Induce Pedagogical Strategy Using Reinforcement Learning

Min Chi, Shitian Shen

591/791: ML for User Adaptive System

March, 2017

Reinforcement Learning in practice

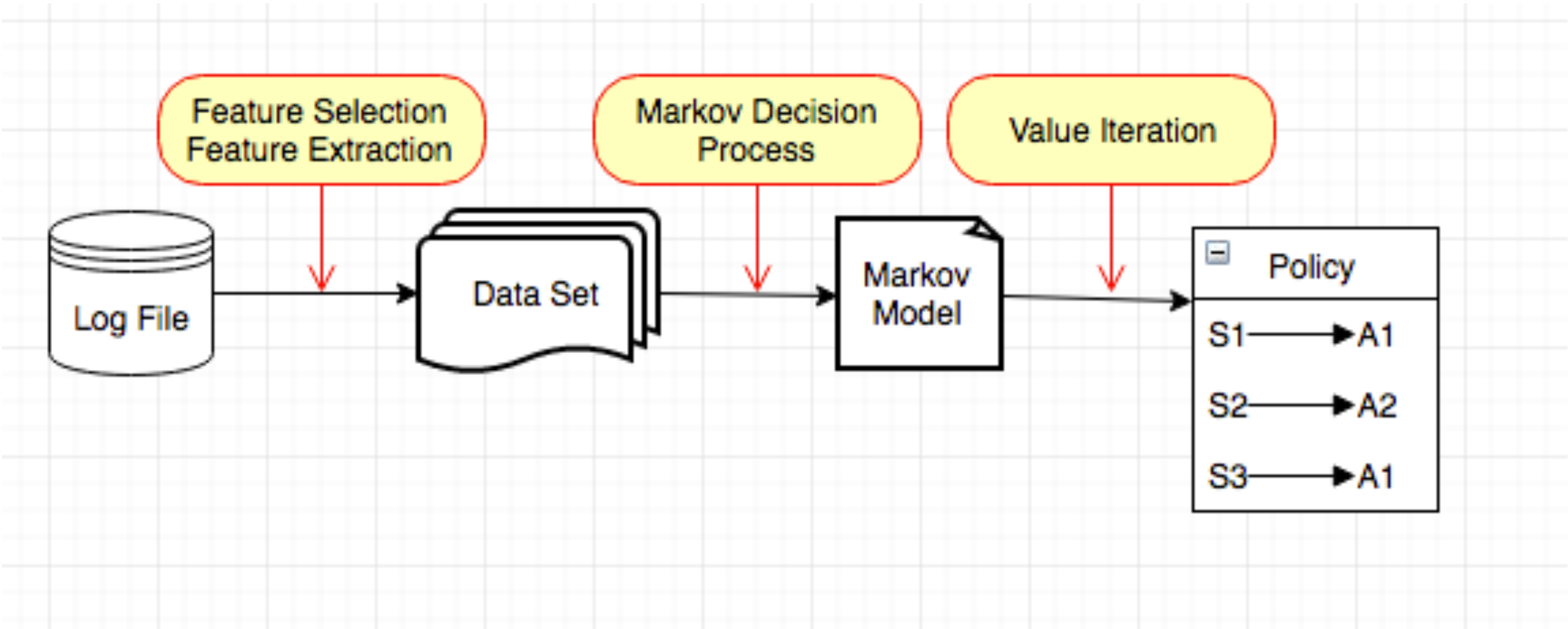
- Induce policy to make Intelligent Tutoring Systems:
  - Adaptive
  - Effective

- Reinforcement Learning in practice:

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

  - Real dataset
  - Including all the necessary steps for RL project

Process


Feature Selection
Feature Extraction
Markov Decision
Process
Value Iteration
Log File
Data Set
Markov
Model
Policy
S1 → A1
S2 → A2
S3 → A1

Markov Decision Processes (MDPs)

$S = \{S_1, \ldots, S_n\}$ state space;

Student Competence, Concept Difficulty

$A = \{A_1, \ldots, A_m\}$ action space;

{Elicit, Tell}

R : reward

Student Learning Gain

T : is a set of transition probabilities between states.

Output:

$\pi: S \rightarrow A$ is defined as a policy.

An Example Log File

Pretest =0

t1	**T: Which principle** will help you calculate the KE of the rock?
t2	**S: Definition of energy.**
t3	T: **No, we should apply the Definition of Kinetic Energy**
t4	**T:** Please write the equation….
t5	S: **KE=0.5*m*v1^2**
t6	T: Go ahead calculate the equation.
t7	S: **0.5*0.6kg*2.0 m/s^2 = 1.2 J**
…	
t423	**T:** SInce the kinetic energy…

Posttest =0.8

NC

State:

Action: *{Elicit, Tell}*

Reward:

t1	**T: Which principle** will help you calculate the KE of the rock?	**Elicit**
t2	S: Definition of energy.	
t3	T: **No, we should apply the Definition of Kinetic Energy**	
t4	**T:** Please write the equation….	**Elicit**
t5	S: **KE=0.5*m*v1^2**	
t6	T: Go ahead calculate the equation.	**Elicit**
t7	S: **0.5*0.6kg*2.0 m/s^2 = 1.2** J	
	…	
t423	**T:** SInce the kinetic energy…	**Tell**

Posttest =0.8

State feature:

Student Competence

Pretest =0

t1	T: **Which principle** will help you calculate the KE of the rock?	Elicit
t2	**S: Definition of energy.**	✗
t3	T: **No, we should apply the Definition of Kinetic Energy**	
t4	T: Please write the equation….	Elicit
t5	S: **KE=0.5*m*v1^2**	✓
t6	T: Go ahead calculate the equation.	Elicit
t7	S: **0.5*0.6kg*2.0 m/s^2 = 1.2 J**	✓
…		
t423	T: SInce the kinetic energy…	Tell

Posttest =0.8

Pretest =0

t1	T: Which principle will help you calculate the KE of the rock?	**Elicit**
t2	S: Definition of energy.	✗
t3	T: No, we should apply the Definition of Kinetic Energy	
t4	T: Please write the equation….	**Elicit**
t5	S: KE=0.5*m*v1^2	✓
t6	T: Go ahead calculate the equation.	**Elicit**
t7	S: 0.5*0.6kg*2.0 m/s^2 = 1.2 J	✓
	…	
t423	T: SInce the kinetic energy…	**Tell**

