MDP Programming

In this workshop, you will explore an intelligent tutoring system called Deep Thought. This system is fundamentally designed to help students learn the concept of logic and proof by providing corresponding questions and examples. Specifically, during the students' learning process, Deep Thought takes two actions, providing 1) Problem Solving (PS) and 2) Work Example (WE), based on the students' state for their best learning gain. The PS action provides a problem to students and lets them solve it while the WE action gives students a tutorial that shows how to solve a sample problem step by step.

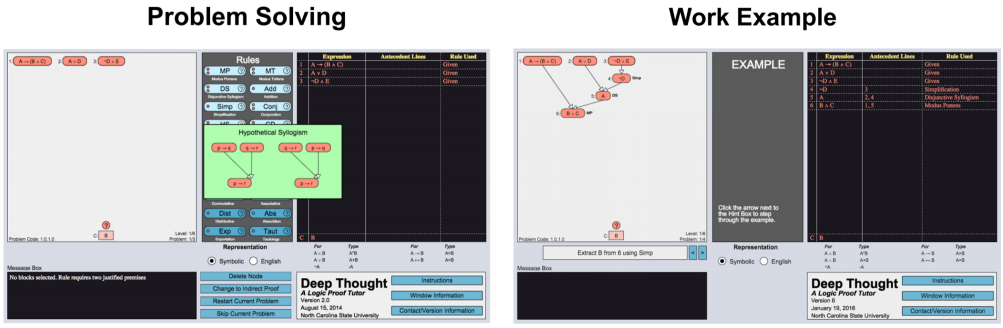

Figure 1: The Deep Thought Environment

In short, this tutoring system can be seen as MDP - where the action is either PS or WE, the state is students' learning context, and the reward is learning gain - and we can induce optimal policy from MDP. However, it is challenging to represent a real-world environment as MDP, and thus, careful feature selection/extraction should be considered, which can best represent the state or environment. The goal of this workshop is to understand this real-world MDP problem by experimenting with a random feature selection method so that you can try or design any new methods for the homework.

For the following parts, you will be given a basic introduction to the Deep Thought environment. Read the following instructions and report all your results.

Deep Thought, an intelligent tutoring system

- A rule-based tutoring system for teaching logic proof problems.
- Each 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.
- Total 303 students participated in Fall 2014 and Spring 2015.
- Average time spend in tutor is 416.60 minutes.

- Total 130 features are collected (Dictionary file is attached.): Columns 1-6 (index from 1) are static information that should be included in the state representation as default. (student, currProb, priorTutorAction, reward, state). Columns 7-130 are the candidate features to represent the environment.

Report You are given a Python file "W8S2_MDP.ipynb" which contains functions needed for the following experiment. Follow the instructions below to explore the environ-ment.

Getting started:

- Install Python and necessary packages at least including: collections, numpy, pandas and pymdptoolbox.
- The sample output from the chosen environment looks like the following:

```
Policy:
state -> action, value-function
4:1 -> PS, 30.8719768758
1:0 -> WE, 36.7151499977
6:-1 -> WE, 0.0
2:-1 -> WE, 20.328730566
4:-1 -> WE, 7.26012984302
5:-1 -> WE, 5.14018097853
2:1 -> PS, 52.5931588259
3:-1 -> WE, 23.2087813231
6:1 -> PS, 7.40501167788
3:1 -> PS, 26.1715004964
5:1 -> PS, 9.49560101443
ECR value: 36.71514999769811
```

Report the followings:

1. Load the dataset "MDP_data_student200.csv" and report the number of static features and candidate features.
2. Discretize the candidate features with the given number of bins. (n_bins=2)
3. Run the random feature selection method and report the ECR from each feature set. (You are allowed to choose maximum 8 features from the candidate feature set.)
4. Plot the ECRs with the number of chosen features. Report the number of features with the highest ECR.
5. Report the policy induced from the above feature set and the number of PS/WE actions.
6. Explore different numbers of bins for the discretization and observe the result.

Solution: A solution file "W8S2_MDP.ipynb" is provided.