6.231 Dynamic Programming and Optimal Control
Midterm Exam II, Fall 2011
Prof. Dimitri Bertsekas

Problem 1: (50 points)

Alexei plays a game that starts with a deck consisting of a known number of "black" cards and a known number of "red" cards. At each time period he draws a random card and decides between the following two options:

(1) Without looking at the card, "predict" that it is black, in which case he wins the game if the prediction is correct and loses if the prediction is incorrect.

(2) "Discard" the card, after looking at its color, and continue the game with one card less.

If the deck has only black cards he wins the game, while if the deck has only red cards he loses the game. Alexei wants to find a policy that maximizes his probability of a win.

If we formulate this as a finite-horizon DP problem, the DP algorithm is

$$J^*(b,r) = \min_{p\in\{0,1\}} \left[p\frac{b}{b+r} + (1-p)\left(\frac{b}{b+r}J^*(b-1,r) + \frac{r}{b+r}\,J^*(b,r-1)\right)\right],$$

where b and r denote the numbers of black and red cards in the box, $p=0$ and $p=1$ correspond to options 1 and 2, respectively.

Now we wish to formulate this problem as an **infinite-horizon** DP problem.

(a) How can you formulate Alexei's problem as an equivalent SSP problem? Identify the state space, control space, termination state, transition probabilities, etc.

(b) Does there exist any improper policy?

(c) How many iterations does value iteration require to converge? Finite or infinite?

(d) Describe how policy iteration will work for this SSP problem.

(e) Can this problem be formulated into an equivalent discounted/average cost problem? Why?

Solution. (a) The state space consists of all possible pairs (b,r) plus a termination state t. The termination state is the state when Alexei decides to predict and ends the game. At each state (b,r), there are two controls: to predict and go to the termination state, or to discard and go to either

$(b, r-1)$ and $(b-1, r)$ with probabilities $r/(b+r)$ and $b/(b+r)$ respectively. Moreover, if $b+r=1$ and Alexie chooses to discard, he also moves to the termination state with a winning probability 0.

(b) There does not exist any improper policy. The reason is that no matter when you choose to predict, you always reach the termination state within $b+r$ steps. Thus all policies are proper.

(c) Starting with an arbitrary cost vector, the value iteration always terminates within $(b+r)$ iterations. The reason is that for states $(1,0)$ and $(0,1)$ the cost-to-go will become the optimal cost-to-go in the 1st iteration; the cost-to-go $J(1,1)$, $J(2,0)$ and $J(0,2)$ will become equal to the optimal in the 2nd iteration. If we argue in this way iteratively, we can show by induction that $J_k(b,r) = J^*(b,r)$ when $k = b+r$. Therefore for the problem with a given starting state (b,r), the value iteration converges in at most $b+r$ iterations.

(d) Starting with any policy μ, we can verify that $J_\mu(b,r) = b/(b+r)$ for all μ. By applying the 1-step lookahead policy iteration, we will obtain that any policy yield the same cost-to-go, i.e.,

$$TJ_\mu = J_\mu, \qquad \forall \mu.$$

Since the cost-to-go J_μ satisfies the Bellman equation, the policy iteration terminates. To conclude, the PI terminates in one step, and any stationary policy is an optimal policy for this SSP.

(e) No, this problem cannot be equivalently formulated as an equivalent discounted problem or an average cost problem. The objective is to maximize the winning probability. Adding discount to future winning probabilities will distorts the objective of the original problem. Also, if we formulate this as an infinite-horizon average discounted problem, the average cost-to-go per stage will be 0 and thus meaningless.

Problem 2: (50 points)

Within the context and notation of the standard n-state discounted MDP, consider the mappings H and $\hat{H}$ given by

$$H(i,u,J) = \sum_{j=1}^{n} p_{ij}(u)\big(g(i,u,j) + \alpha J(j)\big),$$

and

$$\hat{H}(i,u,J) = \sum_{j=1}^{n} p_{ij}(u)\Big(g(i,u,j) + \alpha \min\big\{c(j), J(j)\big\}\Big),$$

where $c(j)$ is a given scalar for each $j = 1, \ldots, n$. Consider the mappings T and T_μ corresponding to H,

$$(TJ)(i) = \min_{u \in U(i)} H(i,u,J), \qquad (T_\mu J)(i) = H\big(i, \mu(i), J\big),$$

and let $\hat{T}$ and $\hat{T}_\mu$ be the corresponding mappings for $\hat{H}$:

$$(\hat{T}J)(i) = \min_{u \in U(i)} \hat{H}(i,u,J), \qquad (\hat{T}_\mu J)(i) = \hat{H}\big(i,\mu(i),J\big).$$

(a) Show that (like H), $\hat{H}$ satisfies the Contraction and Monotonicity Assumptions of Section 1.6.

(b) Let J^* and $\hat{J}^*$ be the unique fixed points of T and $\hat{T}$, respectively. Show that if $c(j) \geq J^*(j)$ for all j, then $\hat{J}^* = J^*$.

(c) Consider the generalized PI algorithm using $\hat{H}$ instead of H. Show that the policy evaluation phase consists of solving an optimal stopping problem.

(d) Assume that $c(j) \geq J^*(j)$ for all j. Then, according to parts (a) and (b), the generalized PI algorithms using H and $\hat{H}$ yield J^* in a finite number of policy iterations. Which one will converge faster? What are some arguments for and against the use of $\hat{H}$ in place of H?

Solution. (a) The monotonicity of $\hat{H}$ can be proved easily since both linear mappings and minimization preserve the monotonicity. Now let us focus on the contraction property.

We have

$$|\min\{c(j), J_1(j)\} - \min\{c(j), J_2(j)\}| \leq |J_1(j) - J_2(j)|,$$

so

$$\begin{aligned}
|\hat{H}(i,u,J_1) - \hat{H}(i,u,J_2)| &= \left| \sum_{j=1}^{n} \alpha p_{ij}(u) \Big(\min\big\{c(j), J_1(j)\big\} - \min\big\{c(j), J_2(j)\big\} \Big) \right| \\
&\leq \alpha \sum_{j=1}^{n} p_{ij}(u) |J_1(j) - J_2(j)| \\
&\leq \alpha \max_j |J_1(j) - J_2(j)| \\
&= \alpha \|J_1 - J_2\|,
\end{aligned}$$

where $\|\cdot\|$ is the sup-norm. Thus $\hat{H}(i,u,J)$ is a contraction in J.

(b) If $c(j) > J^*(j)$ for all j, the vector J^* automatically satisfies the Bellman equation corresponding to $\hat{H}$. Then by using the uniqueness of the solution of the Bellman equation (by using the contraction property), we must have $\hat{J}^* = J^*$.

(c) The policy evaluation for a given μ can be considered as solving the following stopping problem: at each state i, either to continue with μ or to stop and incur a cost of $c(j)$ with probability $p_{ij}(u)$.

(d) The PI of the modified version using $\hat{H}$ converges faster than the PI using H. The reason is that, by placing an upperbound $c(j)$ for $J(j)$, we have ensured that the cost-to-go cannot go way off and thus will remain close to the optimal cost-to-go. However, the modified version involving the upperbound $c(j)$ may be computationally harder to implement, due to that the minimization operation can not be easily implemented with Monte Carlo sampling.

6.231 Dynamic Programming and Stochastic Control

Fall 2011