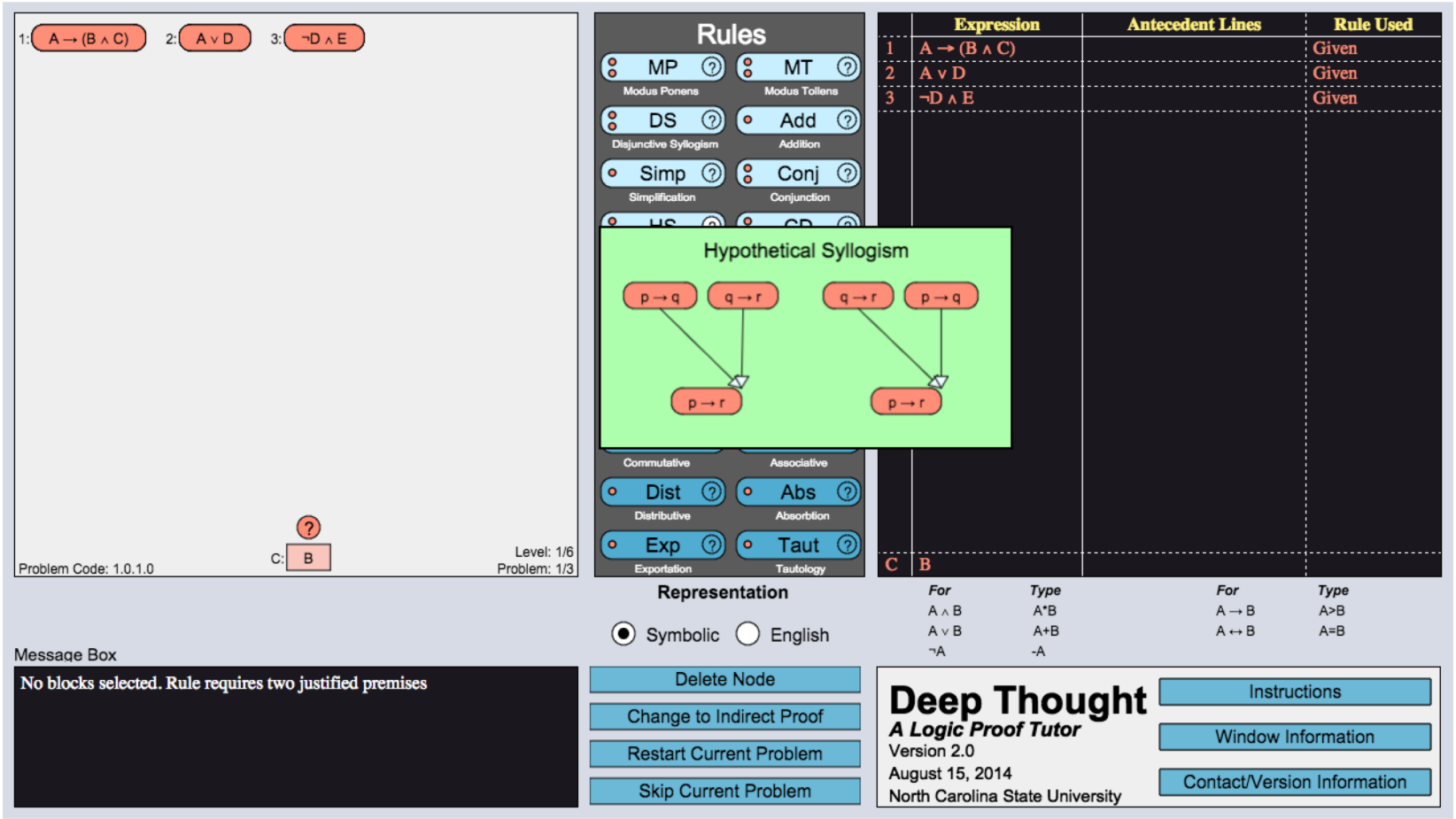
Assigned Project2

Song Ju, Shitian Shen, Min Chi
Department of Computer Science
North Carolina State University

