Bayesian Knowledge Tracing and Other Predictive Models

Zachary A. Pardos

PSLC Summer School 2012

Ph.D. Committee
Dr. Neil Heffernan, Advisor, WPI - Computer Science
Dr. Ryan S.J.d. Baker, WPI - Social Science and Policy Studies
Dr. Gabor Sarkozy, WPI - Computer Science
Dr. Kenneth Koedinger, CMU - Human Computer Interaction Institute

zpardos@gmail.com
http://wpi.edu/~zpardos
Outline of Talk

- Introduction to Knowledge Tracing
  - History
  - Intuition
  - Generative example
  - Influence of parameters
    - Demo?
  - Prior Per Student model
  - Variations (and other models)
  - MATLAB Code demo
History in the literature

• Introduced in 1995 (Corbett & Anderson)
• Four parameter simplification of ACT-R theory of skill acquisition (Anderson 1993)
• Computations based on a variation of Bayesian calculations proposed in 1972 (Atkinson)
• Formalized as equivalent to a Dynamic Bayesian Network (Rye, 2004) “Student modeling based on belief networks”
Real world deployment

- Used in the Cognitive Tutors (Carnegie Learning) to determine when a student has mastered a skill and can move on in the curriculum
- Replies on a skill model (tagging of skills to items)
- Parameters of the model can be learned with Expectation Maximization (EM) or grid search
For some Skill K:

Given a student’s response sequence 1 to n, predict n+1

Chronological response sequence for student Y
[ 0 = Incorrect response 1 = Correct response]
Track knowledge over time
*(model of learning)*
Knowledge Tracing (KT) can be represented as a simple HMM

Latent

Observed

Node representations
K = Knowledge node
Q = Question node

Node states
K = Two state (0 or 1)
Q = Two state (0 or 1)
Four parameters of the KT model:

- $P(L_0)$ = Probability of initial knowledge
- $P(T)$ = Probability of learning
- $P(G)$ = Probability of guess
- $P(S)$ = Probability of slip

Probability of forgetting assumed to be zero (fixed)
Formulas for inference and prediction

If Correct$_n$

$$P(L_{n-1}) = \frac{P(L_{n-1})*(1-P(S))}{P(L_{n-1})*(1-P(S)) + (1-P(L_{n-1}))*P(G)}$$  \hspace{1cm} (1)

Incorrect$_n$

$$P(L_{n-1}) = \frac{P(L_{n-1})*P(S)}{P(L_{n-1})*P(S) + (1-P(L_{n-1}))*(1-P(G))}$$  \hspace{1cm} (2)

$$P(L_n) = P((L_{n-1})*(1-P(F)) + ((1-P(L_{n-1}))*P(T))$$  \hspace{1cm} (3)

• Derivation (Reye, JAIED 2004):

$$p(L_{n-1} \mid C_n) = \frac{p(C_n \mid L_{n-1})p(L_{n-1})}{p(C_n \mid L_{n-1})p(L_{n-1}) + p(\neg L_{n-1})p(C_n \mid \neg L_{n-1})}$$

• Formulas use Bayes Theorem to make inferences about latent variable
Knowledge Tracing

How a Bayesian Knowledge Tracing World Works

Generative - Example
How a Bayesian Knowledge Tracing World Works

Prior = 0.40

\[ P(L_0) \]
How a Bayesian Knowledge Tracing World Works

Prior = 0.40

\[
P(L_0)
\]
How a Bayesian Knowledge Tracing World Works

Prior = 0.40

Knowledge Tracing

P(L₀) = Probability of initial knowledge
P(T) = Probability of learning
P(G) = Probability of guess
P(S) = Probability of slip

Nodes representation
K = knowledge node
Q = question node

Node states
K = two state (0 or 1)
Q = two state (0 or 1)
How a Bayesian Knowledge Tracing World Works

Prior = 0.40

Knowledge Tracing

\[ P(L_0) \]

Nodes representation:
- \( K \) = knowledge node
- \( Q \) = question node

Node states:
- \( K \) = two state (0 or 1)
- \( Q \) = two state (0 or 1)

Model Parameters:
- \( P(L) \) = Probability of initial knowledge
- \( P(T) \) = Probability of learning
- \( P(G) \) = Probability of guess
- \( P(S) \) = Probability of slip
How a Bayesian Knowledge Tracing World Works

Prior = 0.40     Guess = 0.14

knowledge

P(L₀)

0

P(G)

question
How a Bayesian Knowledge Tracing World Works

Prior = 0.40     Guess = 0.14

Model Parameters

- $P(L_0) = \text{Probability of initial knowledge}$
- $P(T) = \text{Probability of learning}$
- $P(G) = \text{Probability of guess}$
- $P(S) = \text{Probability of slip}$

