CS221 Fall 2017 Homework [number]

SUNet ID: ashwini1
Name: Ashwini Pokle
Collaborators:

By turning in this assignment, I agree by the Stanford honor code and declare that all of this is my own work.

Problem 1

(a)

$$V_{max,min}(s,d) = \begin{cases} Utility(s) & \text{isEnd(s)} \\ Eval(s) & \text{d} = 0 \\ max_{a \in Actions(s)} V_{max,min}(Succ(s,a),d) & \text{Player(s)} = a_0 \text{ (pacman)} \\ min_{a \in Actions(s)} V_{max,min}(Succ(s,a),d) & \text{Player(s)} = a_1 \ (ghost_1) \\ \vdots & \vdots \\ min_{a \in Actions(s)} V_{max,min}(Succ(s,a),d) & \text{Player(s)} = a_{n-1} \ (ghost_{n-1}) \\ min_{a \in Actions(s)} V_{max,min}(Succ(s,a),d-1) & \text{Player(s)} = a_n \ (ghost_n) \end{cases}$$

Problem 3

(a)

$$V_{max,opp}(s,d) = \begin{cases} Utility(s) & \text{isEnd(s)} \\ Eval(s) & \text{d} = 0 \\ max_{a \in Actions(s)} V_{max,opp}(Succ(s,a),d) & \text{Player(s)} = a_0 \\ \sum_{a \in Actions(s)} \pi_{ghost}(s,a) V_{max,opp}(Succ(s,a),d) & \text{Player(s)} = a_1 \\ \vdots & \vdots \\ \sum_{a \in Actions(s)} \pi_{ghost}(s,a) V_{max,opp}(Succ(s,a),d) & \text{Player(s)} = a_{n-1} \\ \sum_{a \in Actions(s)} \pi_{ghost}(s,a) V_{max,opp}(Succ(s,a),d-1) & \text{Player(s)} = a_n \end{cases}$$

where $\pi_{ghost}(s,a) = \frac{1}{len(Actions(s))}$, a_0 is pacman, and a_1, a_2, .., a_n are ghosts.

Problem 4

(b) Evaluation function is a weighted sum of following features :

$Eval(s) = currentGameScore + \frac{20}{numberOfFood+1} + \frac{22}{sumDistToCapsule+1} + \frac{10}{minDistanceToFood} + \frac{20}{numberOfCapsules+1} + \frac{22}{sumDistanceToFood+1}$

if $minDistToScaredGhost > 0$, $Eval(s) \mathrel{+}= \frac{155}{minDistToScaredGhost}$

Feature	Reason
currentGameScore	Current Game Score is a very good indicator of how well pacman has done so far.
numberOfFood	This feature has been included as an inverse in the formula. As number of food in the grid goes to 0, inverse of numberOfFood will increase. This means that pacman will try to go to states that result in lesser number of food items.
sumDistanceToFood	Pacman should try to go to the areas of grid where there is food. Inclusion of this feature prevented pacman from thrashing in positions where it is right next to food.
minDistanceToFood	Pacman should try to minimize it's minimum distance from food. Higher the distance, lower the score. Therefore, this feature has been included as an inverse.
numOfCapsules	Pacman should try to consume capsules and eat scared ghosts to score maximum points. Therefore, this feature has been included as an inverse. States with few capsules typically lead to higher score, as pacman can now consume scared ghosts.
sumDistToFood	Pacman should try to go to states where it's closer to food. Therefore, it should minimize it's distance from food. Being closer to food results in higher scores, when this feature is included as an inverse.
minDistToScaredGhost	Pacman should go closer to scared ghosts to score higher points.

Weights for all features are almost in same range, as I think that all of them are almost equally important. Only weight for $minDistToScaredGhost$ is large, as for smaller values, pacman prefers to eat food before eating ghosts, which results in lower scores. For other features, increasing or

decreasing weights of one of the features results in pacman behave in a sub-optimal way. For eg. increasing weights of numFood, causes pacman to ignore capsules.

Features that did not contribute to improve score.

1. MinDistToNormalGhost : The intuition for including this feature was that pacman should try to avoid getting eaten by a ghost by being away from it. Including this feature certainly resulted in higher winning rates (100%), but it also caused pacman to go around the grid to avoid being too close to a normal ghost, which resulted in lower average points. Therefore, to achieve maximum score, I decided to remove this feature from Evaluation function.

2. MinDistanceFromWalls : During many games, it was observed that pacman will thrash near a wall, or a closed grid with 2 or 3 sides. Therefore this feature was added to prevent pacman from thrashing in enclosed spaces, or near walls. However, adding this feature did not help. Infact, it prevented pacman from eating capsules, which are generally in closed spaces.