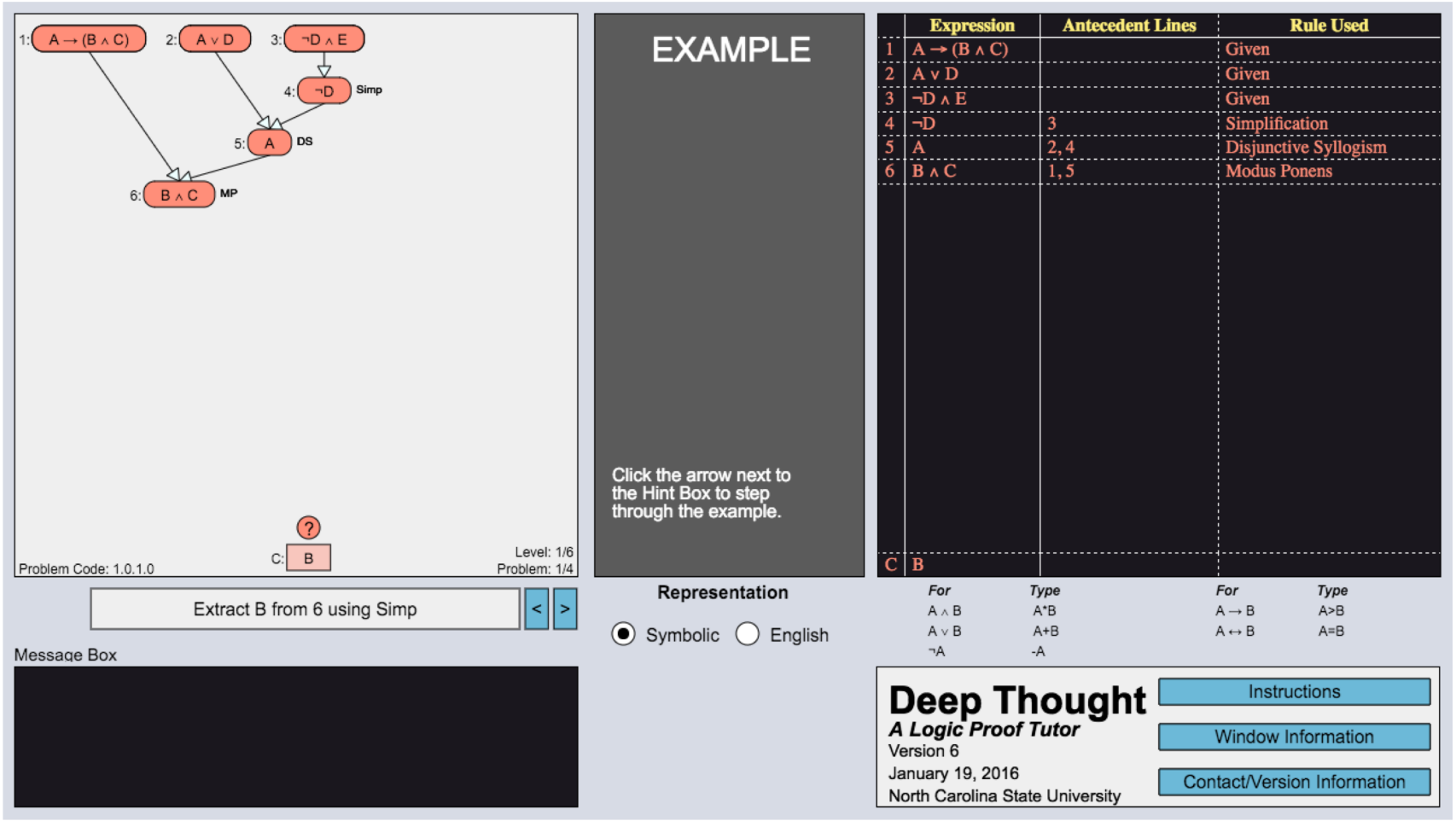
Introduction

- Intelligent Tutoring System contains a set of actions
- Deep Thought (Dr. Barnes, 2015) can take two actions:
  - *Problem Solving* (PS)
  - *Work Example* (WE)

Problem Solving



Work Example



Question

When to assign PS or WE to students ?

Pedagogical Strategy:

- Policy that decide what action to take next in the face of alternatives

Induce Pedagogical Strategy

- Inducing pedagogical strategy is challenging
  - Hard code (ineffective and inefficient)
  - Data driven

