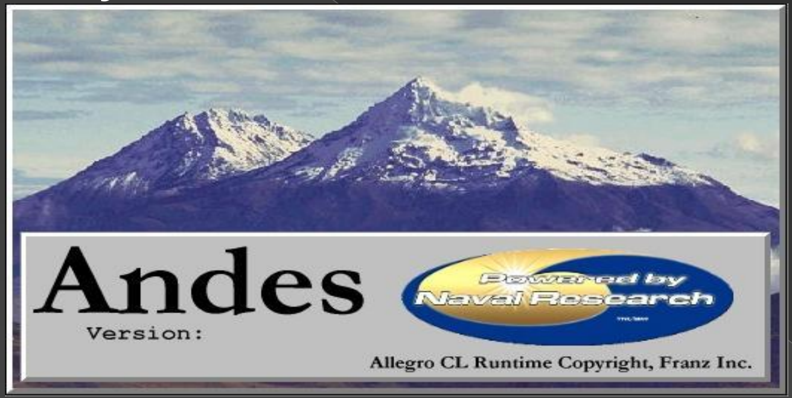
An analysis of student's gaming behaviors in an intelligent tutoring system: predictors and impacts

Kasia Muldner, Winslow Burleson,
Brett Van de Sande, Kurt VanLehn
User Modeling and User-Adapted Interaction
April 2011, Volume 21, Issue 1, pp 99–135

Seminar : Intelligent Tutoring Systems
Presented by: Praharsha Sirsi

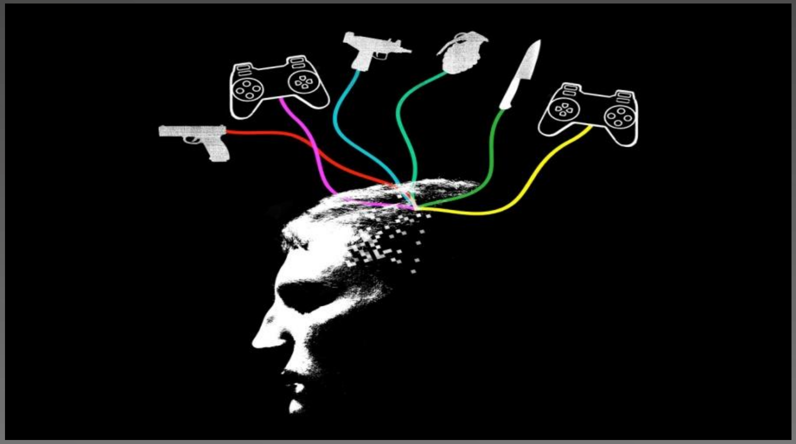
Introduction

- Intelligent Tutoring System



- What is Gaming?

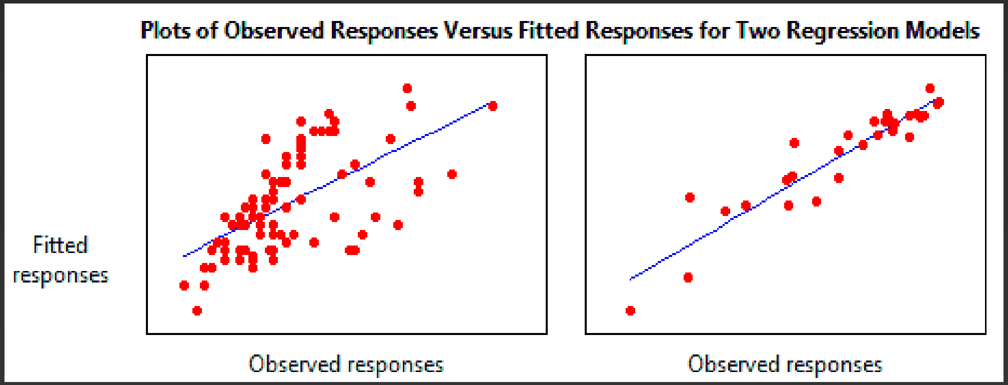
- Predicting Gaming



- Impact of Gaming

Understanding Gaming

- Student Character
  - Goal Orientation
  - Attitude
  - Self resolve



(Model accounted for only 9% of variance)

- Lesson Features
  - Confusing hints
  - Poor interface design

(Model accounted for 56% of variance)

- But, non ITS studies – Student Character is a better predictor

Analyses in the Paper

- Better predictor of gaming – Student or Problem?
- How and where students are gaming?
- Impact of ITS hints
- Model accounts for 61% of variance in gaming
- Various Educational Data Mining (EDM) techniques
- Model applied to Cognitive Tutor ITS as well

Background: Detecting Gaming

- Detecting gaming using algorithms on EDM data
  - Latent Response Model
    - Model was compared to Human Observations
    - Gamed-Hurt detections were good
    - Gamed-Not Hurt detections failed
  - Waikato Environment for Knowledge Analysis (WEKA)
    - Machine Learning package
    - Identified lack of gaming well (98%)
    - Identification of gaming was poor (19%)
- Detecting reasons for gaming
  - Poor ITS design
  - Model explained 56% of the variance
  - Unknown step (or Difficult step) is gamed the most
  - Boredom also leads to gaming

Background: Detecting Gaming

- Detecting gaming using Human Observation
  - Visual Inspection
  - Might make students uncomfortable
  - Ratio of observers to students
  - Consistency of the Human Observer
- Detecting gaming using “hand-labelling” of EDM data
  - Considers student actions in context
  - Labour intensive
  - Inconsistency in labelling
- Detecting gaming based on actions
  - Help Tutor
  - If-Then action rules designed by hand
  - Approach relies heavily on Psychological Experts

Background: Discouraging Gaming

- Delaying the hints
  - Discourages students from rushing through hints
  - Significantly reduced help requests
  - No impact on learning
  - Resulted in another form of gaming
- Supplementary exercises and animated agent
  - Disapproval on detecting gaming
  - Marginal impact on gaming
  - No impact on learning
- Visualizations
  - Of behaviour: Reduced gaming, impact on learning not assessed.
  - Of progress: Reduced gaming, improved learning

Background: Understanding Hints

- High level hints
  - Extract information
- Bottom-out hints
  - Provide information
  - Might not be harmful
- Study correlated bottom-out hints to learning
  - But no correlation to high level hints
- Human tutors are more adaptive and selective in hints
- Reinforced Learning
- Scaffolding Questions

Primary Data

- Pittsburgh Learning Center DataShop
- Andes ITS for assigned class homework and test prep
- Entries
  - Drawing diagrams
  - Equations
  - Any order
  - Provides feedback for correctness
- Request for hints
  - General information
  - Finally bottom-out hint
- Final score decremented for every bottom-out hint

The Gaming Detector

What if the student learns after copying?

