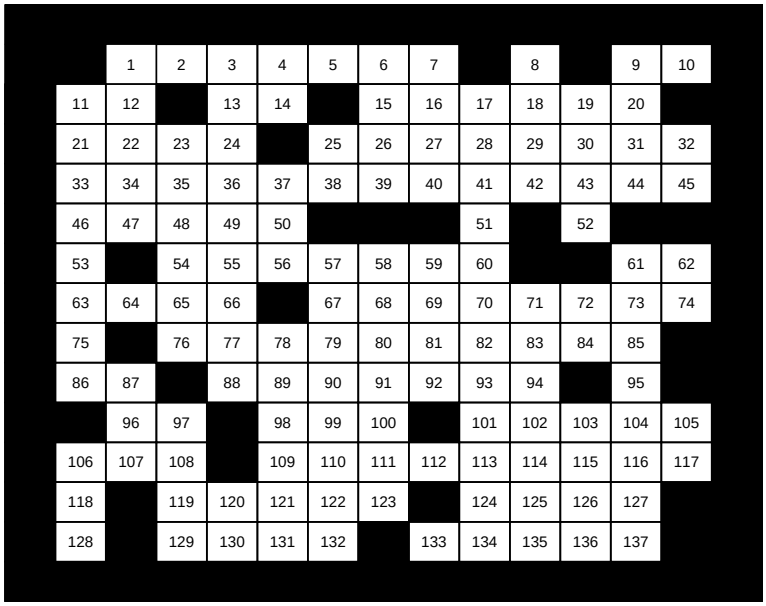
COMP 652 - Homework 6

Assigned: Nov 19, 2008
Due: Nov 26, 2008
Late: Nov 28, 2008
⇒ Turn in on paper only, please. Not electronically!

1. Reinforcement learning (10 pts.)



Consider the "maze" above. It pictures a Markov decision process (MDP) with 137 states. The rules of this process are as follows:

- The agent always starts in state 1.
- There are no terminal states.
- There are four actions in every state: up, right, down, left.
- When taking an action, the agent moves to the adjacent square if it is open, otherwise the agent stays in place. (E.g. Going down from state 1 takes the agent to state 12, but going up leaves it in state 1.)
- After each action, the reward depends on the state the agent moves to (or stays in). It does not depend on the previous state, nor the action taken. (E.g. Going up or left from state 1, or left from state 2, or up from state 12, all result in the agent being in state 1, and thus all result in the same immediate reward.)
- As this is not an episodic task, future rewards are to be discounted by $\gamma = 0.9$.

We will assume the rewards obtained upon entering each state are unknown to you. You will implement the Q-Learning algorithm to learn an optimal policy for navigating this maze.

Recall that Q-Learning maintains a table $Q(s, a)$ of values for every state s and action a. It updates these values incrementally based on experience. In particular, the Q-learning update when the state is s, the agent takes action a, receives reward r, and comes to state s' is:

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a'))$$

where α is a step size parameter. In class I mentioned how, if α is taken to zero over time in an appropriate fashion, and if the agent tries out every (s, a) pair infinitely many times, the Q estimates converge to the Q-values (usually denoted Q^*) of the/an optimal policy.

The file "HW6_Q1_Traj.txt" contains a trajectory $s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_{4885}, a_{4885}, r_{4885}, s_{4886}$, with each state, action, and reward in sequence on a separate line. This trajectory contains at least one occurrence of every state-action pair.

Write code that initializes a Q-table to zero, and then repeatedly runs through the experience in the trajectory performing the Q-update above, until the values in the Q-table converge. To be more precise, each "run" through the experience comprises updates based on the quadruples (s_1, a_2, r_1, s_2), (s_2, a_2, r_2, s_3), $\ldots$, $(s_{4885}, a_{4885}, r_{4885}, s_{4886})$. You will have to perform multiple such "runs" through the trajectory to get the Q-table to converge.

In this task, because all transitions and rewards are deterministic, you can actually use a step size of $\alpha = 1$ for each update. (Though you should feel free to try other values if you want to.) Determining when the stop updating the Q-table is up to you. State whatever stopping criterion you choose.

Report your code, print out the Q-table (which should hold Q^*), and determine the/an optimal (deterministic) policy. (Reminder: If you have computed Q^*, then an optimal policy π^* satisfies $\pi^*(s) \in \arg\max_a Q^*(s, a)$ for all states s.) Finally describe what state sequence results when the optimal policy is applied starting at state 1.

P.S. If you want to display your Q-table or optimal policy in a grid-shape corresponding to the maze, the file "HW6_Q1_StateCoords.txt" may be useful. On each line it lists a state number, the row of that state in the grid, and the column of that state in the grid.

2. Hidden Markov models (10 pts.)

Consider the data in file "HW6_Q2_Data.txt". There are three lines of numbers. Each line contains 200 numbers, each from the set $\{1, 2, 3, 4, 5\}$.

For this problem, imagine that the numbers $\{1, 2, 3, 4, 5\}$ may come from one of two different sources, A and B. In the first line of data, all the numbers are generated from source A. Each number is generated independently and randomly, based on a discrete distribution over $\{1, 2, 3, 4, 5\}$ that is specific to source A. Likewise, the second line of numbers is generated from source B, but according to a different discrete distribution.

A. (2 pts.) Compute the maximum likelihood estimates for A's distribution and for B's distribution, based on the data on lines 1 and 2 respectively. Report the distributions and show your code.

The third line of numbers was generated from a mixture of A's and B's distributions in the following way. The first number was chosen to come from A's or B's distribution with equal probability. Subsequent numbers were generated in the following way. For each $i = 2, 3, 4, \ldots, 200$, number i was generated from the same distribution (A's or B's) as number $i-1$ with probability 0.9. Otherwise (with probability 0.1) the other distribution was used.

One way of modeling the process of generated the third line of numbers is as an HMM. Specifically, the HMM has two states, $S = \{a, b\}$ and observation set $O = \{1, 2, 3, 4, 5\}$. In state a, the HMM generates an observation o according to the distribution of source A. Similarly, in state b, the observation is generated according to the distribution of source B. The start state probabilities are $p_a = p_b = 0.5$. The transition probabilities are $p_{aa} = p_{bb} = 0.9$, and $p_{ab} = p_{ba} = 0.1$.

B. (4 pts.) Implement the forward-backward algorithm and use it, along with the 3rd line of numbers and the HMM model described above, to compute the probabilities of each possible state at each time: $P(S_t = s|o_1, \ldots, o_{200})$, for all $t = 1, 2, 3, \ldots, 200$ and all $s = a, b$. Graph these probabilities as a function of time, t, and turn in your code.

C. (4 pts.) Finally, imagine that we know the third line of numbers comes from a mixture of A's and B's distributions, but we don't know the start probabilities or transition probabilities for the HMM state. (We will assume, however, the we know the observation probabilities, simply taking them to be as computed in part (A).) Suppose we initially take $p_a = p_b = 0.5$ and $p_{aa} = p_{ab} = p_{ba} = p_{bb} = 0.5$. Implement the Baum-Welch reestimation algorithm (except do not update the observation probabilities), and use it to fit the start state distribution and state transitions of the HMM based on the third line of numbers. Report the optimized probabililities $(p_a, p_b, p_{aa}, p_{ab}, p_{ba}, p_{bb})$. Using these optimized probabilities, recompute $P(S_t = s|o_1, \ldots, o_{200})$, as in part (B). How do the new $P(S_t = s|o_1, \ldots, o_{200})$ compare with the old ones? Finally, turn in your code.