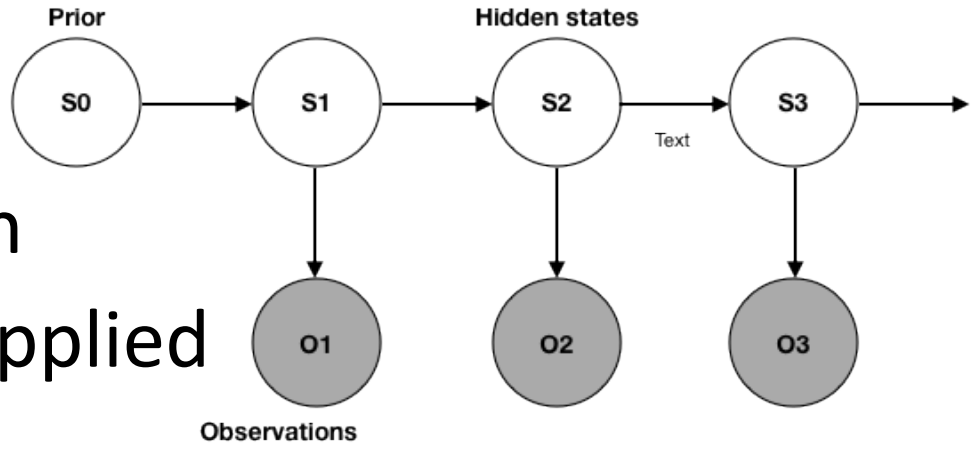
Fitting Bayesian Knowledge Tracing Models for PSLC DataShop

Michael Yudelson
Human-Computer Interaction Institute
Carnegie Mellon University

Bayesian Knowledge Tracing (1)

- Goal: infer whether student knows a skill from the pattern of right/wrong applications of the skill
- A special case of Hidden Markov Model (HMM)
- Hidden states: skill known or not known
- Observations: skill applied [in]correctly



Bayesian Knowledge Tracing (2)

- Parameters
  - pInit (pL_0) – a priori mastery of skill
  - pLearn (pT) – probability the skill will transition into "known" state
  - pSlip (pS) – probability known skill is applied incorrectly
  - pGuess (pG) – probability not known skill is applied correctly
  - pKnown (pL) – running estimate of skill mastery

Preparing Data

- Export data set with one of the KC models[DataShop]
  - PSLC DataShop http://pslcdatashop.web.cmu.edu
- Open file in Excel and rearrange columns
  - 1st column is 'First attempt'
  - 2nd column is 'Anon Student Id'
  - 3rd column is a concatenation of 'Problem Hierarchy', 'Problem Name', and 'Step Name'
  - 4th column, is 'Knowledge Component'
- Remove column headers
- Save as tab-delimited text file[Excel]

Change 'correct' to 1, everything else to 2

Mark rows w/o KC's with dot '.'

Running Analysis (1)

- Which solver configuration to use

		Solver		
		Baum-Welch	**Grad. Desc.**	**Conj. Grad. Desc.**
Structure	**by skill**	1.1	1.2	1.3.1, 1.3.2, 1.3.2*
	by student	2.1	2.2	2.3.1, 2.3.2, 2.3.2
	...			

x.3.1 - Polak-Ribiere, x.3.2 - Fletcher–Reeves, x.3.3 - Hestenes-Stiefel

- by skill + Baum-Welch is the traditional
- [Conjugate] Gradient Descent – optimize log-likelihood directly and could produce a better fit

Running Analysis (2)

- trainhmm executable terminal

```
./trainhmm.mac –s 1.1 ds76__Original_cropped.txt model.txt
```

- Parameters in matrix form: π, A, B

π	
known	pLo
not known	1-pLo

A	to known	to not known
from known	1	0
from not known	pT	1-pT

B	to correct	to incorrect
from known	1-pS	pS
from not known	pG	1-pG

Running Analysis (3)

- Useful configurations

  '-t 0.01' adjust fit **t**olerance

  '-q 1' **q**uiet mode with less output

  '-m 1' print out fit **m**etrics

  '-p 1' write **p**redictions file (specify 3rd file name after model); note column 1 is probability of correct terminal, Finder

Further Tweaking the Analysis (1)

- Parameters in matrix form: π, A, B

π	
known	pLo
not known	1-pLo

A	to known	to not known
from known	1	0
from not known	pT	1-pT

B	to correct	to incorrect
from known	1-pS	pS
from not known	pG	1-pG

- Adjusting initial values of parameters

  '-0 0.5,1.0,0.4,0.8,0.2' – default setting

  pLo = 0.5, pT = 0.4, pS = 0.2, pG = 0.2

π	
state1	1
state2	2
	...
stateN-1	N-1
stateN	

A	state1	state2		stateN-1	stateN
state1			...		
state2					
			...		
stateN-1					
stateN					

B	obs1	obs2		obsM-1	obsM
state1			...		
state2					
			...		
stateN-1					
stateN					

Further Tweaking the Analysis (2)

- Adjusting parameter ranges

  '-l 0,0,1,0,0,0,0,0,0,0' – default lower boundary

  '-u 1,1,1,0,1,1,1,0.3,0.3,1' – default upper boundary

  pForget = 0, pSlip & pGuess are capped at 0.3

π	
known	pLo
not known	1-pLo

A	to known	to not known
from known	1	0
from not known	pT	1-pT

B	to correct	to incorrect
from known	1-pS	pS
from not known	pG	1-pG

Final Notes

- Number of observations can be >2
  - E.g. treat 'hint' instances separately
- Number of states can be >2
  - Via '-n <#states>' option
- Tour de force
- To download and try visit
  - http://sites.google.com/site/myudelson/projects/fitbktatscale