Table 1 *Tutor-Student* turn pairs

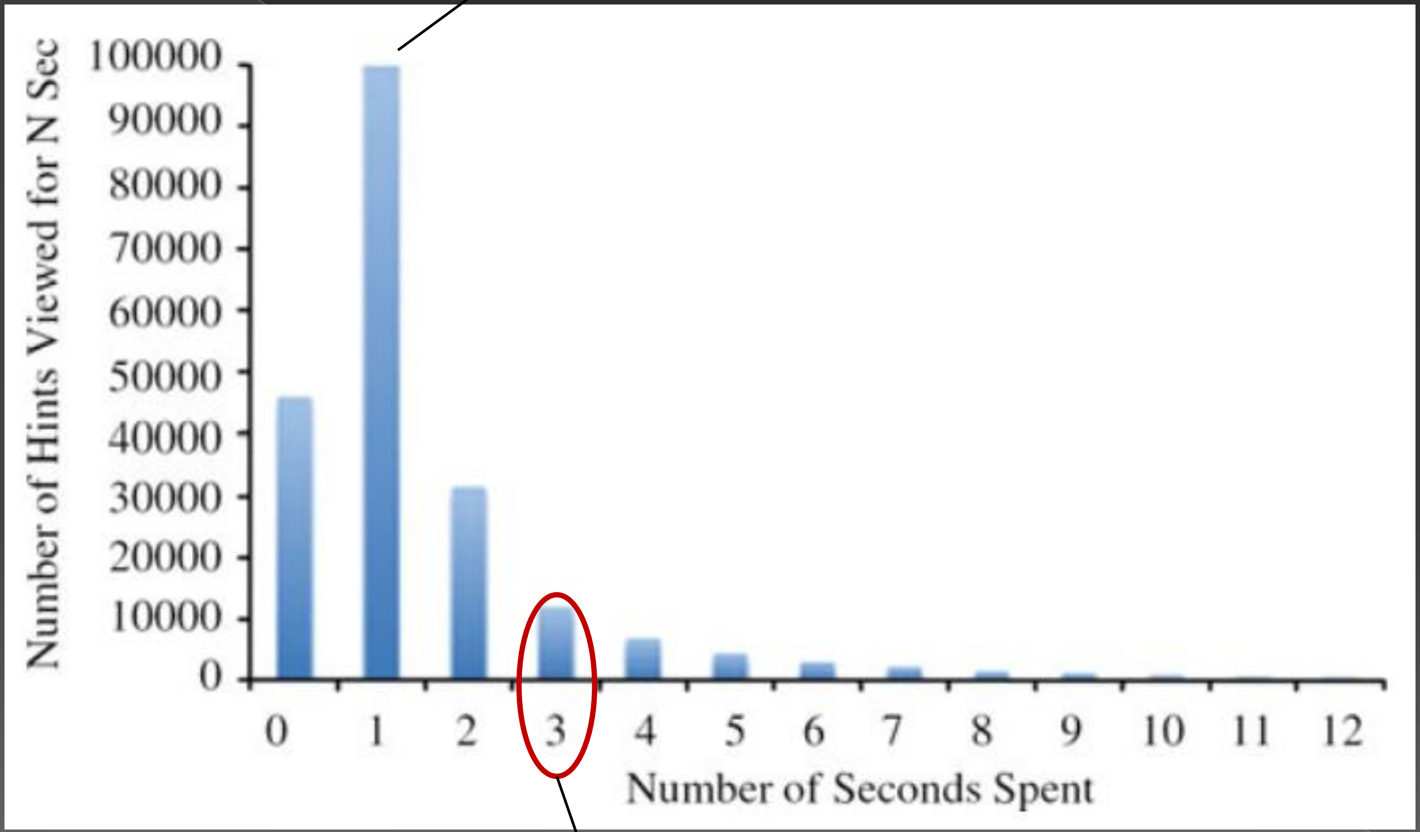
	(a) Student: hint request		(b) Student: entry		
	Fast	Slow	Fast		Slow
(i) Tutor: bottom-out hint	*(S)Skip hint*	–	*(C)Copy hint*		–
(ii) Tutor: high-level hint	*(S)Skip hint*	–	–		–
(iii) Tutor: incorrect (red)	–	–	*(G)Guess (red only)*	–	–
(iv) Tutor: correct (green)	*(P)No planning*	–	–		–

gamed cells *italicized*

Set of 318 unique problems and 286 students

Setting the time thresholds

Most students gamed?



Enough time to read?

Setting the time thresholds

Table 1 *Tutor-Student* turn pairs

	(a) Student: hint request		(b) Student: entry	
	Fast	Slow	Fast	Slow
(i) Tutor: bottom-out hint	*(S)Skip hint* 3s	–	*(C)Copy hint* 4s	–
(ii) Tutor: high-level hint	*(S)Skip hint* 3s	–	–	–
(iii) Tutor: incorrect (red)	–	–	*(G)Guess (red only)* –	–
(iv) Tutor: correct (green)	*(P)No planning* 4s	–	–	–

gamed cells *italicized*

Equations: 4s

Diagrams: 6s

Predictor of gaming

$\mathrm{PerGaming}_{sp}$	*percentage of gaming by a student s on a problem p*	(1)
$\sum_{p=1}^{p=N} \mathrm{perGaming}_p / N$	*average gaming by a student s across all N problems p solved by that student*	(2)
$\sum_{s=1}^{s=M} \mathrm{perGaming}_p / M$	*average gaming on a problem p across all M students s*	(3)

- Student or Problem?
- Problem is the unit of analysis
  - Might consider lesson (too large)
  - Tutor-student turn pair would be ideal (too small, difficult to equate)
- Rapid hint requests: Not lumped
  - If gaming is lumped, then non-gaming must be too
- Using a step
  - Asking for hints, incorrect attempts, correct entry
  - What if a single action in a step is gamed?

Different analyses: Linear Regression

Dependent Variable

Independent Variables

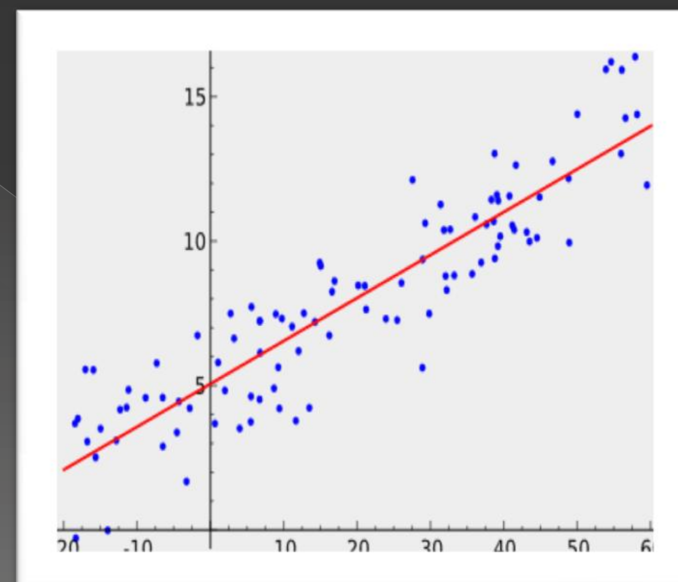
$\mathrm{PerGaming}_{sp}$	*percentage of gaming by a student s on a problem p*	(1)
$\sum_{p=1}^{p=N} \mathrm{perGaming}_{p}/N$	*average gaming by a student s across all N problems p solved by that student*	(2)
$\sum_{s=1}^{s=M} \mathrm{perGaming}_{p}/M$	*average gaming on a problem p across all M students s*	(3)