Nodes representation

- $K = \text{knowledge node}$
- $Q = \text{question node}$

Node states

- $K = \text{two state (0 or 1)}$
- $Q = \text{two state (0 or 1)}$
Knowledge Tracing

How a Bayesian Knowledge Tracing World Works

Prior = 0.40     Guess = 0.14

knowledge

P(L₀)

question

P(G)
How a Bayesian Knowledge Tracing World Works

Prior = 0.40     Guess = 0.14     Learn = 0.20

knowledge

P(L₀)  P(T)

question
How a Bayesian Knowledge Tracing World Works

Prior = 0.40    Guess = 0.14    Learn = 0.20
How a Bayesian Knowledge Tracing World Works

Prior = 0.40     Guess = 0.14     Learn = 0.20

Knowledge Tracing

node states
K = two state (0 or 1)
Q = two state (0 or 1)

Model Parameters
P(L₀) = Probability of initial knowledge
P(T) = Probability of learning
P(G) = Probability of guess
P(S) = Probability of slip
How a Bayesian Knowledge Tracing World Works

Prior = 0.40    Guess = 0.14    Learn = 0.20    Slip = 0.05

Knowledge Tracing

Prior = 0.40    Guess = 0.14    Learn = 0.20    Slip = 0.05
How a Bayesian Knowledge Tracing World Works

Prior = 0.40  Guess = 0.14  Learn = 0.20  Slip = 0.05

Knowledge Tracing

Model Parameters

- $P(L_0)$ = Probability of initial knowledge
- $P(T)$ = Probability of learning
- $P(G)$ = Probability of guess
- $P(S)$ = Probability of slip

Nodes representation

- K = knowledge node
- Q = question node

Node states

- K = two state (0 or 1)
- Q = two state (0 or 1)
How a Bayesian Knowledge Tracing World Works

Prior = 0.40  Guess = 0.14  Learn = 0.20  Slip = 0.05

Knowledge Tracing

Model Parameters

- $P(L_0)$ = Probability of initial knowledge
- $P(T)$ = Probability of learning
- $P(G)$ = Probability of guess
- $P(S)$ = Probability of slip

Nodes representation

- $K$ = knowledge node
- $Q$ = question node

Node states

- $K$ = two state (0 or 1)
- $Q$ = two state (0 or 1)
How a Bayesian Knowledge Tracing World Works

Prior = 0.40  Guess = 0.14  Learn = 0.20  Slip = 0.05

Knowledge Tracing

Generalization of the response prediction calculation:

$$P(Correct_n) = P(L_n)(1 - P(S)) + (1 - P(L_n))P(G)$$
How a Bayesian Knowledge Tracing World Works

Prior = 0.40   Guess = 0.14   Learn = 0.20   Slip = 0.05

Generalization of the probability of learning calculation:

\[ P(L_{n+1}) = P(L_n) + (1 - P(L_n))P(T) \]
Knowledge Tracing

How a Bayesian Knowledge Tracing World Works

Prior = 0.40  Guess = 0.14  Learn = 0.20  Slip = 0.05

You want to infer $P(L_n)$ from the student’s responses

Model Parameters
- $P(L_0)$ = Probability of initial knowledge
- $P(T)$ = Probability of learning
- $P(G)$ = Probability of guess
- $P(S)$ = Probability of slip

Nodes representation
- $K$ = knowledge node
- $Q$ = question node

Node states
- $K$ = two state (0 or 1)
- $Q$ = two state (0 or 1)
How a Bayesian Knowledge Tracing World Works

Prior = 0.40  Guess = 0.14  Learn = 0.20  Slip = 0.05

You want to infer $P(L_n)$ from the student’s responses

First, infer the knowledge at the first opportunity:

$$P(\text{Knowledge}|\text{Response} = 0) = \frac{P(L_0)P(S)}{P(L_0)P(S) + (1 - P(L_0))(1 - P(G))}$$
How a Bayesian Knowledge Tracing World Works

Prior = 0.40     Guess = 0.14     Learn = 0.20     Slip = 0.05

You want to infer $P(L_n)$ from the student’s responses

First, infer the knowledge at the first opportunity:

$$P(Knowledge|Response = 0) = \frac{0.40 \cdot P(S)}{0.40 \cdot P(S) + (1 - 0.40)(1 - P(G))}$$
Knowledge Tracing

