)

| Actual | \( P(L_{n-1}) \) | \( P(L_{n-1} | \text{actual}) \) | \( P(L_n) \) |
|--------|-----------------|-------------------------------|----------------|
| 0      | 0.4             | \( \frac{(0.4)(0.3)}{(0.4)(0.3)+(0.6)(0.8)} \) |                |
Example

- $P(L_0) = 0.4$, $P(T) = 0.1$, $P(S) = 0.3$, $P(G) = 0.2$

<table>
<thead>
<tr>
<th>Actual</th>
<th>$P(L_{n-1})$</th>
<th>$P(L_{n-1} \mid \text{actual})$</th>
<th>$P(L_n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.4</td>
<td>$\frac{0.12}{(0.12) + (0.48)}$</td>
<td></td>
</tr>
</tbody>
</table>
Example

- $P(L_0) = 0.4$, $P(T) = 0.1$, $P(S) = 0.3$, $P(G) = 0.2$

| Actual | $P(L_{n-1})$ | $P(L_{n-1} | \text{actual})$ | $P(L_n)$ |
|--------|--------------|------------------------------|---------|
| 0      | 0.4          | 0.2                          |         |
Example

- \( P(L_0) = 0.4, \ P(T) = 0.1, \ P(S) = 0.3, \ P(G) = 0.2 \)

<table>
<thead>
<tr>
<th>Actual</th>
<th>( P(L_{n-1}) )</th>
<th>( P(L_{n-1} \mid \text{actual}) )</th>
<th>( P(L_n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.4</td>
<td>0.2</td>
<td>0.2 + (0.8)(0.1)</td>
</tr>
</tbody>
</table>
Example

- \( P(L_0) = 0.4, P(T) = 0.1, P(S) = 0.3, P(G) = 0.2 \)

<table>
<thead>
<tr>
<th>Actual</th>
<th>( P(L_{n-1}) )</th>
<th>( P(L_{n-1} \mid \text{actual}) )</th>
<th>( P(L_n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.4</td>
<td>0.2</td>
<td>0.28</td>
</tr>
</tbody>
</table>
Example

- \( P(L_0) = 0.4, \ P(T) = 0.1, \ P(S) = 0.3, \ P(G) = 0.2 \)

| Actual | \( P(L_{n-1}) \) | \( P(L_{n-1} | \text{actual}) \) | \( P(L_n) \) |
|--------|----------------|----------------|----------------|
| 0      | 0.4            | 0.2            | 0.28           |
| 0.28   |                |                |                |
Example

- $P(L_0) = 0.4$, $P(T) = 0.1$, $P(S) = 0.3$, $P(G) = 0.2$

<table>
<thead>
<tr>
<th>Actual</th>
<th>$P(L_{n-1})$</th>
<th>$P(L_{n-1} \mid \text{actual})$</th>
<th>$P(L_n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.4</td>
<td>0.2</td>
<td>0.28</td>
</tr>
<tr>
<td>1</td>
<td>0.28</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

- $P(L_0) = 0.4$, $P(T) = 0.1$, $P(S) = 0.3$, $P(G) = 0.2$

| Actual | $P(L_{n-1})$ | $P(L_{n-1} | \text{actual})$ | $P(L_n)$        |
|--------|--------------|-------------------------------|-----------------|
| 0      | 0.4          | 0.2                           | 0.28            |
| 1      | 0.28         | $\frac{(0.28)(0.7)}{(0.28)(0.7)+(0.72)(0.2)}$ |                |
Example

- \( P(L_0) = 0.4, \ P(T) = 0.1, \ P(S) = 0.3, \ P(G) = 0.2 \)

<table>
<thead>
<tr>
<th>Actual</th>
<th>( P(L_{n-1}) )</th>
<th>( P(L_{n-1} \mid \text{actual}) )</th>
<th>( P(L_n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.4</td>
<td>0.2</td>
<td>0.28</td>
</tr>
<tr>
<td>1</td>
<td>0.28</td>
<td>( \frac{0.196}{(0.196)+(0.144)} )</td>
<td>(0.196)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0.144)</td>
</tr>
</tbody>
</table>
Example

- $P(L_0) = 0.4$, $P(T) = 0.1$, $P(S) = 0.3$, $P(G) = 0.2$

<table>
<thead>
<tr>
<th>Actual</th>
<th>$P(L_{n-1})$</th>
<th>$P(L_{n-1} \mid \text{actual})$</th>
<th>$P(L_n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.4</td>
<td>0.2</td>
<td>0.28</td>
</tr>
<tr>
<td>1</td>
<td>0.28</td>
<td>0.58</td>
<td></td>
</tr>
</tbody>
</table>
Example

- \( P(L_0) = 0.4, P(T) = 0.1, P(S) = 0.3, P(G) = 0.2 \)

<table>
<thead>
<tr>
<th>Actual</th>
<th>( P(L_{n-1}) )</th>
<th>( P(L_{n-1} \mid \text{actual}) )</th>
<th>( P(L_n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.4</td>
<td>0.2</td>
<td>0.28</td>
</tr>
<tr>
<td>1</td>
<td>0.28</td>
<td>0.58</td>
<td>(0.58) + (0.42)(0.1)</td>
</tr>
</tbody>
</table>
Example

- $P(L_0) = 0.4$, $P(T) = 0.1$, $P(S) = 0.3$, $P(G) = 0.2$

<table>
<thead>
<tr>
<th>Actual</th>
<th>$P(L_{n-1})$</th>
<th>$P(L_{n-1} \mid \text{actual})$</th>
<th>$P(L_n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.4</td>
<td>0.2</td>
<td>0.28</td>
</tr>
<tr>
<td>1</td>
<td>0.28</td>
<td>0.48</td>
<td>0.62</td>
</tr>
</tbody>
</table>
Only uses first problem attempt on each item

Throws out information…

But uses the clearest information…

Several variants to BKT break this assumption at least in part – more on that later in the week
Parameter Constraints

- Typically, the potential values of BKT parameters are constrained.
- To avoid *model degeneracy*.
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
Constraints Proposed

- Beck
  - $P(G) + P(S) < 1.0$

  - $P(G) < 0.5, P(S) < 0.5$

  - $P(G) < 0.3, P(S) < 0.1$
Knowledge Tracing

- How do we know if a knowledge tracing model is any good?
- Our primary goal is to predict *knowledge*
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
Knowledge Tracing

- How do we know if a knowledge tracing model is any good?

- Our primary goal is to predict knowledge

- But knowledge is latent

- So we instead check our knowledge predictions by checking how well the model predicts performance
Fitting a Knowledge-Tracing Model

- In principle, any set of four parameters can be used by knowledge-tracing.

- But parameters that predict student performance better are preferred.
Knowledge Tracing

- So, we pick the knowledge tracing parameters that best predict performance.
- Defined as whether a student’s action will be correct or wrong at a given time.
Fit Methods

- I could spend an hour talking about the ways to fit Bayesian Knowledge Tracing models.
Three public tools

- hmmsclbl
  - [http://yudelson.info/hmmsclbl.html](http://yudelson.info/hmmsclbl.html)
- BNT-SM: Bayes Net Toolkit – Student Modeling
- BKT-BF: BKT-Brute Force (Grid Search)
Which one should you use?

- They’re all fine – they work approximately equally well
- My group uses BKT-BF to fit Classical BKT and BNT-SM to fit variant models
- But some commercial colleagues use Fit BKT at Scale
Note…

- The Equation Solver in Excel replicably does worse for this problem than these packages
Extensions

- There have been many extensions to BKT

- We will discuss some of the most important ones in class, later in the week
Next Up

- Performance Factors Analysis