F=16915, $p < 0.001$, R2 = .608

Student: standardized coefficient=.658, t=52.7, $p < 0.001$

Problem: standardized coefficient= .325, t=74.23, $p < 0.001$

$$\mathrm{perGamed}_{s,p} = \mathrm{const} + .658\overline{\mathrm{student}_s} + .325\overline{\mathrm{problem}_p}$$



Different analyses: Self Correlation

- Randomly sub-divide students (or problems) into buckets
- Check for correlation between the buckets
- Student self-correlation
  - Two buckets
    - Randomly splitting problems solved by a given student
    - Storing in each bucket the average gaming
  - $r = .963$, $p < 0.001$
- Problem self-correlation
  - Two buckets
    - Randomly splitting students who solved a given problem
    - Storing in each bucket the average gaming
  - $r = .89$, $p < 0.001$

Different analyses: Self Correlation

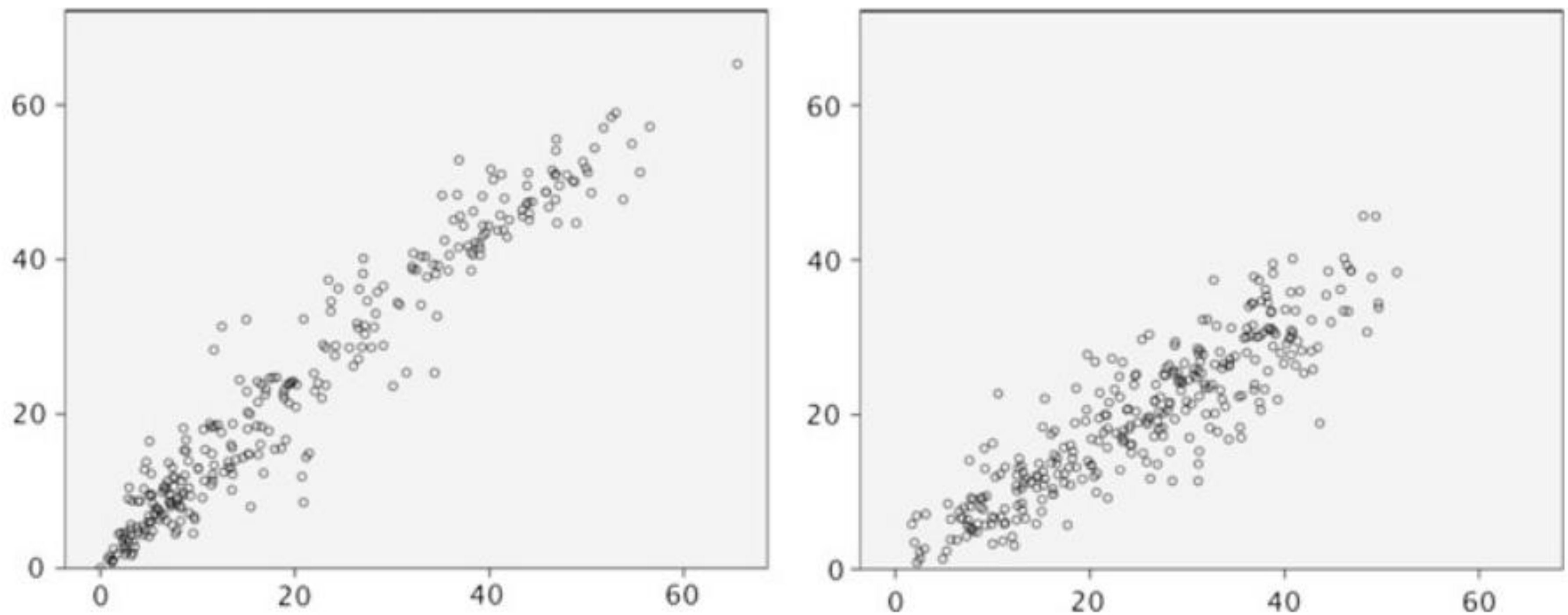


Fig. 2 Scatter plot for student (*left*) and problem (*right*) self correlations from the Andes data (percentage of gaming shown on X and Y axes)

Different analyses: Gaming frequency distributions

- How many students are high frequency gamers?
- How many problems are often gamed?
- Is gaming among students truly random? – Normal Distribution
  - Shapiro-Wilks test of normality: $W = .89$, $p = 0.02$
- Is gaming among problems truly random?
  - $W = .92$, $p > 0.05$