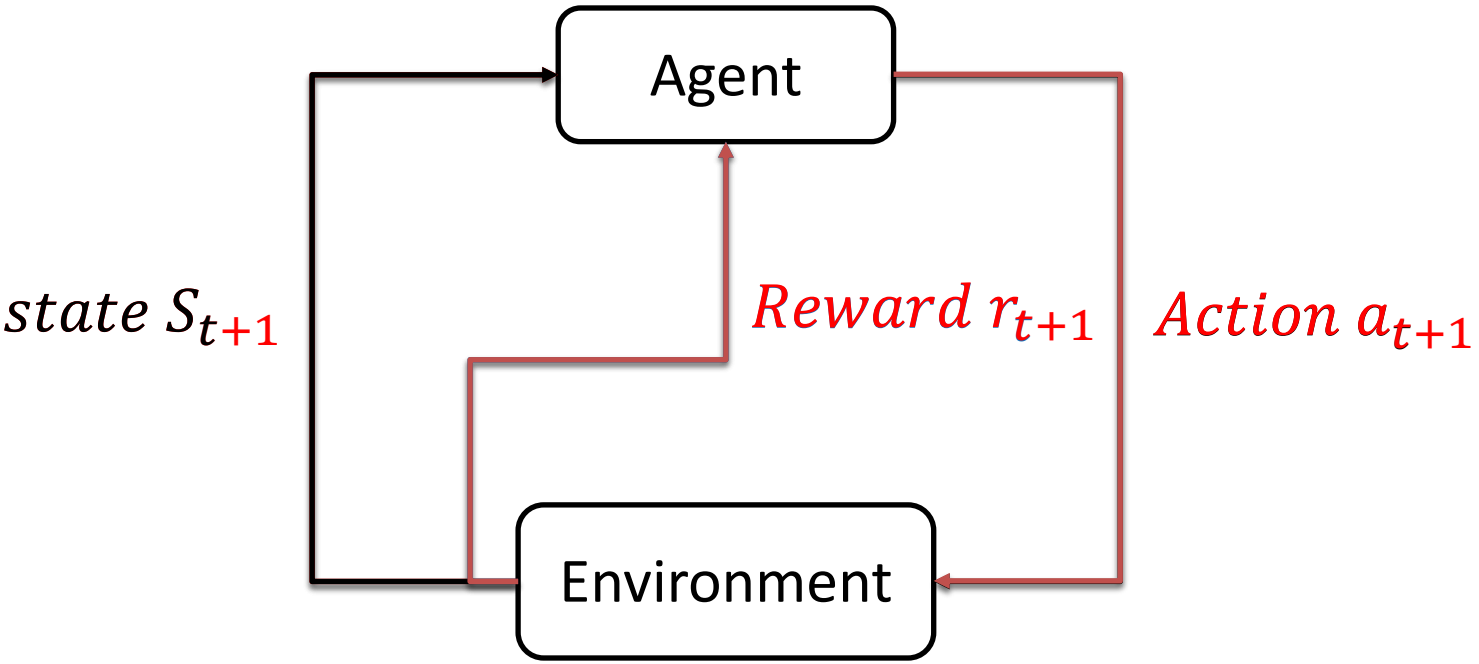
Reinforcement Learning vs. Inducing Pedagogical strategy

What is the best action for the agent (tutor)
to take in any state (learning context)
in order to maximize reward (student learning)

Reinforcement Learning:

- Model-based vs Model-free Reinforcement Learning
  - Model-based
    - Generating data is expensive (ITS)
    - Learn from the model instead of data sets
  - Model-free
    - Collecting data is trivial (playing chess)
    - Learn from data sets directly

Agent Environment Interaction



Markov Decision Process (MDP)

- A mathematical framework for representing a reinforcement learning task
- A tuple $< S, A, T, R, \pi >$

State Set	$S = \{s_1, s_2, \dots, s_n\}$
Action Set	$A = \{a_1, a_2, \dots, a_k\}$
Transition Probability	$T_{ij}^{a_k} = P(S_{t+1} = s_j \mid S_t = s_i, A_t = a_k)$
Rewards	$R_{ij}^{a_k} = \mathrm{E}(r_t \mid S_{t+1} = s_j, S_t = s_i, A_t = a_k)$
Policy	$\pi: S \rightarrow A$

Value Iteration: Algorithm

1. $V_0(s) = 0,\ for\ s\ \in$ S

Initialization

2. For k

   $\Delta \leftarrow 0$

   For each $s\ \in$ S

   $v \leftarrow V_{k-1}(s)$

   $V_k(s) \leftarrow \max_a \sum_{s'} T^a_{ss'}[R^a_{ss'} + \gamma V_{k-1}(s')]$

   $\Delta \leftarrow \max(\Delta, |v - V_k(s)|)$

   Until $\Delta \leftarrow \theta$ (a small positive number)

Maximizing Value Function

3. $\pi(s) = \ arg\max_a \sum_{s'} T^a_{ss'}[R^a_{ss'} + \gamma V^{\pi}(s')]$

Policy Generation

Value Iteration: Example

- Transfer Data into trajectories
  - State set : $\{S_1, S_2\}$
  - Action set: $\{PS,\ WE\}$

$$S_1 \xrightarrow{PS,0} S_2 \xrightarrow{PS,0} S_2 \xrightarrow{WE,50} S_1 \xrightarrow{PS,0} \dots \xrightarrow{WE,0} S_2 \xrightarrow{PS,100} S_1 \xrightarrow{WE,0}$$
$$S_2 \xrightarrow{PS,0} S_2 \xrightarrow{WE,50} S_1 \xrightarrow{PS,0} \dots \xrightarrow{WE,0} S_2 \xrightarrow{PS,0} S_1 \xrightarrow{WE,-50} S_1 \xrightarrow{WE,0}$$
$$S_2 \xrightarrow{PS,100} S_1 \xrightarrow{WE,0} S_2 \xrightarrow{PS,100} \dots \xrightarrow{WE,0} S_1 \xrightarrow{PS,0} S_2 \xrightarrow{WE,0} S_2 \xrightarrow{WE,0} T$$