Posttest =0.8

NC STATE UNIVERSITY

Number of Correct

Pretest =0

			Number of Correct
t1	**T: Which principle** will help you calculate the KE of the rock?	**Elicit**	0
t2	**S: Definition of energy.**	✗	
t3	T: **No, we should apply the Definition of Kinetic Energy**		0
t4	**T:** Please write the equation....	**Elicit**	
t5	S: **KE=0.5*m*v1^2**	✓	1
t6	T: Go ahead calculate the equation.	**Elicit**	
t7	S: **0.5*0.6kg*2.0 m/s^2 = 1.2 J**	✓	
	…		…
t423	**T:** SInce the kinetic energy…	**Tell**	55

Posttest =0.8

Number Correct → Student Competence

if ≤ 40, *Low*; otherwise, *High*

t1	T: Which principle will help you calculate the KE of the rock?	Elicit	0	*Low*
t2	S: Definition of energy.			
t3	T: No, we should apply the Definition of Kinetic Energy		0	*Low*
t4	T: Please write the equation....	Elicit		
t5	S: KE=0.5*m*v1^2		1	*Low*
t6	T: Go ahead calculate the equation.	Elicit		
t7	S: 0.5*0.6kg*2.0 m/sˆ2 = 1.2 J			
	…		…	…
t423	T: SInce the kinetic energy…	Tell	55	*High*

Posttest =0.8

State: Student Competence {*Low, High*}

t1	T: Which principle will help you calculate the KE of the rock?	Elicit	*Low*
t2	S: Definition of energy.		
t3	T: No, we should apply the Definition of Kinetic Energy		*Low*
t4	T: Please write the equation....	Elicit	
t5	S: KE=0.5*m*v1^2		*Low*
t6	T: Go ahead calculate the equation.	Elicit	
t7	S: 0.5*0.6kg*2.0 m/s^2 = 1.2 J		
...			...
t423	T: SInce the kinetic energy...	Tell	*High*

Posttest =0.8

State: Student Competence {*Low, High*}

Action: {*Elicit, Tell*}

Reward:

t1	T: Which principle will help you calculate the KE of the rock?	Elicit
t2	S: Definition of energy.	
t3	T: No, we should apply the Definition of Kinetic Energy	
t4	T: Please write the equation....	Elicit
t5	S: KE=0.5*m*v1^2	
t6	T: Go ahead calculate the equation.	Elicit
t7	S: 0.5*0.6kg*2.0 m/s^2 = 1.2 J	
...		
t423	T: SInce the kinetic energy...	Tell

Posttest =0.8

Low

↓ Elicit

Low

↓ Elicit

Low

↓ Elicit

...

High

⊥ Tell

Reward: Normalized Learning Gain (NLG) X100

Pretest =0

t1	T: Which principle will help you calculate the KE of the rock?	Elicit
t2	S: Definition of energy.	
t3	T: No, we should apply the Definition of Kinetic Energy	
t4	T: Please write the equation….	Elicit
t5	S: KE=0.5*m*v1^2	
t6	T: Go ahead calculate the equation.	Elicit
t7	S: 0.5*0.6kg*2.0 m/s^2 = 1.2 J	
	…	
t423	T: SInce the kinetic energy…	Tell

Posttest =0.8

$$NLG = \frac{\text{Posttest} - \text{Pretest}}{1 - \text{Pretest}}$$

$$NLG \times 100$$

$$= \frac{0.8 - 0}{1 - 0} \times 100 = 80$$

State: Student Competence {*Low, High*}

Action: {*Elicit, Tell*}

Reward: NLG × 100

t1	T: Which principle will help you calculate the KE of the rock?	Elicit
t2	S: Definition of energy.	
t3	T: No, we should apply the Definition of Kinetic Energy	
t4	T: Please write the equation....	Elicit
t5	S: KE=0.5*m*v1^2	
t6	T: Go ahead calculate the equation.	Elicit
t7	S: 0.5*0.6kg*2.0 m/s^2 = 1.2 J	
...		
t423	T: SInce the kinetic energy...	Tell

Posttest =0.8

Low

$\downarrow$ Elicit,0

Low

$\downarrow$ Elicit,0

Low

$\downarrow$ Elicit,0

...

High

$\perp$ Tell,80

One student's Log File ➔ One Trajectory

$$\textbf{\textit{Low}} \xrightarrow{\text{Elicit, 0}} \textbf{\textit{Low}} \xrightarrow{\text{Elicit, 0}} \textbf{\textit{Low}} \xrightarrow{\text{Elicit, 0}} \textbf{\textit{High}} \xrightarrow{\text{Tell,80}} \bullet$$

Training Dataset → Trajectories

$\textit{Low} \xrightarrow{\text{Elicit, 0}} \textit{Low} \xrightarrow{\text{Elicit, 0}} \textit{Low} \xrightarrow{\text{Elicit, 0}} \textit{High} \xrightarrow{\text{Tell,80}} \bullet$

$\textit{High} \xrightarrow{\text{Elicit, 0}} \textit{Low} \xrightarrow{\text{Elicit, 0}} \textit{Low} \xrightarrow{\text{Tell, 0}} \textit{Low} \xrightarrow{\text{Elicit, 60}} \bullet$

$\textit{Low} \xrightarrow{\text{Tell, 0}} \textit{Low} \xrightarrow{\text{Elicit, 0}} \textit{High} \xrightarrow{\text{Elicit, 0}} \textit{High} \xrightarrow{\text{Tell, 20}} \bullet$

…

$\textit{High} \xrightarrow{\text{Elicit, 0}} \textit{High} \xrightarrow{\text{Tell, 0}} \textit{High} \xrightarrow{\text{Elicit, 0}} \textit{High} \xrightarrow{\text{Elicit, -30}} \bullet$

Training Dataset → Trajectories

Low $\xrightarrow{\text{Elicit, 0}}$ ***Low*** $\xrightarrow{\text{Elicit, 0}}$ ***Low*** $\xrightarrow{\text{Elicit, 0}}$ ***High*** $\xrightarrow{\text{Tell,80}} \bullet$

High $\xrightarrow{\text{Elicit, 0}}$ ***Low*** $\xrightarrow{\text{Elicit, 0}}$ ***Low*** $\xrightarrow{\text{Tell, 0}}$ ***Low*** $\xrightarrow{\text{Elicit, 60}} \bullet$

Low $\xrightarrow{\text{Tell, 0}}$ ***Low*** $\xrightarrow{\text{Elicit, 0}}$ ***High*** $\xrightarrow{\text{Elicit, 0}}$ ***High*** $\xrightarrow{\text{Tell, 20}} \bullet$

…

High $\xrightarrow{\text{Elicit, 0}}$ ***High*** $\xrightarrow{\text{Tell, 0}}$ ***High*** $\xrightarrow{\text{Elicit, 0}}$ ***High*** $\xrightarrow{\text{Elicit, -30}} \bullet$

Transition Probabilities T estimated from Training Dataset.