Different analyses: Gaming frequency distributions

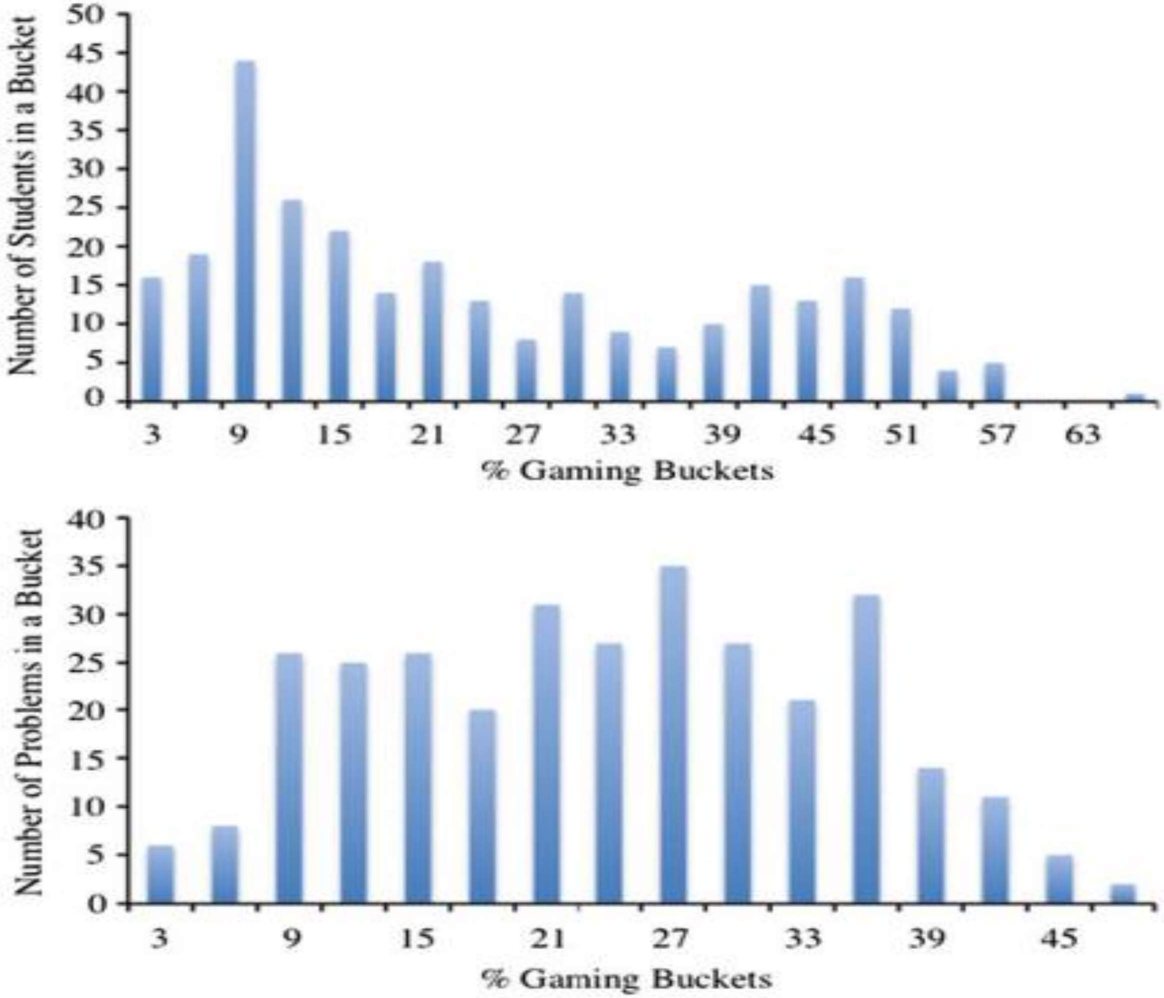


Fig. 3 Andes data on student (*top*) and problem (*bottom*) gaming distributions. Each bucket contains students (or problems) with a 3% gaming range (e.g., bucket 6 has 3% <gaming <6%)

Different analyses: Parameter learning

- Bayesian network parameter learning
  - ‘studentClass’: whether a student is a low or high gamer
  - ‘problemClass’: whether a problem is a low or high gamed problem
  - ‘gamed’: true if a tutor-student turn pair was gamed and false otherwise

- Classification:
  - Each student as a low or high gamer, based on a median split from

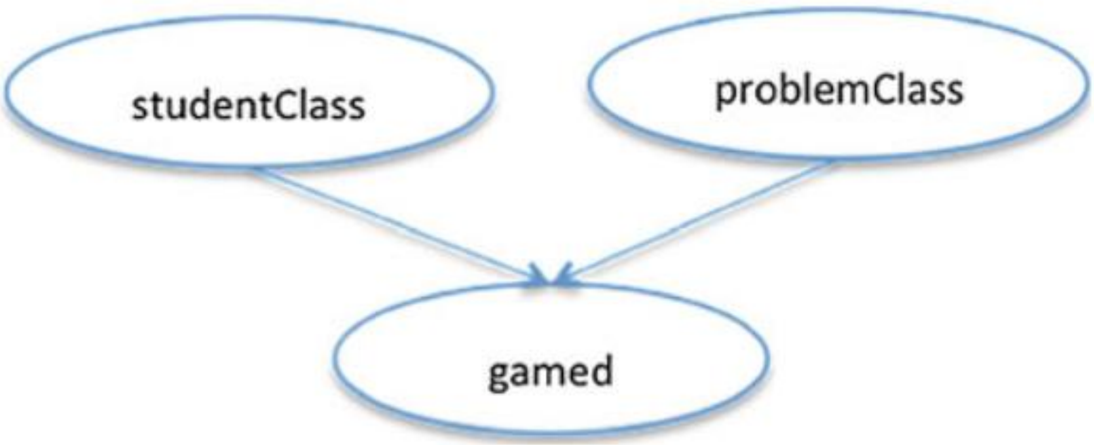
$$\sum_{p=1}^{p=N} \text{perGaming}_p / N \quad \textit{average gaming by a student s across all N problems p solved by that student}$$

  - Each problem as a low or high gamed problem based on a median split

$$\sum_{s=1}^{s=M} \text{perGaming}_p / M \quad \textit{average gaming on a problem p across all M students s}$$

- Netica
  - Learn the network parameters from the Andes data

Bayesian Network 1



P (gamed = *true* | problemClass = *low*, studentClass = *low*) = .10

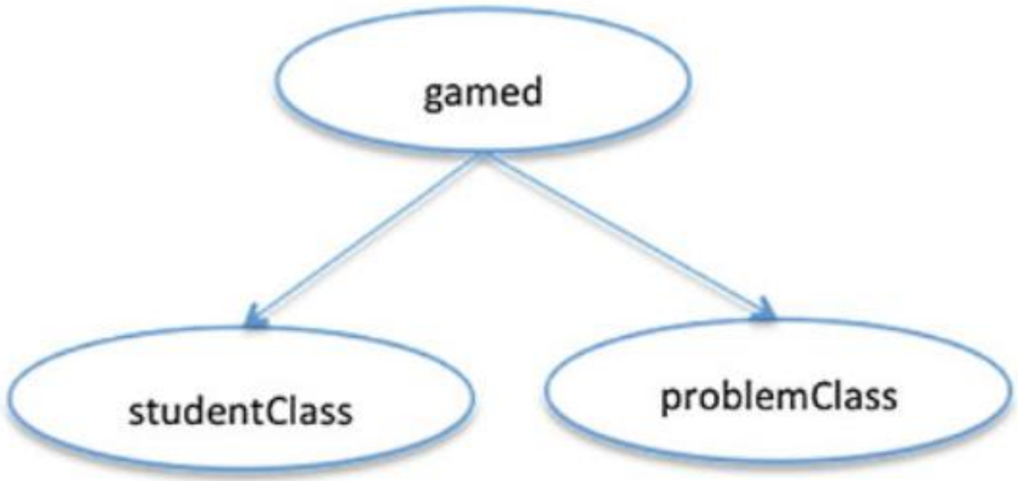
P (gamed = *true* | problemClass = *low*, studentClass = *high*) = .37

P (gamed = *true* | problemClass = *high*, studentClass = *low*) = .18

P (gamed = *true* | problemClass = *high*, studentClass = *high*) = .54

Fig. 4 Predictive Bayesian network (*top*) and corresponding parameters obtained from the Andes log data via the Netica counting algorithm (*bottom*). The parameters for P(gamed = false |. . .), not shown here, are simply 1 − P (gamed = true |. . .)

Bayesian Network 2



P (studentClass =*high* | gamed = *true*) = .86
P (studentClass =*high* | gamed = *false*) = .52
P (problemClass =*high* | gamed = *true*) = .77
P (problemClass =*high* | gamed = *false*) = .58