Value Iteration: Example

- Transition Probability

$$P(S_1|S_2, PS) = \frac{\#(S_2 \overset{PS}{\rightarrow} S_1)}{\#(S_2 \overset{PS}{\rightarrow} S_1) + \#(S_2 \overset{PS}{\rightarrow} S_2)} = \frac{1}{4}$$

- Expected Rewards

$$R(S_1|S_2, PS) = \frac{\sum r(S_2 \overset{PS}{\rightarrow} S_1)}{\#(S_2 \overset{PS}{\rightarrow} S_1)} = 20$$

Value Iteration: Example

- Transition probability $T^a_{ss'}$

PS	S_1	S_2
S_1	1/4	3/4
S_2	1/2	1/2

WE	S_1	S_2
S_1	1/2	1/2
S_2	2/3	1/3

- Reward function $R^a_{ss'}$

PS	S_1	S_2
S_1	10	40
S_2	20	30

WE	S_1	S_2
S_1	20	30
S_2	45	5

Value Iteration: Example

K	$V(S_1)$		$V(S_2)$	
0	0		0	
1	32.50	PS	31.67	WE
2	61.18	PS	60.67	WE
3	87.22	PS	86.58	WE
4	110.56	PS	109.97	WE
⋮	⋮		⋮	
121	320.90	PS	320.30	WE
122	320.90	PS	320.30	WE

$$V_1(S_1) = max\begin{cases}\frac{1}{4}(10 + 0.9 * 0) + \frac{3}{4}(40 + 0.9 * 0) = 32.50 & PS \\ 1\end{cases}$$

$$V_1(S_2) = max\begin{cases}\frac{1}{2}(20 + 0.9 * 0) + \frac{1}{2}(30 + 0.9 * 0) = 25 & PS \\ \frac{2}{3}(45 + 0.9 * 0) + \frac{1}{3}(5 + 0.9 * 0) = 31.67 & WE\end{cases}$$

$$V_2(S_1) = max\begin{cases}\frac{1}{4}(10 + 0.9 * 32.5) + \frac{3}{4}(40 + 0.9 * 31.67) = 61.18 & PS \\ 1\end{cases}$$

$$V_2(S_2) = max\begin{cases}\frac{1}{2}(20 + 0.9 * 32.5) + \frac{1}{2}(30 + 0.9 * 31.67) = 53.87 & PS \\ \frac{2}{3}(45 + 0.9 * 32.5) + \frac{1}{3}(5 + 0.9 * 31.67) = 60.67 & WE\end{cases}$$

Optimal policy π^*:

$$S_1 \rightarrow PS$$
$$S_2 \rightarrow WE$$

Policy Evaluation: ECR

- Expected Cumulative Reward (Tetreault, 2006)

$$ECR = \sum_{i=1}^{m} \frac{N_i}{N_1 + N_2 + \cdots + N_m} \times V^{\pi}(S_i)$$

Where S_i is the starting state, N_i is the times that S_i exists as starting state

- The higher ECR of the policy means the better policy

Policy Evaluation: IS

- **Importance Sampling (IS)-Based Evaluation Metrics:**

  (Precup, 2000; Thomas, 2015)

  - Importance Sampling (IS)
  - Weighted Importance Sampling (WIS)
  - Per-Decision Importance Sampling (PDIS)

Policy Evaluation: IS

- Student-ITS Interaction Trajectory:

$$H = s_1 \xrightarrow{a_1, r_1} s_2 \xrightarrow{a_2, r_2} s_3 \xrightarrow{a_3, r_3} \cdots s_L$$

$$H^L = (s_1, a_1, r_1, s_1, a_1, r_1, \dots, s_L, a_L, r_L)$$

- Return of trajectory:

$$G(H^L) = \sum_{t=1}^{L} \gamma^{t-1} r_t$$

- Historical Dataset:

$$D = \{H_1, H_2, H_3, \dots, H_n\}$$

- Behavior Policy: π_b
- Target Policy: π_e

Policy Evaluation: IS

- Probability of a trajectory H^L under π:

$$P_r(H^L, \pi) = P_r(s_1)\pi(a_1|s_1)P_r(s_2|s_1, a_1)\pi(a_2|s_2) \dots P_r(s_L|s_{L-1}, a_{L-1})$$

$$= P_r(s_1) \prod_{t=1}^{L} \pi(a_t|s_t)P_r(s_{t+1}|s_t, a_t)$$

- Importance Weight of the Trajectory:

$$W = \frac{P_r(H^L, \pi_e)}{P_r(H^L, \pi_b)} = \frac{P_r(s_1) \prod_{t=1}^{L} \pi_e(a_t|s_t)P_r(s_{t+1}|s_t, a_t)}{P_r(s_1) \prod_{t=1}^{L} \pi_b(a_t|s_t)P_r(s_{t+1}|s_t, a_t)} = \prod_{t=1}^{L} \frac{\pi_e(a_t|s_t)}{\pi_b(a_t|s_t)}$$