How a Bayesian Knowledge Tracing World Works

Prior = 0.40  Guess = 0.14  Learn = 0.20  Slip = 0.05

Knowledge Tracing

You want to infer $P(L_n)$ from the student’s responses

First, infer the knowledge at the first opportunity:

$$P(Knowledge|Response = 0) = \frac{0.40 \cdot 0.05}{0.40 \cdot 0.05 + (1 - 0.40)(1 - P(G))}$$
Knowledge Tracing

How a Bayesian Knowledge Tracing World Works

Prior = 0.40  Guess = 0.14  Learn = 0.20  Slip = 0.05

You want to infer $P(L_n)$ from the student’s responses

First, infer the knowledge at the first opportunity:

$$P(\text{Knowledge}|\text{Response} = 0) = \frac{0.40 \cdot 0.05}{0.40 \cdot 0.05 + (1 - 0.40)(1 - 0.14)}$$
How a Bayesian Knowledge Tracing World Works

Prior = 0.40   Guess = 0.14   Learn = 0.20   Slip = 0.05

You want to infer $P(L_n)$ from the student’s responses.

First, infer the knowledge at the first opportunity:

$$P(\text{Knowledge} | \text{Response} = 0) = \frac{0.40 \cdot 0.05}{0.40 \cdot 0.05 + (1 - 0.40)(1 - 0.14)} =$$
Knowledge Tracing

How a Bayesian Knowledge Tracing World Works

Prior = 0.40  Guess = 0.14  Learn = 0.20  Slip = 0.05

You want to infer $P(L_n)$ from the student’s responses

First, infer the knowledge at the first opportunity:

$$P(\text{Knowledge} | \text{Response} = 0) = \frac{0.40 \cdot 0.05}{0.40 \cdot 0.05 + (1 - 0.40)(1 - 0.14)} = 0.0373$$
How a Bayesian Knowledge Tracing World Works

Prior = 0.40  Guess = 0.14  Learn = 0.20  Slip = 0.05

You want to infer $P(L_n)$ from the student’s responses

Next, apply the learning transition formula:

$$P(L_{n+1}) = 0.0373 + (1 - 0.0373)(0.20) =$$
How a Bayesian Knowledge Tracing World Works

Prior = 0.40     Guess = 0.14     Learn = 0.20     Slip = 0.05

You want to infer $P(L_n)$ from the student’s responses

Next, apply the learning transition formula:

$$P(L_{n+1}) = 0.0373 + (1 - 0.0373)(0.20) = 0.2298$$

New prior for $L_{n+1}$
Knowledge Tracing

How a Bayesian Knowledge Tracing World Works

Prior = 0.40     Guess = 0.14     Learn = 0.20     Slip = 0.05

You want to infer \( P(L_n) \) from the student’s responses

Lastly, infer the knowledge at the second opportunity:

\[
P(Knowledge|Response = 1) = \frac{0.2298 \cdot (1 - 0.05)}{0.2298 \cdot (1 - 0.05) + (1 - 0.2298) \cdot 0.14} = 0.6694
\]
How a Bayesian Knowledge Tracing World Works

Prior = 0.40  Guess = 0.14  Learn = 0.20  Slip = 0.05

You want to infer $P(L_n)$ from the student’s responses

Inference calculations are applications of Bayes theorem: $P(K|Q) = \frac{P(Q|K)P(K)}{P(Q)}$
Model Tracing Step – Skill: Subtraction

P(K) = Probability of initial knowledge
P(T) = Probability of learning
P(G) = Probability of guess
P(S) = Probability of slip

Nodes representation:
K = knowledge node
Q = question node

Node states:
K = two state (0 or 1)
Q = two state (0 or 1)

Student’s last three responses to Subtraction questions (in the Unit):
- 0
- 1
- 1

Test set questions:
- P(Q) = 71%
- 74%

Latent (knowledge)
Observable (responses)
Influence of parameter values

Estimate of knowledge for student with response sequence: 0 1 1 1 1 1 1 1 1 1

Student reached 95% probability of knowledge
After 4\textsuperscript{th} opportunity

\begin{align*}
P(L_0): 0.50 &\quad P(T): 0.20 &\quad P(G): 0.14 &\quad P(S): 0.09
\end{align*}
Influence of parameter values

Estimate of knowledge for student with response sequence: 0 1 1 1 1 1 1 1 1 1

Student reached 95% probability of knowledge
After 8th opportunity

\[ P(L_0): 0.50 \quad P(T): 0.20 \quad P(G): 0.14 \quad P(S): 0.09 \]

\[ P(L_0): 0.50 \quad P(T): 0.20 \quad P(G): 0.64 \quad P(S): 0.03 \]
Intro to Knowledge Tracing

( Demo )
Parameter fitting

- EM, Grid-search, Spectral DS (Gordon)
- 1st workshop on Parameter fitting (ITS 2012)

Standard Knowledge Tracing

Prior Per Student (cold start heuristic)

Intro to Knowledge Tracing

Prior Per Student Model
• Knowledge Tracing, the current state of the art in knowledge assessment
  – Has no student specific parameters
    • Individual prior knowledge
    • Individual learn rates
  – Research objective is to add individualization to improve knowledge assessment and prediction accuracy.
Prior Individualization Approach

Do all students enter a lesson with the same background knowledge?