An Example of *P*(*Low*|*High, Elicit*)

$$Low \xrightarrow{\text{Elicit, 0}} Low \xrightarrow{\text{Elicit, 0}} Low \xrightarrow{\text{Elicit, 0}} High \xrightarrow{\text{Tell, 80}} \bullet$$

$$\boxed{High \xrightarrow{\text{Elicit, 0}} Low} \xrightarrow{\text{Elicit, 0}} Low \xrightarrow{\text{Tell, 0}} Low \xrightarrow{\text{Elicit, 60}} \bullet$$

$$Low \xrightarrow{\text{Tell, 0}} Low \xrightarrow{\text{Elicit, 0}} \boxed{High \xrightarrow{\text{Elicit, 0}} High} \xrightarrow{\text{Tell, 20}} \bullet$$

$$\boxed{High \xrightarrow{\text{Elicit, 0}} High} \xrightarrow{\text{Tell, 0}} \boxed{High \xrightarrow{\text{Elicit, 0}} High} \xrightarrow{\text{Elicit, -30}} \bullet$$

$$P(Low \mid High, Elicit) = \frac{\# High \xrightarrow{Elicit} Low}{\# \mathrm{High} \xrightarrow{\mathrm{Elicit}} \mathrm{Low} + \# \mathrm{High} \xrightarrow{Elicit} High} = \frac{1}{4}$$

An Example **Single-feature** Policy

$$
\begin{aligned}
<Low> \quad &\rightarrow \quad Tell \\
<High> \quad &\rightarrow \quad Elicit
\end{aligned}
$$

Two state features:

Competence; Difficulty

4 states:

{<*Low*, *Easy*>, <*Low*, *Difficult*>,
<*High*, *Easy*>, <*High*, *Difficult*>}

t3	T: No, we should apply the Definition of Kinetic Energy	
t4	T: Please write the equation....	Elicit
t5	S: KE=0.5*m*v1^2	
t6	T: Go ahead calculate the equation.	Elicit
t7	S: 0.5*0.6kg*2.0 m/s^2 = 1.2 J	
...		
t423	T: SInce the kinetic energy...	Tell

Posttest =0.8

<*Low*, *Easy*>

$\downarrow$ Elicit, 0

<*Low*, *Difficult*>

$\downarrow$ Elicit, 0

<*Low*, *Easy*>

$\downarrow$ Elicit, 0

<*High*, *Difficult*>

$\perp$ Tell, 80

A Two-Feature Policy Example

$$
\begin{aligned}
< Low, Easy > \quad &\rightarrow \quad Elicit \\
< High, Difficult > \quad &\rightarrow \quad Elicit \\
< High, Easy > \quad &\rightarrow \quad Tell \\
< Low, Difficult > \quad &\rightarrow \quad Tell
\end{aligned}
$$

Value Iteration

1. Initial value of each state: $V(s)$

Value Iteration

1. Initial value of each state: $V(s)$
2. Update each state's $V(s)$ using neighboring $V(s')s$:

$$V(s) = \max_a \sum_{s'} P(s'|s,a) \; [R(s,a,s') \; + \; \gamma \cdot \; V(s')]$$

Probability of landing on a neighboring state.

Reward/Cost of going to the neighboring state.

Value of the neighboring state

Value Iteration

1. Initial value of each state: $V(s)$
2. Update each state's $V(s)$ using neighboring $V(s')s$:

$$V(s) = \max_a \sum_{s'} P(s'|s,a) \;\; [R(s,a,s') \;\; + \;\; \gamma \cdot \quad V(s')]$$

Probability of landing on a neighboring state.

Reward/Cost of going to the neighboring state.

Value of the neighboring state

3. Induce the policy:

$$\pi(s) = \operatorname*{argmax}_a \sum_{s'} P(s'|s,a)\,[R(s,a,s') \;\; + \;\; \gamma \cdot \; V(s')]$$

For any state s, take an action to the neighboring s' with highest V(s')

γ :: Discount factor 0.9.

Adapted from: Sutton & Barto (1998)

Purpose

- Induce policy to make Intelligent Tutoring System:
  - Adaptive: **Feature Discretization/Selection/Extraction**
  - Effective: **ECR**

- Reinforcement Learning in practice:

  What is the best **action** for the **agent** *(tutor)*

  to take in any **state** *(learning context)*

  in order to maximize **reward** *(student learning)*
  - Real dataset
  - Including all the necessary steps for RL project.