- Importance Sampling:

$$IS(\pi_e|H^L, \pi_b) = W \cdot G(H^L) = \left(\prod_{t=1}^{L} \frac{\pi_e(a_t|s_t)}{\pi_b(a_t|s_t)}\right) \cdot \left(\sum_{t=1}^{L} \gamma^{t-1} r_t\right)$$

Policy Evaluation: IS

- Estimator for single trajectory:

$$IS(\pi_e|H^L,\pi_b)=\left[\left(\prod_{t=1}^{L}\frac{\pi_e(a_t|s_t)}{\pi_b(a_t|s_t)}\right)\left(\sum_{t=1}^{L}\gamma^{t-1}r_t\right)\right]$$

- Estimator for the whole dataset:

$$IS(\pi_e|D)=\frac{1}{n_D}\sum_{i=1}^{n_D}\left[\left(\prod_{t=1}^{L^i}\frac{\pi_e(a_t^i|s_t^i)}{\pi_b(a_t^i|s_t^i)}\right)\left(\sum_{t=1}^{L^i}\gamma^{t-1}r_t^i\right)\right]$$

The Challenge is:

What is the best action for the agent (tutor)
to take in any state (learning context)
in order to maximize reward (student learning)

How to design states representing environment ?

State Representation: Feature Selection for RL

- Three types of feature selection methods
  - Filtered approach
    - Feature Selection process is independent to model construction
    - Evaluating the independence between reward with feature (Hirotaka, Masashi 2010)
  - Wrapper approach
    - Feature subsets are evaluated by predefined score function
    - Monte Carlo tree search algorithm (Gaudel 2010)
  - Embedded approach
    - Feature selection and model construction are executed simultaneously
    - Least Square Temporal Difference with lasso regularized item (Kolter 2009)

Previous Research: Correlation-based Methods: High vs Low

- When selecting features, should we select the feature that is most correlated (High) or uncorrelated (Low) to current optimal feature set ?
- In Supervised Learning, the feature with high correlation with labels are selected
- In Reinforcement Learning, the answer is not straightforward

Research Question: High vs Low

- Choosing most correlated features (High)
  - Most likely to be related to decision making
  - May not make more contribute than current optimal feature set
- Choosing most uncorrelated features (Low)
  - Raise the diversity of feature set
  - Take the risk of involving irrelevant or noisy features

Correlation Metrics

- Chi-square (CHI) (Zibran, 2007)

$$\chi^2 = \sum_i \frac{(X_i - Y_i)^2}{Y_i}$$

- Information Gain (IG) (C. Lee, 2010)

$$IG(X, Y) = H(Y) - H(X|Y)$$

Correlation Metrics

- Information Gain Ratio (IGR) (J. T. Kent, 1983)

$$IGR(X,Y) = \frac{H(X) - H(X|Y)}{H(Y)}$$

- Symmetric Uncertainty (SU) (L. Yu, H. Liu, 2003)

$$SU(X,Y) = \frac{H(X) - H(X|Y)}{H(X) + H(Y)}$$

- Weighted Information Gain (WIG) (We proposed)

$$WIG(X,Y) = \frac{H(X) - H(X|Y)}{(H(X) + H(Y))H(Y)}$$

10 Correlation-based Methods

- Explore both high and low correlation
- Obtain 10 correlation-based feature selection methods (5 correlation metrics × 2 correlation types)

	High	Low
CHI	CHI-High	CHI-Low
IG	IG-High	IG-Low
IGR	IGR-High	IGR-Low
SU	SU-High	SU-Low
WIG	WIG-High	WIG-Low

Other Implemented Methods

- RLPreviousFS (M Chi, 2011)
  - 4 RL based methods
  - 2 PCA based methods
  - 4 PCA & RL based methods
- Ensemble Methods
  - 10 correlation-based methods
  - 4 RL based methods

Intelligent Tutoring System

- Deep Thought (Dr. Barnes, 2015)
  - A rule-based tutoring system for teaching logic proof problems
  - Student solves 3-4 problems per level (Total 6 levels)
  - Level score ($LevelScore_i, i \in [1,6]$) is given for each student based on his/her performance on the last problem in the level i