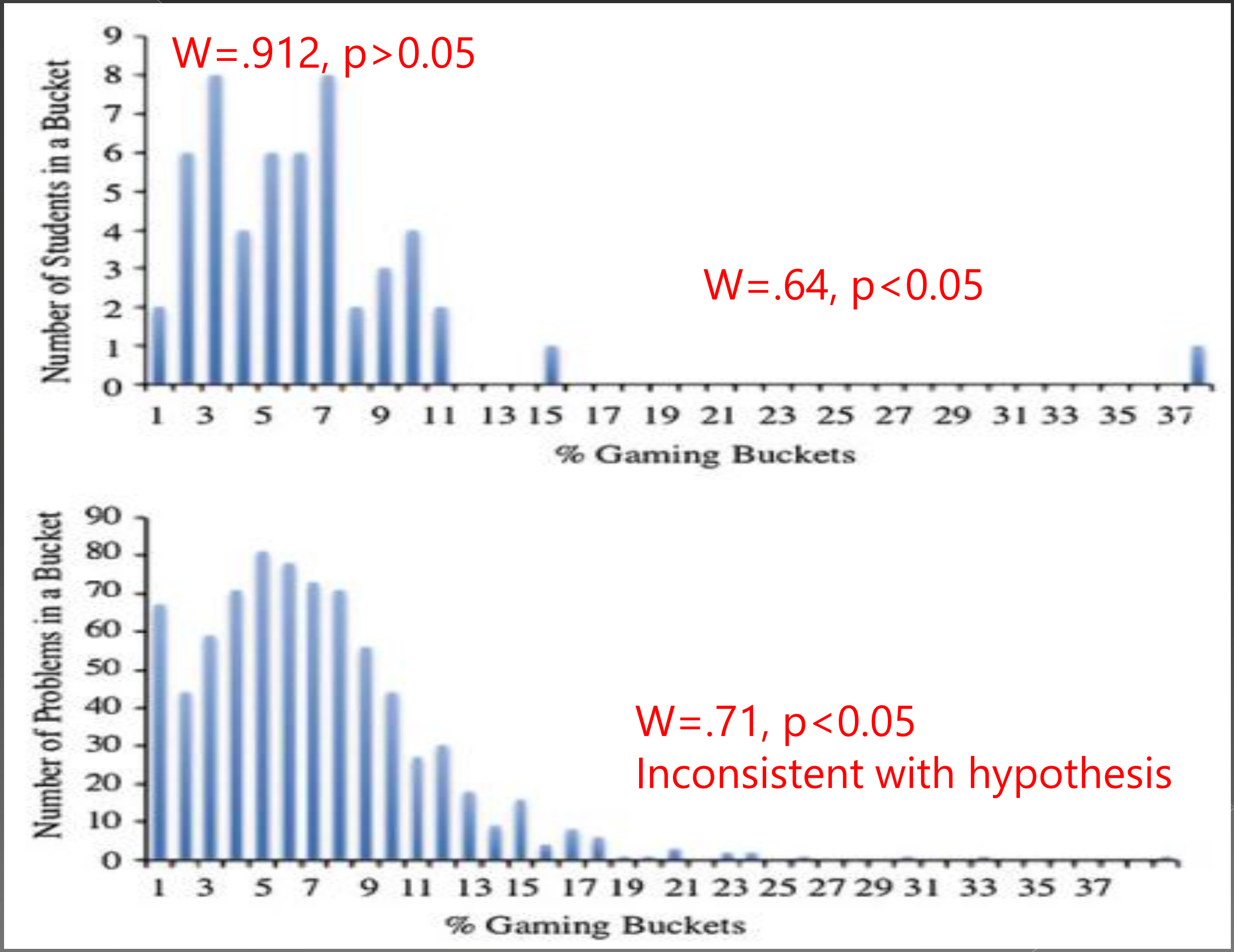
Fig. 5 Naïve Bayes classifier (*top*) and corresponding parameters obtained from the Andes log data via the Netica counting algorithm (*bottom*). The parameters for P([problem,student]Class = low |. . .), not shown here, are simply 1 − P ([problem, student]Class = high |. . .)

Gaming Predictor in Cognitive Tutor

- Method of solving problems: similar to Andes
- 53 individual students, 775 unique problems
- Linear Regression Analysis
  - $F = 4588$, $p < 0.001$, $R2 = .420$ (Andes: $F=16915$, $p < 0.001$, $R2 = .608$)
  - Student: standardized coefficient= .52, $t = 76$, $p < 0.001$ (Andes: standardized coefficient=.658, $t=52.7$, $p < 0.001$)
  - Problem: standardized coefficient= .35, $t = 51$, $p < 0.001$ (Andes: standardized coefficient= .325, $t=74.23$, $p < 0.001$
- Self Correlation Analysis
  - Student: $r = .973$, $p < 0.001$ (Andes: $r = .963$, $p < 0.001$)
  - Problem: $r = .55$, $p < 0.001$ (Andes: $r = .89$, $p < 0.001$)

Gaming Predictor in Cognitive Tutor

- Gaming frequency distributions



Gaming Predictor in Cognitive Tutor

- Bayesian Network

P (gamed = *true* | problemClass = *low*, studentClass = *low*) = .03

P (gamed = *true* | problemClass = *low*, studentClass = *high*) = .10

P (gamed = *true* | problemClass = *high*, studentClass = *low*) = .07

P (gamed = *true* | problemClass = *high*, studentClass = *high*) = .17

Gaming Profiles

- 22.5% of tutor-student turn pairs were gamed

	(a) Student: Hint Request		(b) Student: Entry	
	fast	slow	fast	slow
(1) Tutor: B-O Hint	*S: 0.02 (.3)*	.2 (2.1)	*C: 1.8 (23.6)*	5.7 (73.9)
(2) Tutor: H-L Hint	$\boxed{\textit{S:18.4 (58.6)}}$	5.8 (18.5)	.7 (2.3)	=100% 6.4 (20.6)
(3) Tutor: Incorrect	3.0 (12.4)	2.5 (10.3)	*G: 5.4 (22.1) (RED)* 4.8 (20) (GREEN)	8.7 (35.8)
(4) Tutor: Correct	*P: 5.3 (14.5)*	3.6 (10.0)	14.1 (38.9)	13.3 (36.6)