Policy Evaluation

- 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

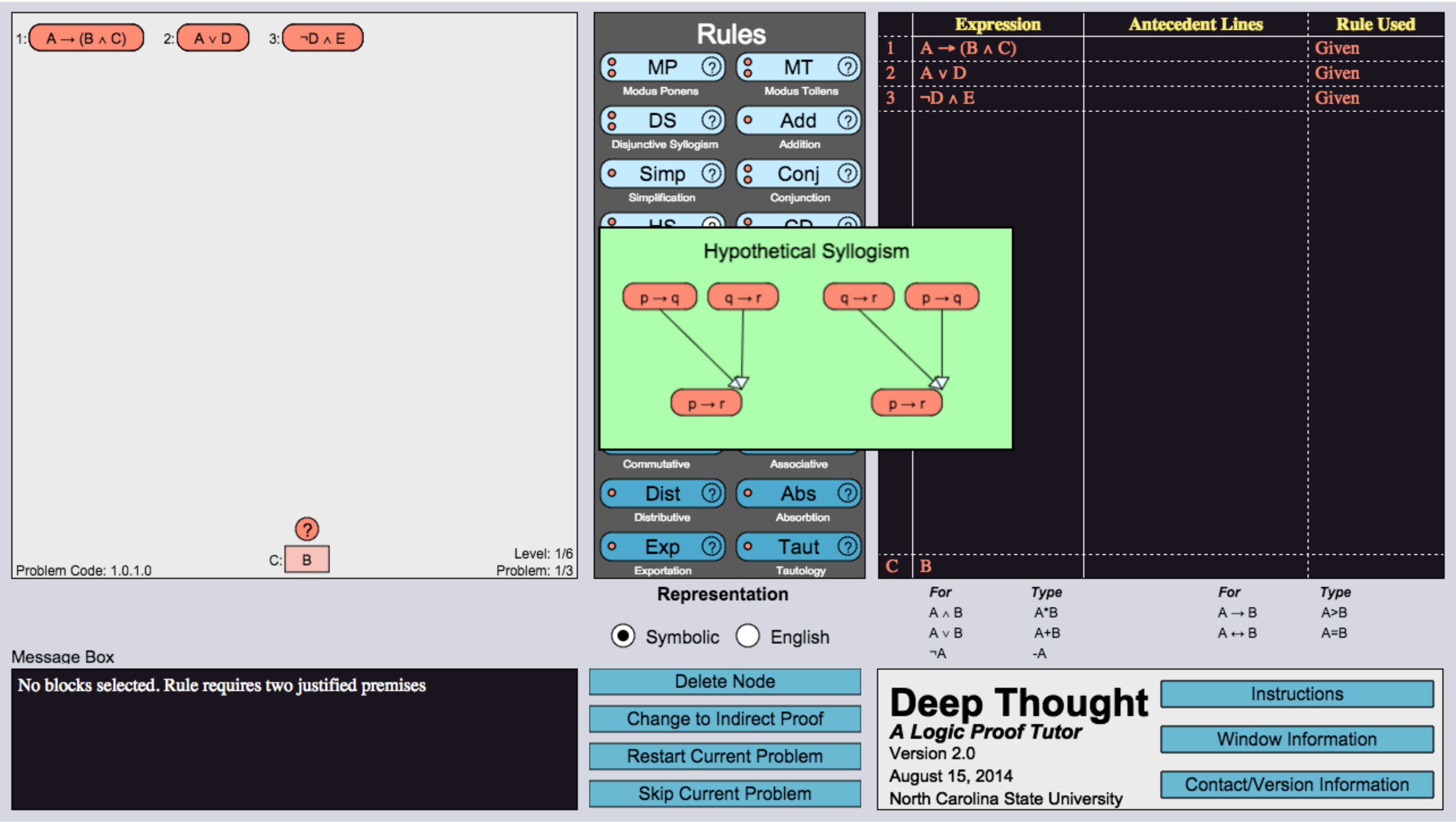
- The higher ECR of the policy means the better policy

Shitian's Work

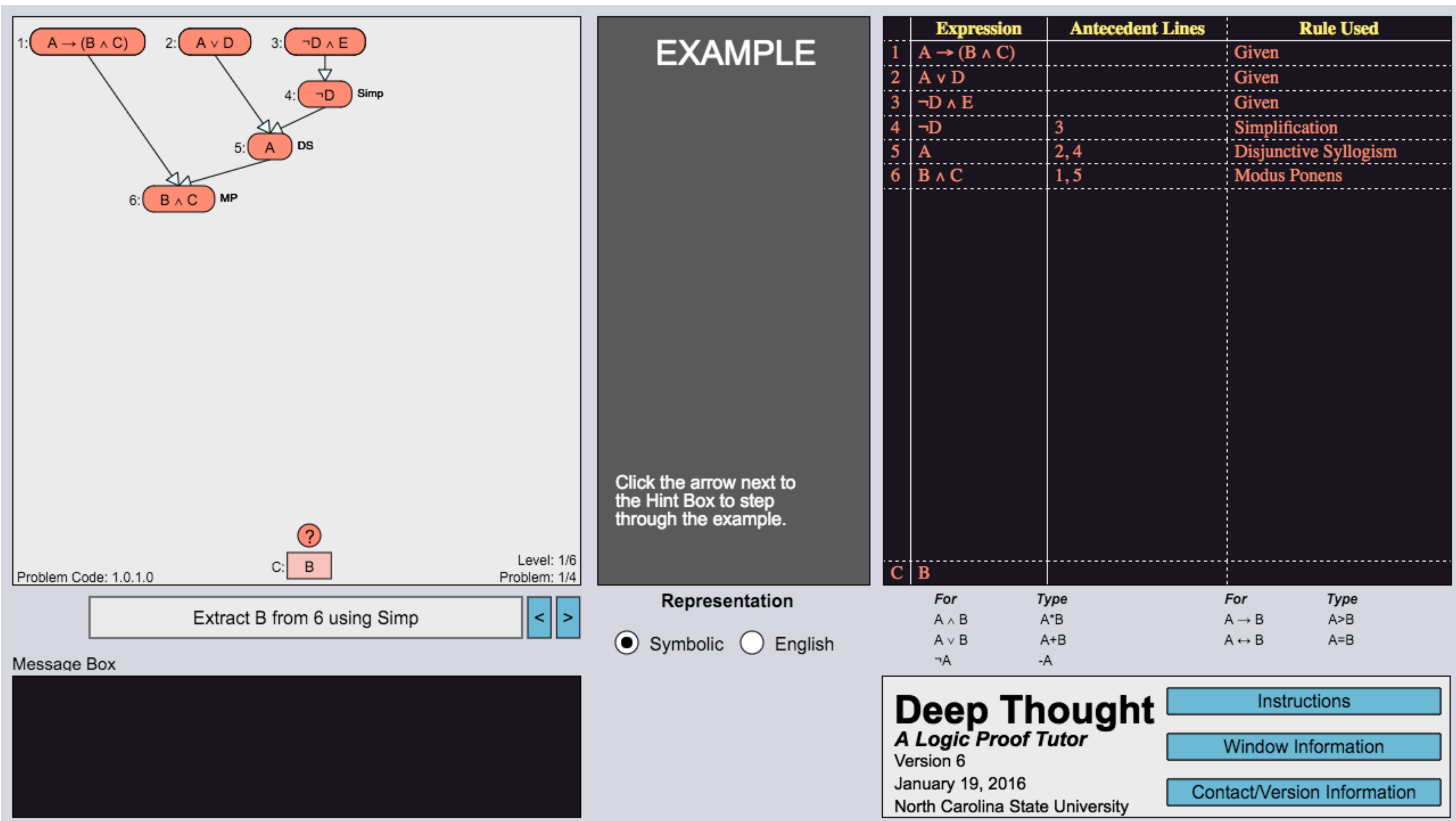
Deep Thought (Dr. Barnes, 2015)