Deep Thought : Reward Function

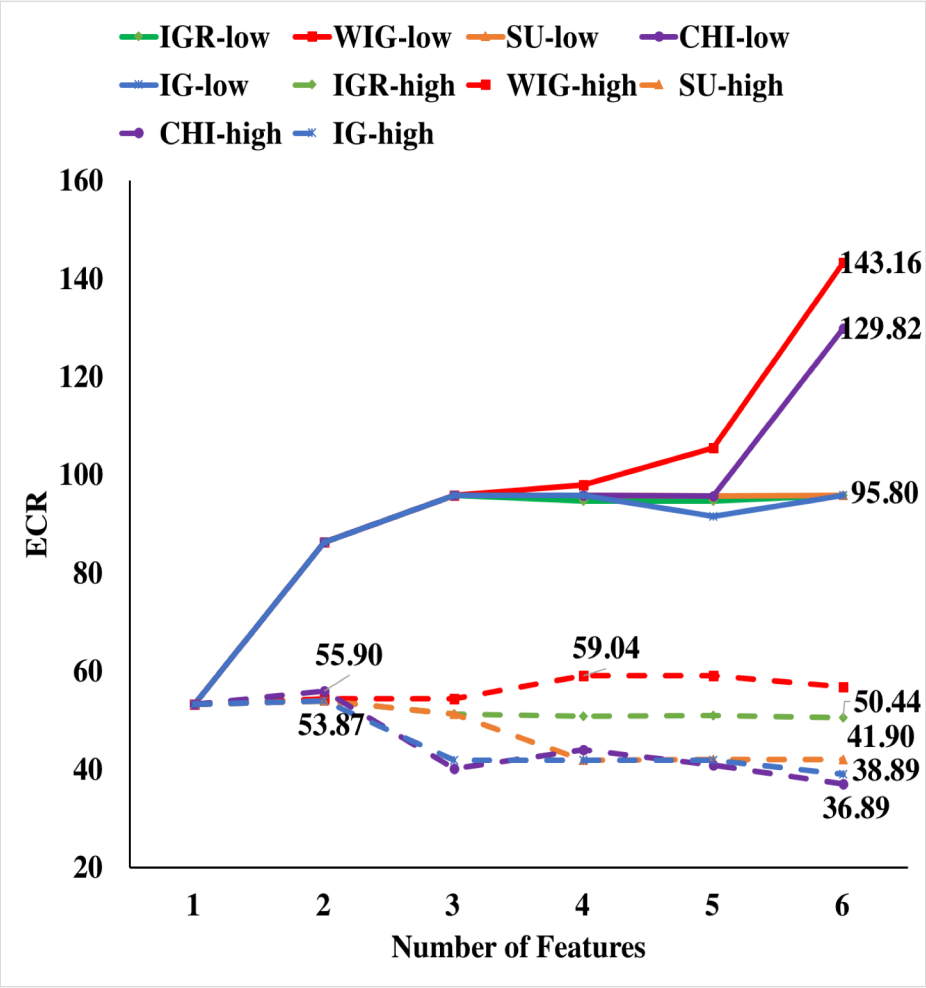
- Immediate Reward
  - $R_1 = LevelScore_1$
  - $R_i = LevelScore_i - LevelScore_{i-1},\ i \in [2,6]$
- Delayed Reward