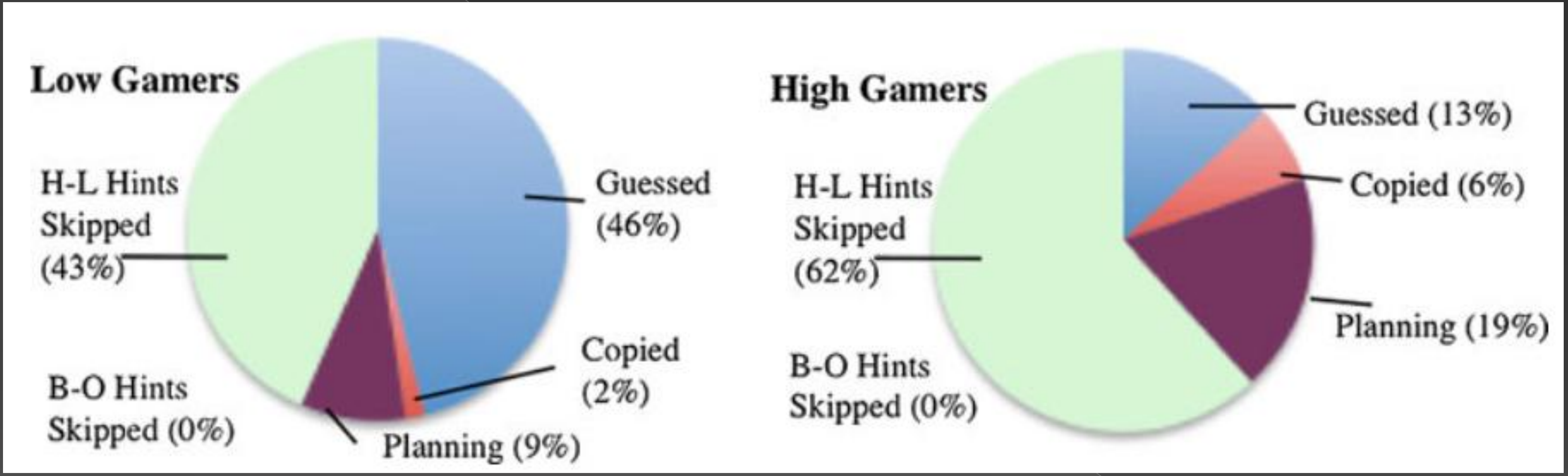
=100%

High Gamers and Low gamers

- Median split on the values of average gaming of student

		(a) Student: Hint Request		(b) Student: Entry		
		fast	slow	fast		slow
(1) Tutor: B-O Hint	LG	*S: 0.02*	0.07	*C: 0.34*		3.2
	HG	*S: .03*	.25	*C: 3.3*		8.3
(2) Tutor: H-L Hint	LG	*S: 7.1*	6.4	0.53		7.3
	HG	*S: 29.7*	5.1	0.92		5.7
(3) Tutor: Incorrect	LG	3.2	3.2	*G: 5.5 (RED)*	5.3 (GREEN)	11.4
	HG	2.8	1.8	*G: 5.2 (RED)*	4.2 (GREEN)	6.0
(4) Tutor: Correct	LG	*P: 1.5*	3.4	20.3		21.1
	HG	*P: 9.1*	3.9	8.1		5.5

High Gamers and Low gamers



Analyses of Hints: Hint Viewing

Average time spent on viewing hints
High-Level Hint: 5.7s
Bottom-out hint: 9.2s

	High Gamers	Low Gamers
Bottom-out hints	7.5s	10.9s
High level hints	3.2s	8.1s

Analyses of Hints: Basic analysis

	High Gamers	Low Gamers
Generate entry after high level hint	16%	36%
Generate entry after bottom-out hint	97%	97%

Generating correct entry

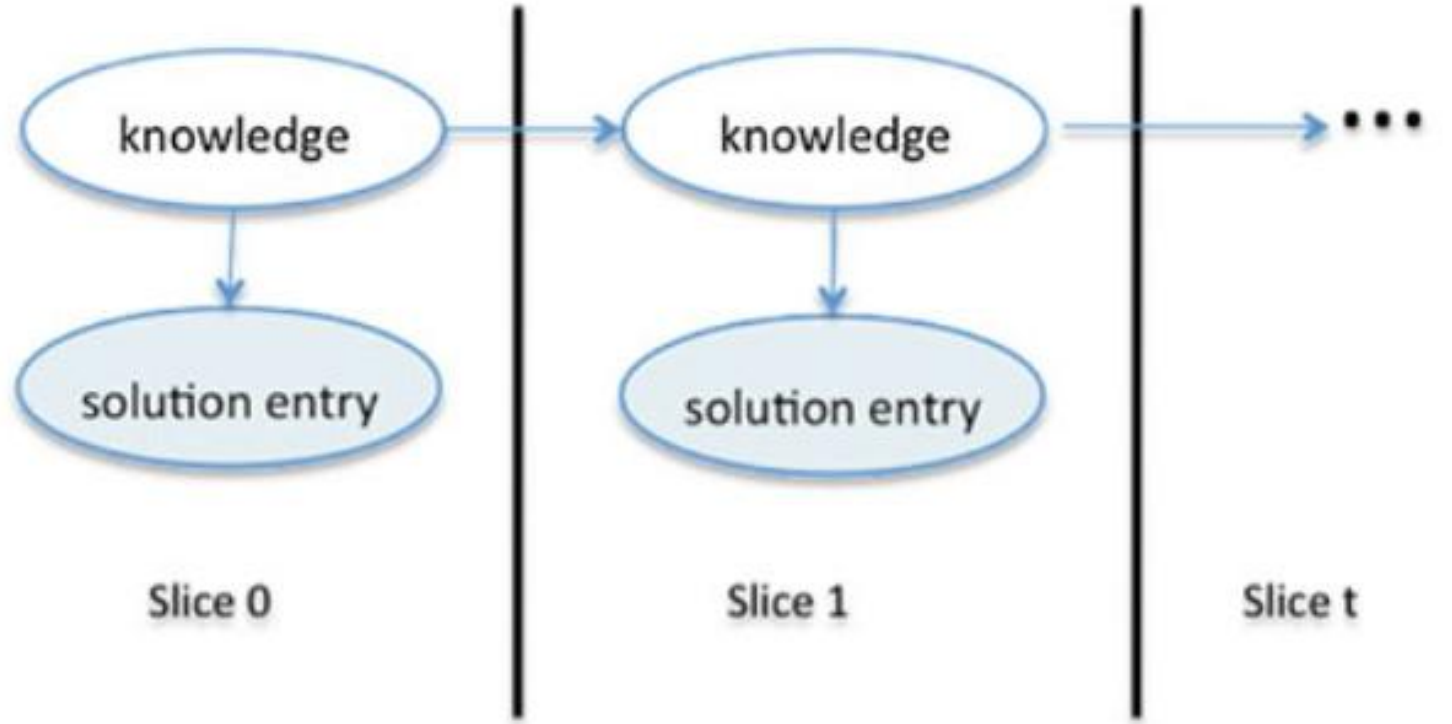
	High Gamers	Low Gamers
After high level hint (eventually)	73%	72%
After bottom-out hint	92%	89%

Analyses of Hints: Basic analysis

Generating correct entry

	High Gamers	Low Gamers
After bottom-out hint (attempts)	1.19 attempts	1.23 attempts
After bottom-out hint (Time)	23s	34s
After high level hint (attempts)	2.01 attempts	1.66 attempts
After high level hint (Time)	28s	37s

Dynamic Bayesian Model
Basic Model



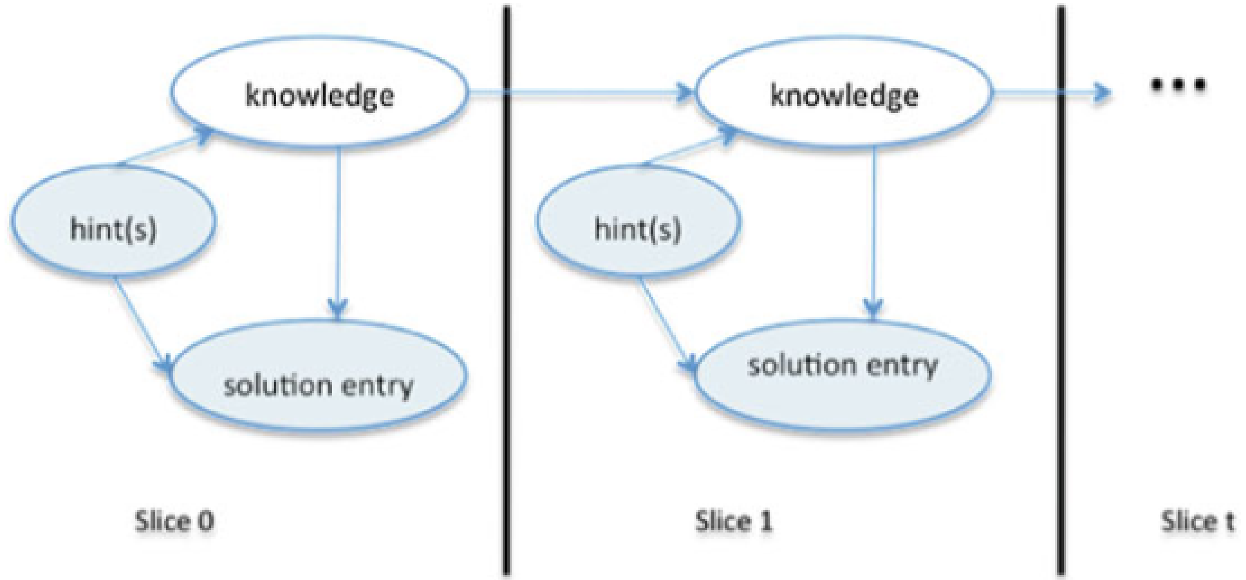
guess: P(solution entry $_t$ = *correct* | knowledge $_t$ = *unmastered*) = 0.33