- A rule-based tutoring system for teaching logic proof problems
- Student solves 1-3 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

Problem Solving



Work Example



Deep Thought Data Sets

- Total 303 students
- Average time spend in tutor is 416.60 minutes
- Total 4 categories and 124 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)

Four categories: 124 State Features

- Autonomy: the amount of work done by the student
  - PSCount,
  - PercPS
- Temporal Situation: the time related information about the work process
  - TotalTime,
  - avgPSTime,
- Student Action:the statistical measurement of student's behavior
  - AppCount,
  - hintCount
- Performance: Students' performance on current problem
  - ruleScore,
  - wrongApp

Feature Discretization

- Median split

TotalTime	[0: <=172.34, 1: >172.34]
avgTime	[0: <=6.25, 1: >6.25]
hint	[0: <=0.04, 1: >0.04]

- Kmeans
  - Data points didn’t uniformly distribute
  - Particular data points may group together
  - Avoid unbalanced clustering

  CurrPro_NumProbRule [0: close to 4.1, 1: close to 6.4]

Feature-Selection: Correlation Metrics

Given labeled data, we can compute some simple score S(i) that measures how informative each feature X is about class labels Y.

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

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

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

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

Feature-Selection: Correlation Metrics

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

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

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

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

- Weighted Information Gain (WIG) (We proposed)

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

Correlation-based Feature Selection Methods

- Feature Selection for model-based RL
- Apply correlation between current optimal feature set and potential feature as the feature selection criteria
- Forward feature selection strategy

10 Correlation-based Methods

- Explore both high and low correlation
- Obtain 10 correlation-based feature selection methods (5 correlation metrics $\times$ 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

Correlation-based Methods: Algorithm

Algorithm

Require: Ω: $Feature\ Space$; $\mathcal{D}$: $Training\ Data$;
$\mathcal{N}$: $Maximun\ Number\ of\ Selected\ Features$;

Ensure: $\mathcal{S}^*$: $Optimal\ Feature\ Set$

```
for f_i in Ω do
    ECR_i ← CalculateECR(D, f_i)
end for
Add f* with highest ECR to S*
while SIZE(S*) < N do
    for f_i in Ω − S* do
        C_i ← CalculateCORRELATION(S*, f_i, m)
    end for
    F ← SelectTop(C, 5, reverse)
    for f_i in F do
        ECR_i ← CalculateECR(D, S* + f_i)
    end for
    Replace S* by S* + f_i with highest ECR
end while
```