Node representations

K = Knowledge node
Q = Question node
S = Student node

Node states

K = Two state (0 or 1)
Q = Two state (0 or 1)
S = Multi state (1 to N)
Prior Individualization Approach

Conditional Probability Table of Student node and Individualized Prior node

- Now that the model enables a prior parameter per student, how are these parameters going to be learned?

| S value | P(L₀|S) |
|---------|--------|
| 1       | 0.05   |
| 2       | 0.30   |
| …       | 0.95   |

Several strategies tried

<table>
<thead>
<tr>
<th>P(L₀) Strategy</th>
<th>Most accurate predictor (of 42)</th>
<th>Avg. Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(L₀) Strategy</td>
<td>PPS</td>
<td>KT</td>
</tr>
<tr>
<td>Percent correct heuristic</td>
<td>33</td>
<td>8</td>
</tr>
<tr>
<td>Cold start heuristic</td>
<td>30</td>
<td>12</td>
</tr>
<tr>
<td>Random parameter values</td>
<td>26</td>
<td>16</td>
</tr>
</tbody>
</table>

(Pardos & Heffernan, 2010a)
Prior Individualization Approach

Conditional Probability Table of Student node and Individualized Prior node

• Cold Start Heuristic

CPT of Individualized Prior node

| S value | P(L₀|S) |
|---------|--------|
| 0       | 0.05   |
| 1       | 0.30   |

Zach Pardos  
Predictive Models of Student Learning  
July 20, 2012
Prior Individualization Approach

What values to use for the two priors?

| $S$ value | $P(L_0 | S)$ |
|-----------|-------------|
| 0         | 0.05        |
| 1         | 0.30        |

CPT of Individualized Prior node

Knowledge Tracing with Individualized $P(L_0 | S)$

Prior Individualization Approach

Predictive Models of Student Learning

July 20, 2012
Prior Individualization Approach

What values to use for the two priors?

1. Use ad-hoc values

| S value | P(L₀ | S) |
|---------|-------|
| 0       | 0.10  |
| 1       | 0.85  |

CPT of Individualized Prior node

P(L₀ | S)

S

K

Q
Prior Individualization Approach

What values to use for the two priors?

1. Use ad-hoc values
2. Learn the values

CPT of Individualized Prior node

| S value | P(L₀|S) |
|---------|--------|
| 0       | EM     |
| 1       | EM     |

Prior Individualization Approach

Prior Per Student Model

Predictive Models of Student Learning

July 20, 2012
Prior Individualization Approach

What values to use for the two priors?

1. Use ad-hoc values
2. Learn the values
3. Link with the guess/slip CPT

CPT of Individualized Prior node

| S value | P(L₀|S)   |
|---------|----------|
| 0       | Slip     |
| 1       | 1-Guess  |

Prior Per Student Model

Zach Pardos

Predictive Models of Student Learning

July 20, 2012
Prior Individualization Approach

What values to use for the two priors?

1. Use ad-hoc values
2. Learn the values
3. Link with the guess/slip CPT

CPT of Individualized Prior node

| S value | P(L₀|S)   |
|---------|----------|
| 0       | Slip     |
| 1       | 1-Guess  |

With an ASSISTments Platform dataset, PPS (ad-hoc) achieved an R² of 0.301 (0.176 with KT)

(Pardos & Heffernan, UMAP 2010)
Cold start heuristic was a success
• Performed well, improvement in prediction over KT in 30/42 problem sets

• Requires no extra information outside of the responses in the problem set being predicted

• Reduces the free parameters to three instead of four
  • Faster parameter training time with more accurate prediction

• The most simple individualization technique to add to existing KT models
  • One binary node addition and one arc

• Parameters can be learned from one population of students to predict another
Variations on Knowledge Tracing (and other models)
1. BKT-BF

Learns values for these parameters by performing a grid search (0.01 granularity) and chooses the set of parameters with the best squared error.