slip: P(solution entry $_t$ = *incorrect* | knowledge $_t$ = *mastered*) = 0.15

learn: P(knowledge $_t$ = *mastered* | knowledge $_{t-1}$ = *unmastered*) = 0.25

forget: P(knowledge $_t$ = *unmastered* | knowledge $_{t-1}$ = *mastered*) = 0.06

Dynamic Bayesian Model
Help Model



scaffold: P(solution entry $_t$ = *correct* | knowledge $_t$ = *unmastered*, hint(s) $_t$ = *true*) = 0.38

guess: P(solution entry $_t$ = *correct* | knowledge $_t$ = *unmastered*, hint(s) $_t$ = *false*) = 0.31

slipHint(s): P(solution entry $_t$ = *incorrect* | knowledge $_t$ = *mastered,* hint(s) $_t$ = *true*) = 0.17

slipNoHint(s): P(solution entry $_t$ = *incorrect* | knowledge $_t$ = *mastered* , hint(s) $_t$ = *false*) = 0.16

learnHint(s): P(knowledge $_t$ = *mastered* | knowledge $_{t-1}$ = *unmastered* , hint(s) $_t$ = *true*) = 0.21

learnNoHint(s): P(knowledge $_t$ = *mastered* | knowledge $_{t-1}$ = *unmastered,* hint(s) $_t$ = *false*) = 0.23

forgetHint(s): P(knowledge $_t$ = *unmastered* | knowledge $_{t-1}$ = *mastered,* hint(s) $_t$ = *true*) = 0.05

forgetNoHint(s): P(knowledge $_t$ = *unmastered* | knowledge $_{t-1}$ = *mastered,* hint(s) $_t$ = *false*) = 0.04

Dynamic Bayesian Model
Help Model

scaffold-HL: P(solution entry $_t$ =*correct* | knowledge $_t$ = *unmastered*, hint(s) $_t$ =*high-level*) = 0.21

scaffold-BO: P(solution entry $_t$ =*correct* | knowledge $_t$ = *unmastered*, hint(s) $_t$ =*bottom-out*) = 0.49

guess: P(solution entry $_t$ = *correct* | knowledge $_t$ = *unmastered* , hint(s) $_t$ =*none*) = 0.25

slipWith-HL-Hint(s): P(solution entry $_t$ =*incorrect* | knowledge $_t$ = *mastered*, hint(s) $_t$ =*high-level*) = 0.32

slipWith-BO-Hint(s): P(solution entry $_t$ =*incorrect* | knowledge $_t$ = *mastered*, hint(s) $_t$ =*bottom-out*) = 0.13

slipNoHint(s): P(solution entry $_t$ =*incorrect* | knowledge $_t$ = *mastered*, hint(s) $_t$ =*none*) = 0.17

learnWith-HL-Hint(s): P(knowledge $_t$ = *mastered* | knowledge $_{t-1}$ =*unmastered*, hint(s) $_t$ =*high-level*) = 0.25

learnWith-BO-Hint(s): P(knowledge $_t$ = *mastered* | knowledge $_{t-1}$ =*unmastered*, hint(s) $_t$ =*bottom-out*) = 0.28

learnNoHint(s): P(knowledge $_t$ = *mastered* | knowledge $_{t-1}$ =*unmastered* , hint(s) $_t$ =*none*) = 0.23

forgetWith-HL-Hint(s): P(knowledge $_t$ =*unmastered* | knowledge $_{t-1}$ =*mastered*, hint(s) $_t$ =*high-level*) = 0.08

forgetWith-BO-Hint(s): P(knowledge $_t$ =*unmastered* | knowledge $_{t-1}$ =*mastered*, hint(s) $_t$ =*bottom-out*) = 0.05

forgetNoHint(s): P(knowledge $_t$ = *unmastered* | knowledge $_{t-1}$ = *mastered* , hint(s) $_t$ =*none*) = 0.03

Impact of gaming
DBN with 2-valued hints node

Low Gamers (2-valued 'hint(s)' node):

scaffold: P(solution entry $_t$ =*correct* | knowledge $_t$ = *unmastered*, hint(s) $_t$ =*true*) = 0.37

guess: P(solution entry $_t$ = *correct* | knowledge $_t$ = *unmastered* , hint(s) $_t$ =*false*) = 0.27

slipWithHint(s): P(solution entry $_t$ =*incorrect* | knowledge $_t$ = *mastered,* hint(s) $_t$ =*true*) = 0.22

slipNoHint(s): P(solution entry $_t$ =*incorrect* | knowledge $_t$ = *mastered,* hint(s) $_t$ =*false*) = 0.15

***learnWithHint(s):* P(knowledge $_t$ = *mastered* | knowledge $_{t-1}$ =*unmastered,* hint(s)=*true*) = 0.26**

***learnNoHint(s):* P(knowledge $_t$ = *mastered* | knowledge $_{t-1}$ =*unmastered* , hint(s)=*false*) = 0.22**

High Gamers (2-valued 'hint(s)' node):

scaffold: P(solution entry $_t$ =*correct* | knowledge $_t$ = *unmastered*, hint(s) $_t$ =*true*) = 0.39

guess: P(solution entry $_t$ = *correct* | knowledge $_t$ = *unmastered* , hint(s) $_t$ =*false*) = 0.32

slipWithHint(s): P(solution entry $_t$ =*incorrect* | knowledge $_t$ = *mastered,* hint(s) $_t$ =*true*) = 0.16

slipNoHint(s): P(solution entry $_t$ =*incorrect* | knowledge $_t$ = *mastered,* hint(s) $_t$ =*false*) = 0.18