$$R_{delay} = LevelScore_6 - LevelScore_1$$

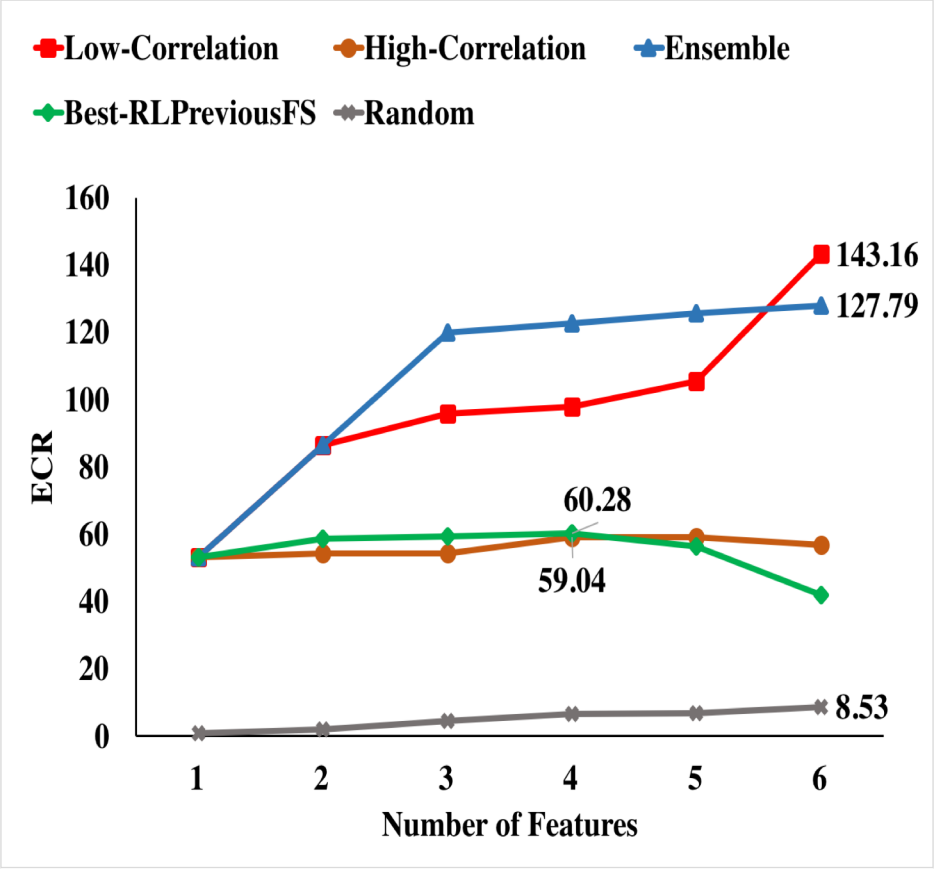
Deep Thought Data Sets

- Total 303 students in Fall 2014 and Spring 2015
- Average time spend in tutor is 416.60 minutes
- Total 135 features
- Action set
  - should it ask student to solve the next problem (PS)
  - should it provide an example to show the student how to solve the next problem (WE)