Initialization

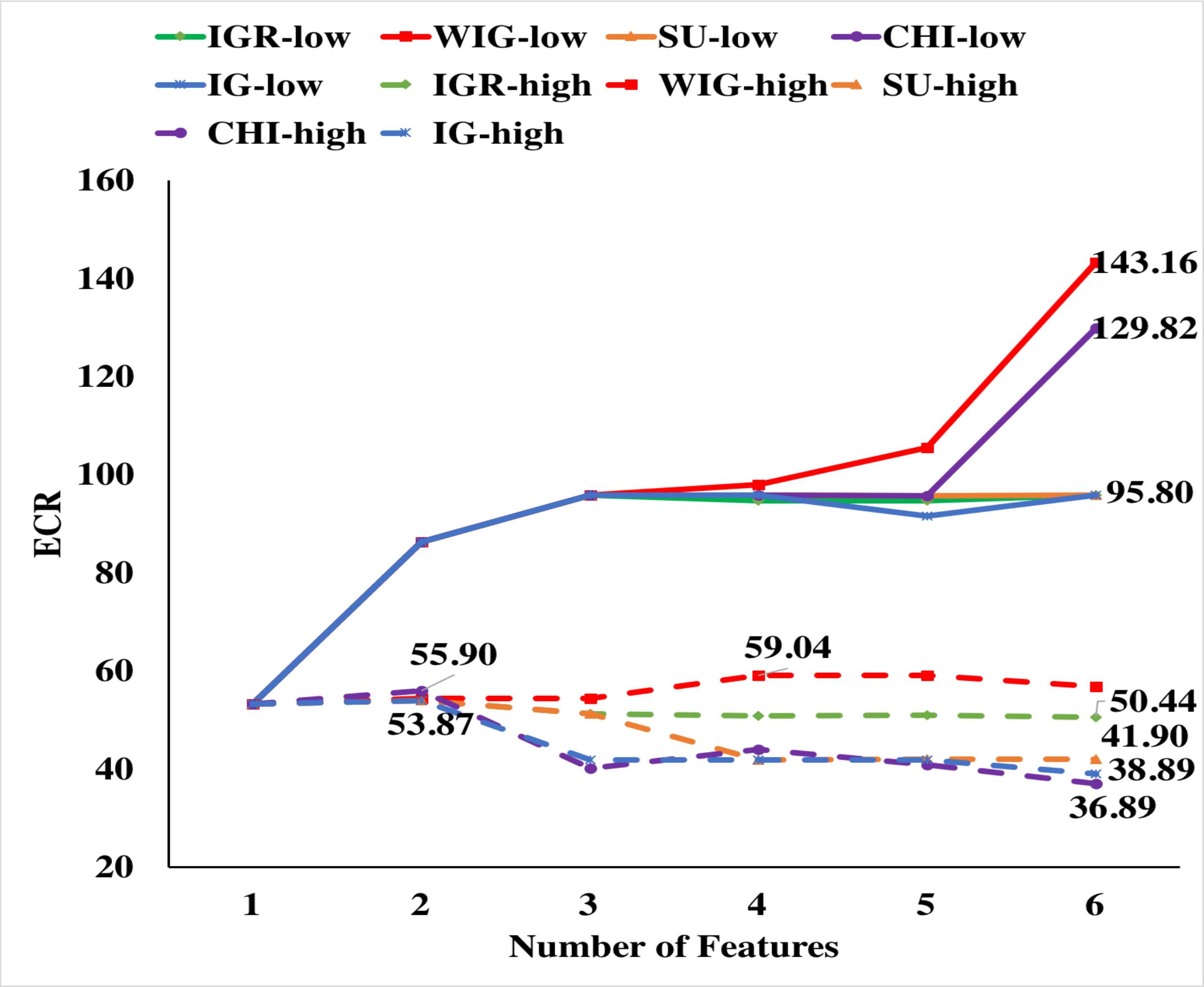
Feature Selection

Updating Feature Set

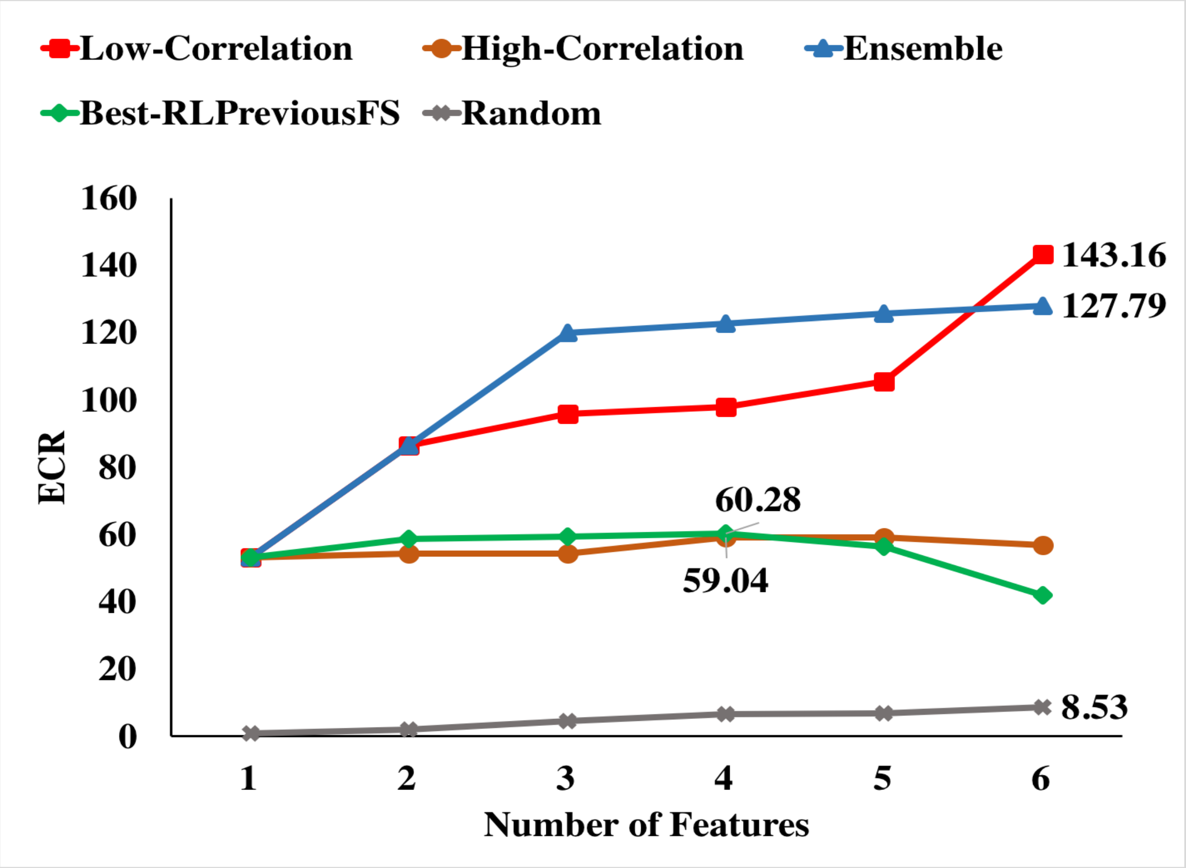
Ensemble Methods

- Integrate multiple selection methods
- 10 correlation-based methods
- 4 other RL based methods