\[ P(L_0) = \text{Probability of initial knowledge} \]
\[ P(T) = \text{Probability of learning} \]
\[ P(G) = \text{Probability of guess} \]
\[ P(S) = \text{Probability of slip} \]

(Baker et al., 2010)
2. BKT-EM

Learns values for these parameters with Expectation Maximization (EM). Maximizes the log likelihood fit to the data.

\[ P(L_0) = \text{Probability of initial knowledge} \]
\[ P(T) = \text{Probability of learning} \]
\[ P(G) = \text{Probability of guess} \]
\[ P(S) = \text{Probability of slip} \]

(Chang et al., 2006)
3. BKT-CGS

**Guess and slip parameters** are assessed contextually using a regression on features generated from student performance in the tutor.

- $P(L_0) = \text{Probability of initial knowledge}$
- $P(T) = \text{Probability of learning}$
- $P(G) = \text{Probability of guess}$
- $P(S) = \text{Probability of slip}$

(Baker, Corbett, & Aleven, 2008)
4. **BKT-CSlip**

Uses the student’s averaged contextual Slip parameter learned across all incorrect actions.

\[
P(L_0) = \text{Probability of initial knowledge} \\
P(T) = \text{Probability of learning} \\
P(G) = \text{Probability of guess} \\
P(S) = \text{Probability of slip}
\]

(Baker, Corbett, & Aleven, 2008)
5. BKT-LessData

Limits students response sequence length to the most recent 15 during EM training.

\[
P(L_0) = \text{Probability of initial knowledge} \quad P(T) = \text{Probability of learning} \\
P(G) = \text{Probability of guess} \quad P(S) = \text{Probability of slip}
\]

Most recent 15 responses used (max)

(Nooraiei et al, 2011)
6. BKT-PPS

Prior per student (PPS) model which individualizes the prior parameter. Students are assigned a prior based on their response to the first question.

\[ P(L_0|S) \]

\[ P(L_0) = \text{Probability of initial knowledge} \]
\[ P(T) = \text{Probability of learning} \]
\[ P(G) = \text{Probability of guess} \]
\[ P(S) = \text{Probability of slip} \]

(Pardos & Heffernan, 2010)
7. CFAR

Correct on First Attempt Rate (CFAR) calculates the student’s percent correct on the current skill up until the question being predicted.

Student responses for Skill X: 0 1 0 1 0 1

Predicted next response would be 0.50

(Yu et al., 2010)
8. Tabling

Uses the student’s response sequence (max length 3) to predict the next response by looking up the average next response among student with the same sequence in the training set.

Training set
Student A: 0 1 1 0
Student B: 0 1 1 1
Student C: 0 1 1 1

Max table length set to 3:
Table size was $2^0 + 2^1 + 2^2 + 2^3 = 15$

Test set student: 0 0 1 _

Predicted next response would be 0.66

(Wang et al., 2011)
9. IRT

Item response theory (IRT) the standard assessment tool used for GRE testing.

\[ p(+ | v, i) = \frac{\exp(\theta_v - \sigma_i)}{1 + \exp(\theta_v - \sigma_i)} = \frac{1}{1 + e^{-(\theta_v - \sigma_i)}} \]

Where \( p(+ | v, i) \) is the probability of positive performance of student \( v \) on test item \( i \) and \( \sigma_i \) is the difficulty of item \( i \).

Extension which breaks down an item into cognitive operations (Scheiblechner, 1972)

\[ \sigma_i = \sum_{j=1}^{m} q_{ij} n_j + c \]

Where \( j \) is a cognitive operation, \( q_{ij} \) is the number of times the operation occurs in item \( i \), \( n_j \) is the difficulty of cognitive operation \( j \) and \( c \) is a scaling constant

Learning from the test addition:

\[ \sigma = \sum_{j=1}^{m} q_{ij} n_j - q_{uj} h_{ij}^* \beta_j + c \]
10. PFA

Performance Factors Analysis (PFA). Logistic regression model which elaborates on the Rasch IRT model. Predicts performance based on the count of student’s prior failures and successes on the current skill.

An overall difficulty parameter $\beta$ is also fit for each skill or each item in this formula the variant of PFA that fits $\beta$ for each skill is shown. The PFA equation is:

$$m(i, j \in KCs, s, f) = \beta_j + \sum(\gamma_j S_{ij} + \rho_j F_{ij})$$

(Pavlik et al., 2009)
Conclusion

Time Left? If yes then KT-IDEM / IEM
Else
next slide
Bayesian Knowledge Tracing
MATLAB Demo / code
Questions?


zpardos@gmail.com
http://wpi.edu/~zpardos