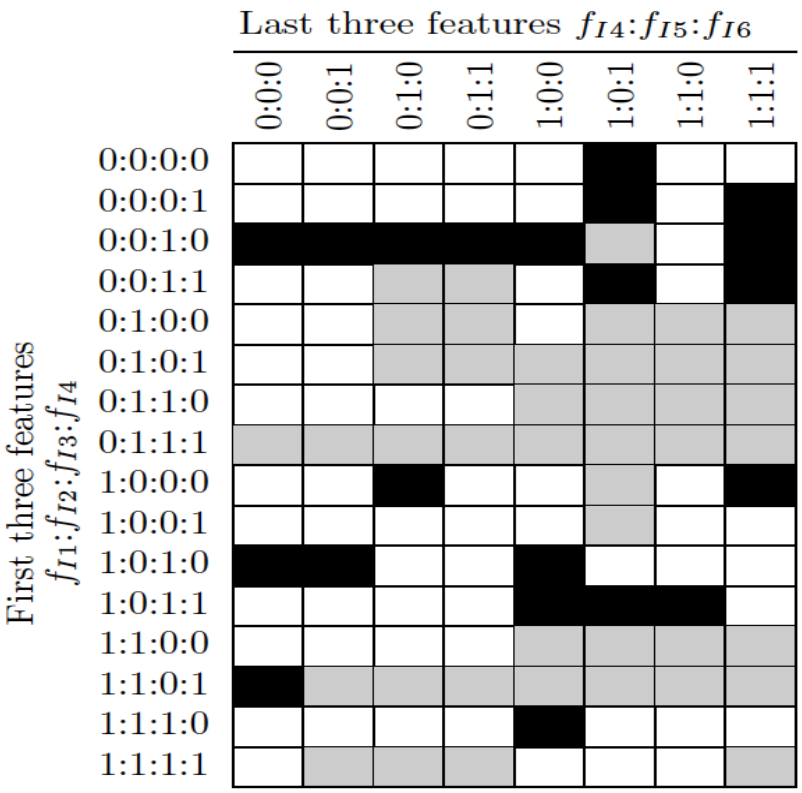
Result: High vs Low correlation



Results: Overall Evaluation



Induced Pedagogical Strategy

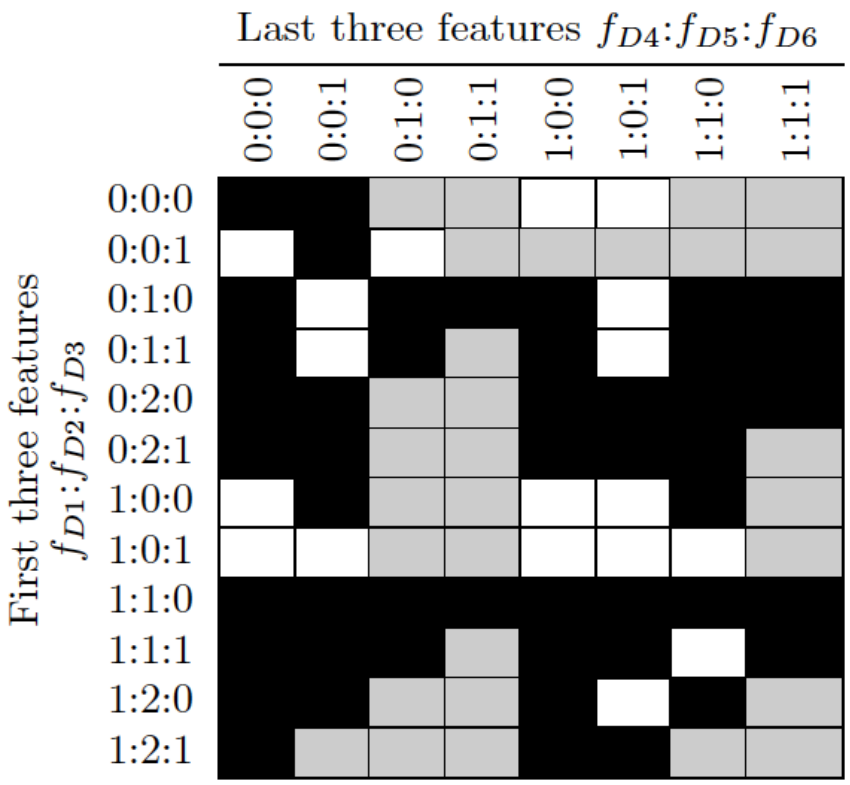


The best Policy

64 rules associated with WE (White)

21 rules associated with PS (Black)

43 no rules (Gray)



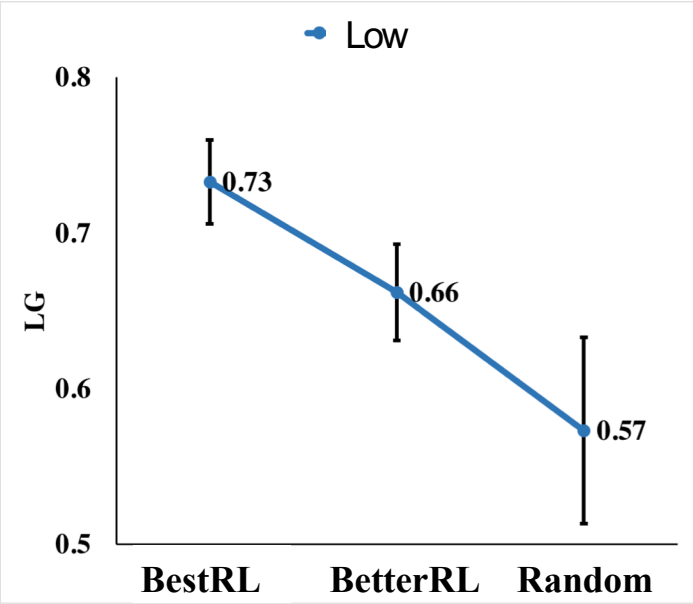
Another Policy

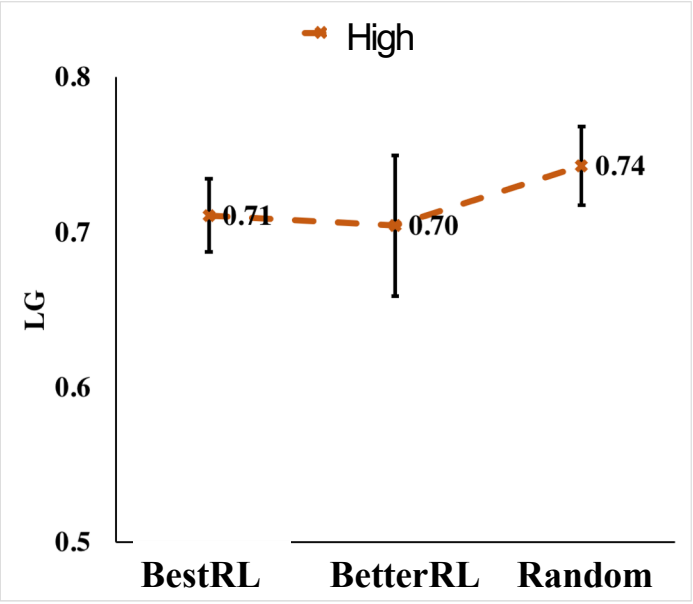
18 rules associated with WE

48 rules associated with PS

30 no rules

Learning Performance Result





- Significant difference among three Low groups
  - $F(2,46) = 3.99,\ p = 0.025$ LG
- BestRLPolicy-Low group significant outperforms Random-Low
  - $t(27) = 2.69,\ p = 0.012$ LG
- BestRLPolicy-Low group marginally outperforms BetterRLPolicy-Low
  - $t(35) = 1.67,\ p = 0.098$ LG
- No significant difference between the three High groups

Your Task: State Representation

- Discretization the features
- Feature Extraction and/or Feature selection (explore new methods)
- No more than 8 features (new or selected features)
- Evaluation: Both ECR and IS

- Rank all the project by ECR: [80-100] * 0.025 points.
- Rank all the project by IS: [80-100] * 0.025 points.
- Presentation + Report: 5 points
- Submit your code and we will run it.

Thank you !