Machine Learning Project Proposal
Learning to Play Maze Game with Reinforcement Learning

Chen, Yuying	Herlambang, Mega Bagus	Kan, Zhuoyun	Meng, Li
S3421902	P279285	S3346935	S2651513

November 23, 2017

1 Application

This application is designed to play maze game with reinforcement learning algorithm.

- **Simulator**

  We plan to use TKinter to implement the simulator which defines the environment of this game. It basically contains three main component elements: one entrance, one terminal point, and several obstacles. Another advanced maze would also be designed to be solved as a much more complicated snenario for the reinforcement learning and possible implementation of neural networks. This maze would contain one entrance, one terminal point, several moving obstacles, keys, doors with penalties, and doors with rewards.

- **Input**

  There would be different rewards based on different states. The simulator will update the observation and reward as an input to an agent.

- **Output**

  The agent will take an action as an output according to the input.

- **How many examples**

  The agent learns until it performs the optimal strategy to reach the terminal point.

2 Methods

We compare a few algorithms (Value iteration algorithm, Q-learning, SARSA, etc) based on the score they obtain in solving the maze by varying various parameters such as explore probability, learning rate, discount factor, convergence tolerance, etc. The time to solve the maze by each algorithm is compared, and the shortest route would be the best algorithm.

3 Setup of Experiments

Our experiments basically consist of following elements:

- **Data Collection**

  After the maze simulator is correctly built, data should be collected by running the simulator. We might run a few examples to grasp a better sense of how the agent interacts with the game environment in the beginning.

- **Feature extraction**

  Features in data that are found useful for our reinforcement learning experiments should be picked out. For example, rewards are not easily defined as states and actions for a maze game and we need to set different numbers of rewards for different in-game outcomes.

- **Learning**

  The agent learns how to play the game by using chosen reinforcement learning methods. In the first stage of the experiments, we store Q-values in a table. However, neural networks would be much plausible as complexity of the maze increases. Performances of Q-learning(off-policy) and SARSA(on-policy) will also be compared afterward.

- **Optimization**

  It is likely that parameters are not set to optimal during our first run. Therefore, we need to tune different kinds of parameters to get an optimal model, such as ε(explore probability), α(learning rate), γ(discount factor).

4 Programming Language

We choose python as our programming language.

5 Planning

- **22-11-2017:**

  Discuss the project and finish the proposal.

- **29-11-2017:**

  Modify the proposal and update the planned schedule based on the feedbacks. Start implementing the simulator and Q-learning algorithm for simplae maze. Design the advanced maze and set the rules for it.

- **06-12-2017:**

  Implement the GUI of the simulator. Learn Reinforcement Learning from the lab, and look for solutions of issues encountered from implementation and integration of GUI.

- **13-12-2017:**

  Try to explore and experiment different setups and parameters of Q-learning, including different parameters (e.g., alpha, gamma), attempts runs, etc. Also, try to implement the simulator of advanced maze.

- **20-12-2017:**

  Implement the second learning algorithm, SARSA, for the purpose of comparison. If possible, the implementation of Deep Q-learning (Neural Network) can also be an option. Apply both learners to the advanced maze.

- **27-12-2017:**

  Summarize the results so far, and consider any improvements could be made.

- **03-01-2018:**

  Winter break.

- **10-01-2018:**

  Catch up delayed tasks.

- **17-01-2018:**

  Complete the system as a stable version. Summarize the final results and start writing report.

- **24-01-2018:**

  Prepare for the presentation and finish the final report.