Result: High vs Low correlation

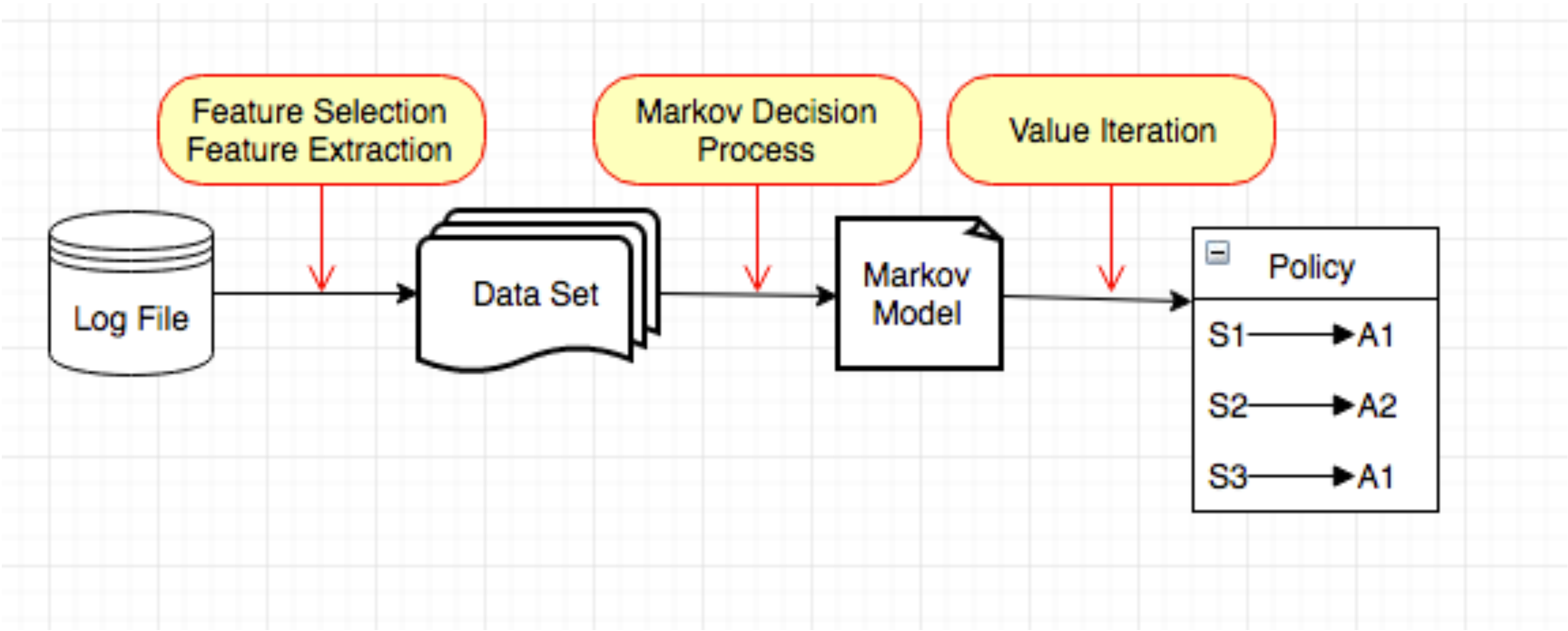


Results: Overall Evaluation

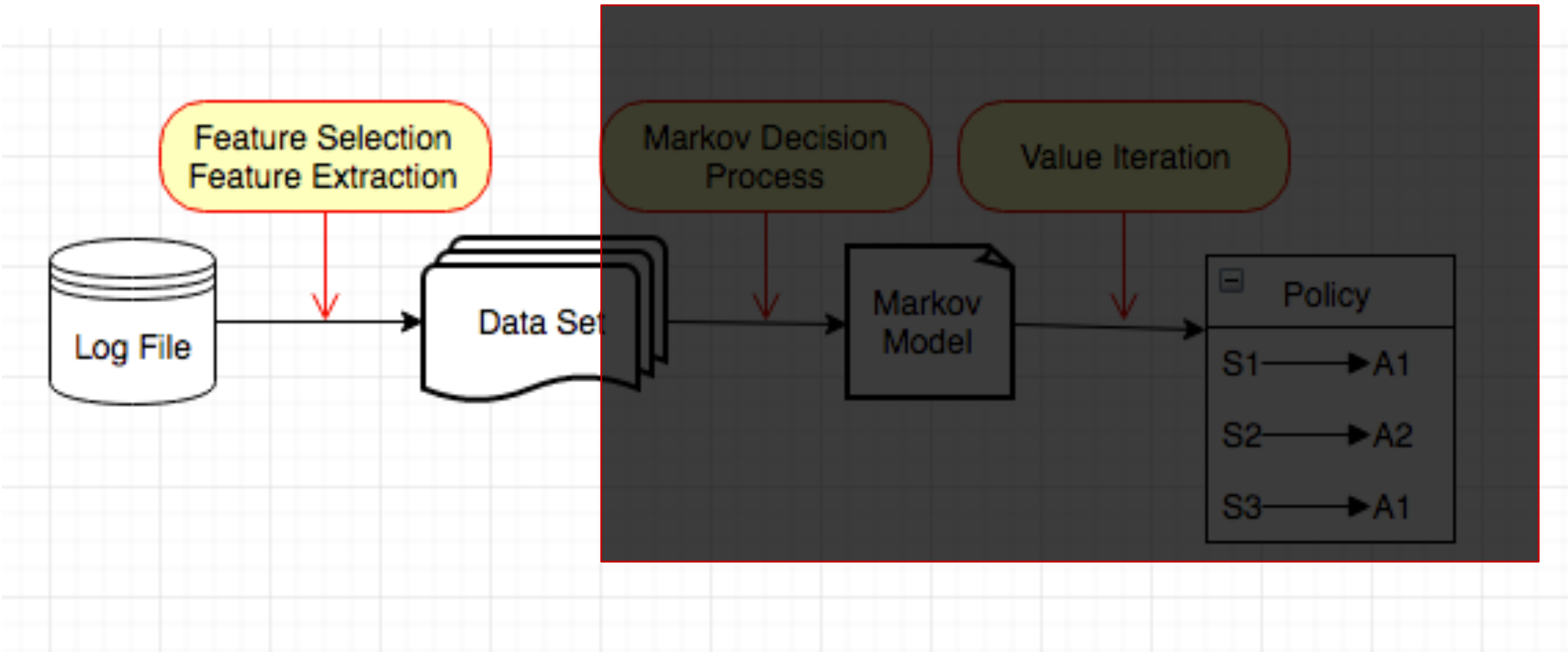


The Assigned Project:

Process


Feature Selection
Feature Extraction
Markov Decision
Process
Value Iteration
Log File
Data Set
Markov
Model
Policy
S1
A1
S2
A2
S3
A1

Process


Feature Selection
Feature Extraction
Markov Decision Process
Value Iteration
Log File
Data Set
Markov Model
Policy
S1
A1
S2
A2
S3
A1

Goals for Feature Selection or Extraction

- Representing interactive environment effectively (Maximize ECR)
- Please Note:
  - Discrete features
  - Maximum feature size is 8

Suggestion: Discretization Procedure

- Feature Selection
  - Discretize features first
  - Explain how to discretize features in report
  - Keep the original names of selected features

- Feature Extraction
  - Not necessary to discretize features first
  - Explain how to extract features in report
  - Get a new name the extracted features: f1, f2….

To run MDP package, all features must be discretized.

Suggested: Feature Selection

- Discrete features (median split, distribution)
- Filtering approach
  - Design a ranking function and select top n features
  - ECR of each single-feature policy
- Forward feature selection
  - Good selection strategy
    - Use ECR as a selection standard
  - Maintain a limited search space
    - Apply correlation, mutual information gain as the condition

Try other methods.

Additional points for exploring novel methods.

