CPTS 570 Machine Learning, Fall 2016
Homework #5
Due Date: Nov 29

1. (**40 points**) Implementation of Expectation Maximization (EM) algorithm and experimentation.

   You will implement the EM algorithm and Gaussian Mixture Models (GMMs) to cluster the data into k clusters. As we discussed in the class, we assume a GMM with k components for the data and we find the parameters of the model using maximum likelihood estimation.

   Implement the EM algorithm for one-dimensional GMMs (assume that each data point has only one feature). You are provided with one-dimensional dataset. Use the algorithm to cluster the data. Run the algorithm multiple times from a number of different initialized values (random) and pick the one that results in the highest log-likelihood.

   Run the algorithm for different values of k (3, 4, 5) and report the parameters you get for each value of k.

   You can use WEKA (http://weka.sourceforge.net/doc.dev/weka/clusterers/EM.html) to debug your implementation.

2. (**10 points**) Devise two example tasks of your own that fit into the reinforcement learning framework, identifying for each its states, actions, and rewards. Make the two examples as different as possible.

3. (**50 points**) Implementation of Q-Learning algorithm and experimentation.

   You are given a Gridworld environment that is defined as follows:

   State space: GridWorld has 10x10 = 100 distinct states. The start state is the top left cell. The gray cells are walls and cannot be moved to.

   Actions: The agent can choose from up to 4 actions (left, right, up, down) to move around.

   Environment Dynamics: GridWorld is deterministic, leading to the same new state given each state and action

   Rewards: The agent receives +1 reward when it is in the center square (the one that shows R 1.0), and -1 reward in a few states (R -1.0 is shown for these). The state with +1.0 reward is the goal state and resets the agent back to start.

   In other words, this is a deterministic, finite Markov Decision Process (MDP). Assume the discount factor β=0.9.

   Implement the Q-learning algorithm (slide 46) to learn the Q values for each state-action pair. Assume a small fixed learning rate α=0.01.

   Experiment with different explore/exploit policies:

   1) ϵ-greedy. Try ϵ values 0.1, 0.2, and 0.3.

   2) Boltzman exploration. Start with a large temperature value T and follow a fixed scheduling rate. Give these details in your report.

   How many iterations did it take to reach convergence with different exploration policies?

   Please show the converged Q values for each state-action pair.

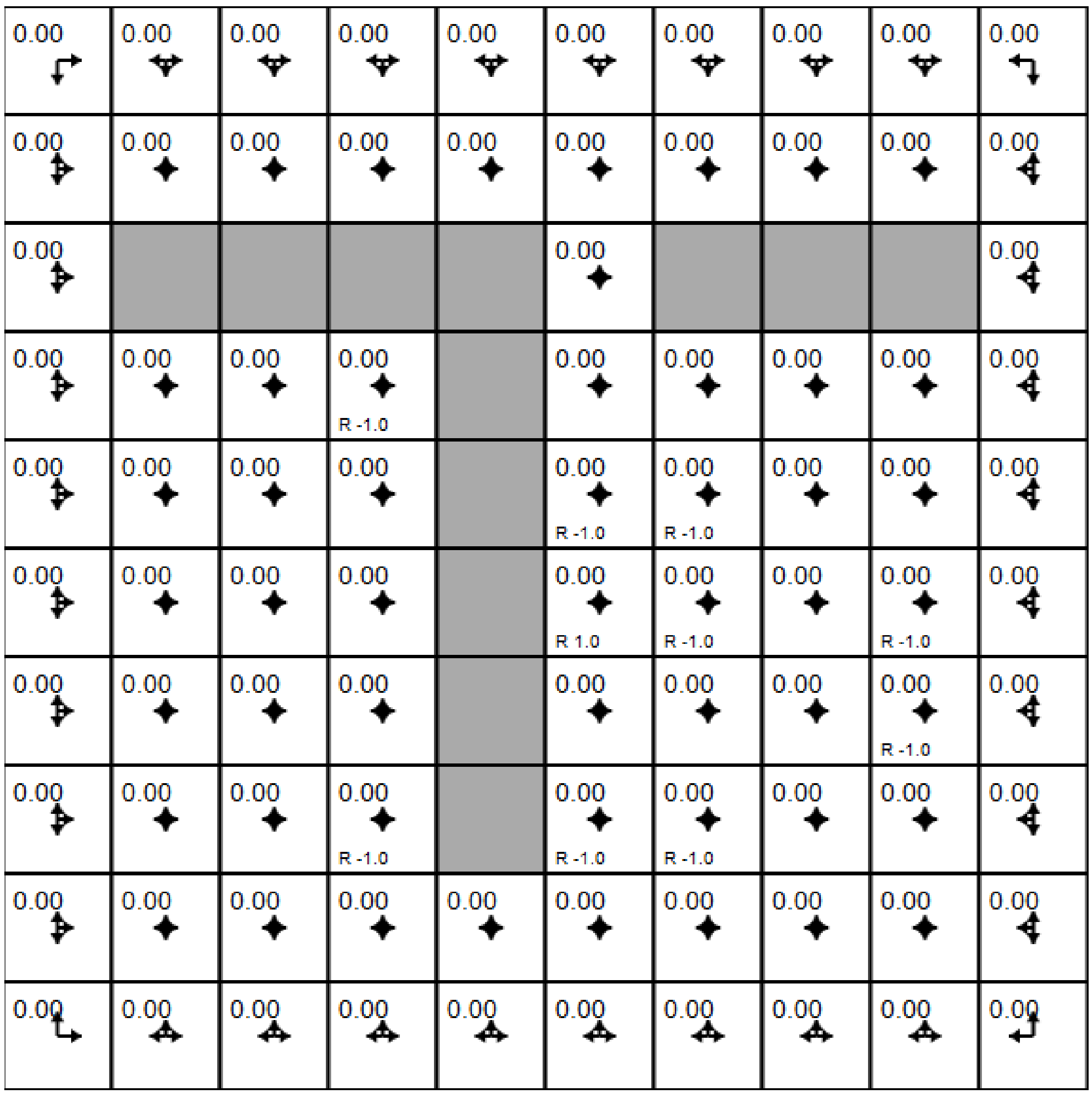

Figure 1: Grid world domain with states and rewards.

4. Additional instructions for submitting the programming code. This will greatly help the TA in grading your code.

   1. Please use the data files in the given format.

   2. Please print only the desired outputs.

   3. Please mention the python version in your code. Without the version information, it's hard to run the code.

   4. Please avoid submitting the homework in iPython Notebook.

   5. For C/C++ code, please provide the Makefile/run command with proper parameters.

   6. Please provide one example to run the code in a stand-alone mode with some valid parameters. For example, python decisiontree.py -dataDirectory ~/HW/data/