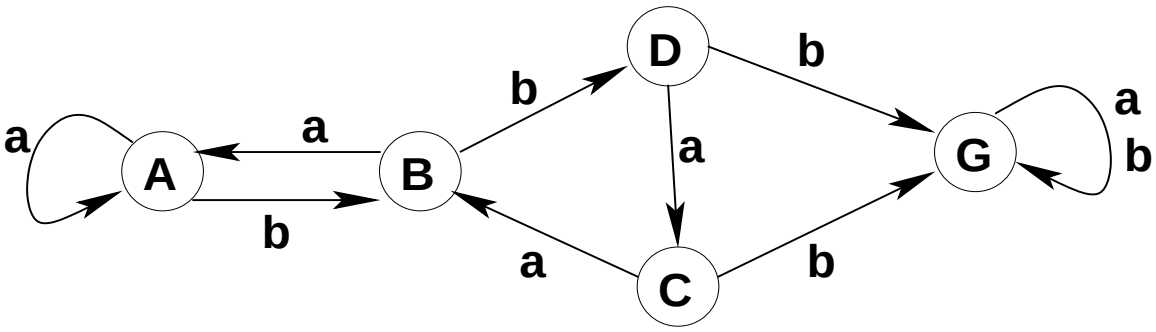
COS 402: Artificial Intelligence

Homework #6 — Fall 2008
Cat and mouse — Due: Thursday, December 11

Part I: Written Exercises

See instructions on the assignments page on how and when to turn these in. Approximate point values are given in parentheses. Be sure to show your work and justify all of your answers.

1. (15) Consider the following MDP:



There are five states: A, B, C, D and G. The reward at every state is -1, except at G where the reward is 0. There are two actions, a and b, and the effect of each action is deterministic as indicated in the figure. For instance, executing a in state B leads to state A. Assume $\gamma = 1$ in this problem.

a. Show the sequence of utility estimates U_i that would result from executing value iteration on this MDP. Also show the optimal policy that is computed using the final utility estimate.

b. Show the sequence of policies π_i and utility estimates U^{π_i} that would result from executing policy iteration on this MDP. Assume that you start with a policy that assigns action a to every state. Note that U^{π_i} will be infinite for some states. Also, assume that all ties between the actions a and b in the policy improvement step are always broken in favor of a.

c. Generalizing this example, suppose we are given a graph with a distinguished node (i.e., state) G, and k edges emanating from every node corresponding to k (deterministic) actions. As in this example, all of the edges emanating from G are self-loops, the node G is assigned reward 0, and all other nodes are assigned reward -1. In terms of properties of the graph, what is the optimal utility function U^*, and what is the optimal policy π^*? If value iteration is applied to this graph (viewed as an MDP), exactly how many iterations will be needed until the algorithm converges? How about for policy iteration?

2. (10) Sometimes MDP's are formulated with a reward function $R(s, a)$ that depends on the action taken, or a reward function $R(s, a, s')$ that also depends on the outcome state. For each of these formulations, show how to appropriately modify each of the following:

- the Bellman equation (Eq. (17.5) in R&N);
- the formula for converting the optimal utility U^* (denoted simply U in R&N) into an optimal policy π^* (Eq. (17.4) in R&N);
- the value iteration algorithm;
- the policy iteration algorithm.

3. (15) Let $B(U)$ and $\|\cdot\|_\infty$ be as defined in class. (This is the same as BU and $\|\cdot\|$ defined in Section 17.2 of R&N.) The purpose of this exercise is to prove that B is a *contraction*, i.e., that $\|B(U) - B(U')\|_\infty \leq \gamma \|U - U'\|_\infty$. As discussed in the book and lecture, this is the key step in showing that value iteration converges to the right answer.

We will begin by proving some basic facts. Be sure to give genuine mathematical proofs for each part of this problem. Also, your proofs should use elementary facts — in other words, do not give proofs that rely on mathematical sledge-hammers like the Cauchy-Schwartz inequality.

a. Let $u_1, \ldots, u_n$ and $v_1, \ldots, v_n$ be any sequences of real numbers. Prove that if $u_i \leq v_i$ for all i then

$$\max_i u_i \leq \max_i v_i.$$

b. Let $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ be any sequences of real numbers. Prove that

$$\left(\max_i x_i\right) - \left(\max_i y_i\right) \leq \max_i (x_i - y_i),$$

and also that

$$\max_i (x_i - y_i) \leq \max_i |x_i - y_i|.$$

(Hint: both of these inequalities can be proved using part (a) for an appropriate choice of u_i and v_i.)

Finally, use these facts to prove that

$$\left|\left(\max_i x_i\right) - \left(\max_i y_i\right)\right| \leq \max_i |x_i - y_i|.$$

c. Let $x_1, \ldots, x_n$ be any real numbers, and suppose that $p_1, \ldots, p_n$ are nonnegative real numbers such that $\sum_i p_i = 1$. Use the fact that $|a + b| \leq |a| + |b|$ for any real numbers a and b to prove that

$$\left|\sum_i p_i x_i\right| \leq \max_i |x_i|.$$

d. Now let s be any state, and let $(B(U))(s)$ denote the value of $B(U)$ at state s. By plugging in the definition of B, and using the properties proved above, prove that

$$|(B(U))(s) - (B(U'))(s)| \leq \gamma \|U - U'\|_\infty.$$

Conclude that

$$\|B(U) - B(U')\|_\infty \leq \gamma \|U - U'\|_\infty.$$

4. (15) In class, we looked at the following dataset:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	y
1	1	0	0	0	1	0	1	1
1	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0
0	1	0	0	1	0	0	0	0
1	0	0	0	0	1	0	1	0
0	1	1	0	1	1	0	1	1
1	1	0	1	0	1	0	1	1
0	1	0	1	0	1	0	0	1
0	0	0	0	0	0	1	1	0

In this formulation, there are eight attributes (or features or dimensions), $x_1, \ldots, x_8$, each taking the values 0 or 1. The label (or class) is given in the last column denoted y; it also takes the values 0 or 1. In class, it was noticed that the label y is 1 if and only if x_2 and x_6 are both equal to 1. Since attributes and labels are $\{0, 1\}$-valued, we can write this rule succinctly as $y = x_2 x_6$. In general, such a product of any number of attributes is called a *monomial*. (This includes the "empty" monomial, which, being a product of no variables, is always equal to 1.)

Throughout this problem, you can assume that the attributes and labels are all $\{0, 1\}$-valued. Also, let n be the number of attributes (for instance, $n = 8$ in the example above).

a. Describe a simple algorithm that, given a dataset, will efficiently (in polynomial time) find a monomial consistent with it, assuming that one exists.

b. What is the total number of monomials that can be defined on n attributes?

c. Suppose you applied your algorithm to the dataset above, and that a consistent monomial was found. Use the bound derived in class (or the results in R&N) to compute an upper bound on the generalization error of this monomial. ("Generalization error" is the same as what R&N calls simply the "error" in Section 18.5.) Derive a bound that holds with 95% confidence (so that $\delta = 0.05$).

d. Continuing the last question in which your algorithm is applied to data with $n = 8$ attributes, how many training examples would be needed to be sure the generalization error of a consistent monomial is at most 10% with 95% confidence?

Part II: Programming

The programming part of this assignment is described on the course website at:
`http://www.cs.princeton.edu/courses/archive/fall08/cos402/assignments/mdp`