***learnWithHint(s):* P(knowledge $_t$ = *mastered* | knowledge $_{t-1}$ =*unmastered,* hint(s)=*true*) = 0.17**

***learnNoHint(s):* P(knowledge $_t$ = *mastered* | knowledge $_{t-1}$ =*unmastered* , hint(s)=*false*) = 0.19**

Impact of gaming
DBN with 3-valued hints node

Low Gamers (3-valued 'hint(s)' node):

scaffold-HL: P(solution entry $_t$ =*correct* | knowledge $_t$ = *unmastered*, hint(s) $_t$ =*high-level*) = 0.27

scaffold-BO: P(solution entry $_t$ =*correct* | knowledge $_t$ = *unmastered*, hint(s) $_t$ =*bottom-out*) = 0.43

guess: P(solution entry $_t$ = *correct* | knowledge $_t$ = *unmastered* , hint(s) $_t$ =*none*) = 0.25

slipWith-HL-Hint(s): P(solution entry $_t$ =*incorrect* | knowledge $_t$ = *mastered*, hint(s) $_t$ =*high-level*) = 0.26

slipWith-BO-Hint(s): P(solution entry $_t$ =*incorrect* | knowledge $_t$ = *mastered*, hint(s) $_t$ =*bottom-out*) = 0.14

slipNoHint(s): P(solution entry $_t$ =*incorrect* | knowledge $_t$ = *mastered*, hint(s) $_t$ =*none*) = 0.15

learnWith-HL-Hint(s): **P(knowledge $_t$ = *mastered* | knowledge $_{t-1}$ =*unmastered*, hint(s)=*high-level*) = 0.31**

learnWith-BO-Hint(s): **P(knowledge $_t$ = *mastered* | knowledge $_{t-1}$ =*unmastered*, hint(s)=*bottom-out*) = 0.41**

learnNoHint(s): **P(knowledge $_t$ = *mastered* | knowledge $_{t-1}$ =*unmastered* , hint(s)=*none*) = 0.22**

High Gamers (3-valued 'hint(s)' node):

scaffold-HL: P(solution entry $_t$ entry =*correct* | knowledge $_t$ = *unmastered*, hint(s)=*high-level*) = 0.22

scaffold-BO: P(solution entry $_t$ =*correct* | knowledge $_t$ = *unmastered*, hint(s)=*bottom-out*) = 0.46

guess: P(solution entry $_t$ = *correct* | knowledge $_t$ = *unmastered* , hint(s)=*none*) = 0.26

slipWith-HL-Hint(s): P(solution entry $_t$ =*incorrect* | knowledge $_t$ = *mastered*, hint(s)=*high-level*) = 0.37

slipWith-BO-Hint(s): P(solution entry $_t$ =*incorrect* | knowledge $_t$ = *mastered*, hint(s)=*bottom-out*) = 0.14

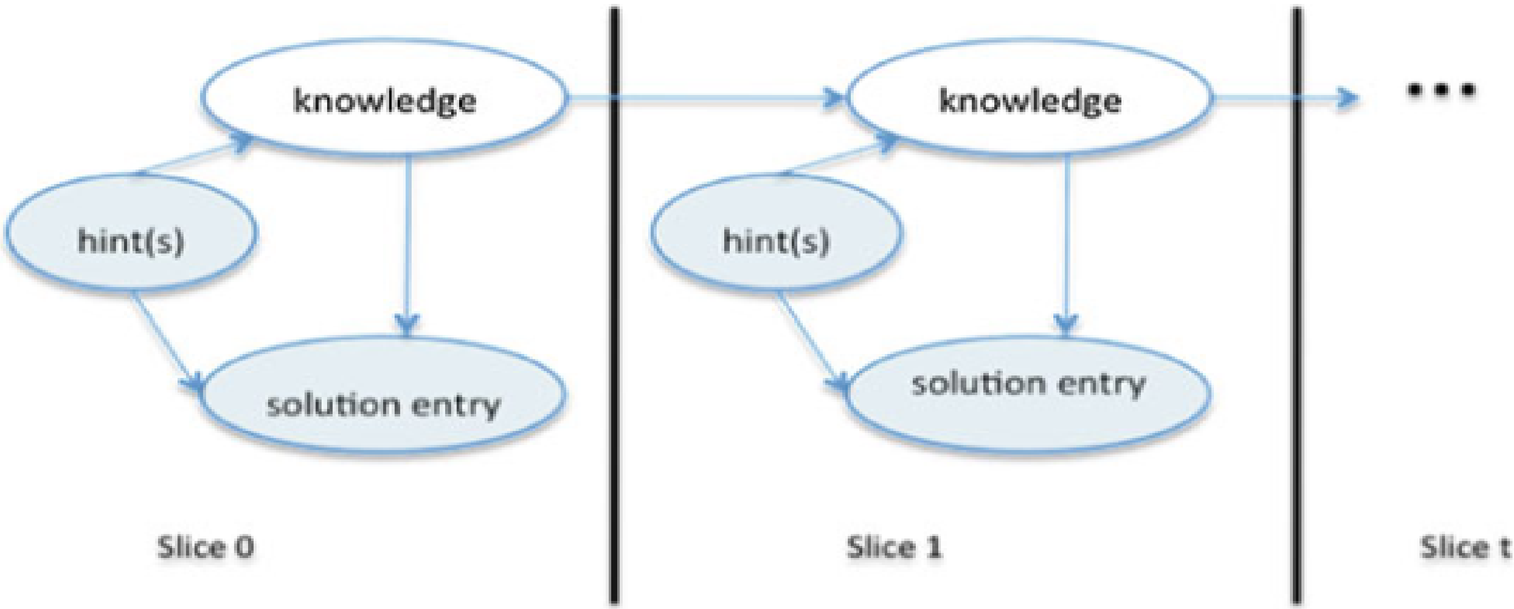
slipNoHint(s): P(solution entry $_t$ =*incorrect* | knowledge $_t$ = *mastered*, hint(s)=*none*) = 0.21

learnWith-HL-Hint(s): **P(knowledge $_t$ = *mastered* | knowledge $_{t-1}$ =*unmastered*, hint(s)=*high-level*) = 0.20**

learnWith-BO-Hint(s): **P(knowledge $_t$ = *mastered* | knowledge $_{t-1}$ =*unmastered*, hint(s)=*bottom-out*) = 0.25**

learnNoHint(s): **P(knowledge $_t$ = *mastered* | knowledge $_{t-1}$ =*unmastered* , hint(s)=*none*) = 0.20**

Impact on learning



guessWithGaming: P(solution entry $_t$ =*correct* | knowledge $_t$ = *unmastered*, gaming $_t$ =*true*) = 0.47

guessNoGaming: P(solution entry $_t$ =*correct* | knowledge $_t$ = *unmastered*, gaming $_t$ =*false*) = 0.47

slipWithGaming: P(solution entry $_t$ =*incorrect* | knowledge $_t$ = *mastered,* gaming $_t$ =*true*) = 0.26

slipNoGaming: P(solution entry $_t$ =*incorrect* | knowledge $_t$ = *mastered,* gaming $_t$ =*false*) = 0.15

learnWithGaming: **P(knowledge $_t$ =*mastered* | knowledge $_{t-1}$ =*unmastered,* gaming=*true*) = 0.19**

learnNoGaming: **P(knowledge $_{t-}$=*mastered* | knowledge $_{t-1}$=*unmastered,* gaming=*false*) = 0.33**

forgetWithGaming: P(knowledge $_t$ =*unmastered* | knowledge $_{t-1}$ =*mastered,* gaming=*true*) = 0.06

forgetNoGaming: P(knowledge $_t$ = *unmastered* | knowledge $_{t-1}$ =*mastered* , gaming=*false*) = 0.06

Conclusion

Thank you!