Suggested: Feature Extraction

- PCA
  - How to deal with discrete features
  - Factor analysis of mixed data (FAMD)
- Fuzzy Clustering
  - Data point can belong to multiple clusters
  - Specify distance function, handing continuous and discrete features
- Autoencoder
  - Do Not recommend
- Output Discrete Features
  - Explain how to extract features in report
  - Transfer extracted features into discrete ones

Suggested: Feature Extraction

- Unsupervised feature extraction
- Construct connection between feature extraction with reinforcement learning
- Apply ECR as the condition in the feature extraction process

Markov Decision Process

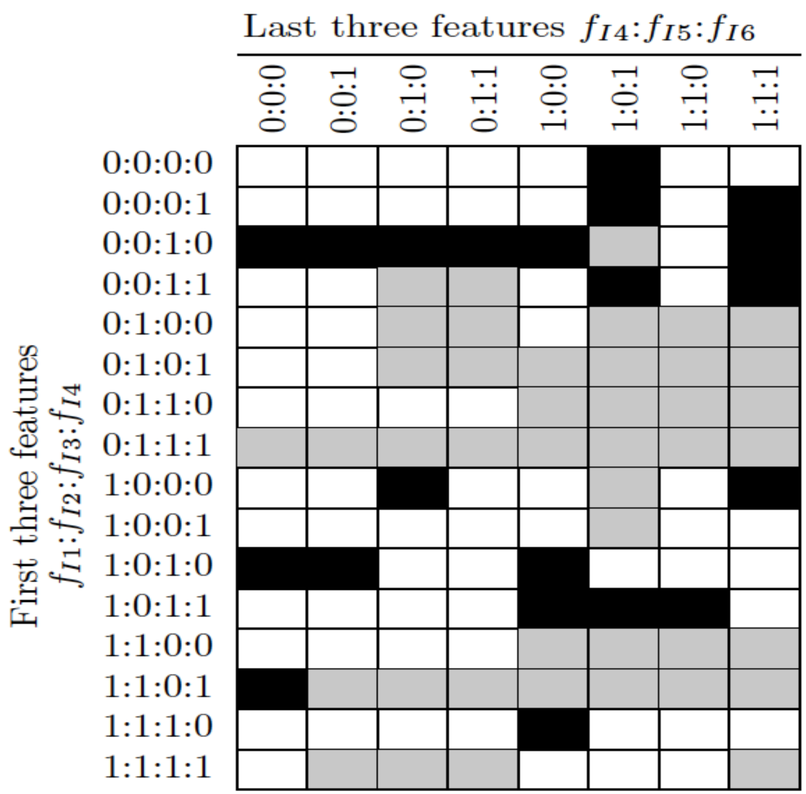
1	student	currProb	course	session	priorTutorAc	reward	Level	probDiff
2	0006-F14	1.0.1.0	226-001-BAR	1	PS	0	1	0
3	0006-F14	1.0.2.0	226-001-BAR	1	WE	0	1	0
4	0006-F14	1.0.3.0	226-001-BAR	1	WE	0	1	0
5	0006-F14	1.0.4.0	226-001-BAR	1	PS	-94.078947	1	0
6	0006-F14	2.1.1.0	226-001-BAR	1	PS	0	2	1
7	0006-F14	2.1.2.0	226-001-BAR	1	WE	0	2	1
8	0006-F14	2.1.3.0	226-001-BAR	1	PS	161.81004	2	1
9	0006-F14	3.1.1.0	226-001-BAR	1	WE	0	3	1
10	0006-F14	3.1.2.0	226-001-BAR	1	WE	0	3	1
11	0006-F14	3.1.3.0	226-001-BAR	1	PS	-43.265073	3	1

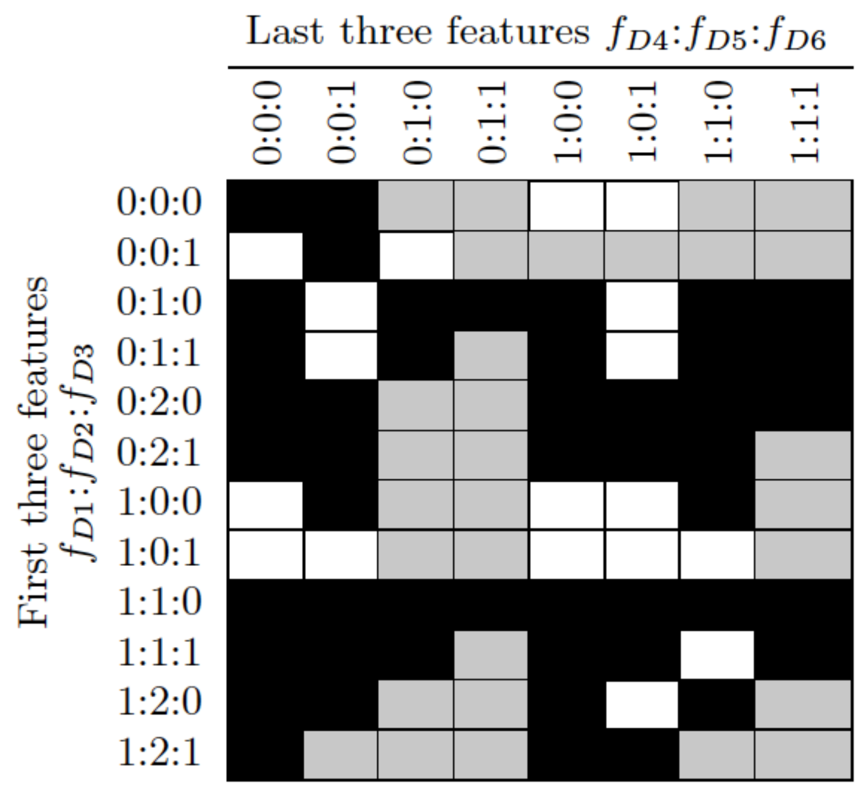
- States:
  - Level=1, probDiff=0, then state = ‘1:0’,
  - Level=2, probDiff=1, then state = ‘2:1’,
  - Level=3, probDiff=1, then state = ‘3:1’

Policy Example

```
Policy:
state -> action, value-function
1:0 -> WE, 36.2980141792
2:1 -> PS, 50.6234894691
3:-1 -> WE, 30.7327270058
4:1 -> PS, 38.0276536834
5:1 -> PS, 31.1271670497
6:1 -> PS, 38.7017874912
3:1 -> PS, 37.5870989962
2:-1 -> WE, 28.4686139413
4:-1 -> WE, 26.6276418488
5:-1 -> WE, 26.4065388562
6:-1 -> WE, 0.0
ECR value: 36.2980141792
```

Policy Visualization- For your Presentation





Policy 1

64 rules associated with WE (White)

21 rules associated with PS (Black)

43 no rules (Gray)

Policy 2

18 rules associated with WE

48 rules associated with PS

30 no rules

Demo

Q & A

Thank you !