Topics in Artificial Intelligence: Dynamic Programming and Approximate Dynamic Programming.
CPSC 532M, Term 2, Winter 2007–2008
Assigned March 3. Due April 11.

Homework #1

1. **Formulating and Solving Dynamic Programming.** We have a widget making machine called the "widgetomatic" with which we hope to make our fortune. The widgetomatic has the following properties:
   - If it is brand new, it never breaks on its first day of operation.
   - After the first day of operation and if it is not broken at the start of the day, we can choose whether to oil the machine. If we oil it then it has a 20% chance of breaking during the day, otherwise it has a 40% chance of breaking during the day.
   - If we oil the machine and it operates all day we make \$900. If we do not oil the machine and it operates all day we make \$1000.
   - If we oil the machine and it breaks, we make (on average) \$450. If we do not oil the machine and it breaks, we make (on average) \$500.
   - If the machine is broken at the start of a day, we make nothing that day.
   - If the machine is broken at the start of the day, we can choose to repair it for \$500 and it will be operational the next day.
   - If the machine is broken at the start of the day, we can choose to replace it with a brand new machine for \$2000 and the new machine will be operational the next day.

   Formulate this system as a dynamic programming problem, and then answer the following questions.

   (a) Our widgets are important to the Stanley Cup playoffs, and so in order that they be distributed by the start of the playoffs we can only sell them for the next week. Find the optimal oiling, repair and replacement policy the maximizes total income over the next seven days, assuming a brand new machine on the first day.

   (b) How do you expect that this policy would change if we could sell the widgets all year long?

2. **Dynamic Programming Operator.** For an MDP model with states $x \in \{1, 2, \dots, n\}$, define the dynamic programming operator

$$(TJ)(x) = \min_{u \in \mathcal{U}} \sum_{y \in \mathcal{S}} p_{xy}(u)(g(x,u,y) + \alpha J(y))$$

   (a) Define the operator T_x such that

$$(T_x J)(y) = \begin{cases} (TJ)(y), & \text{if } y = x; \\ J(y), & \text{otherwise.} \end{cases}$$

   Then let $J_{k+1} = T_n T_{n-1} \cdots T_2 T_1 J_k$. Prove that $J_k \to J^*$.

   (b) Which converges faster: the scheme above, or $J_{k+1} = TJ_k$?

3. **Shortest Path Tour Problem.** You are given a set of nodes V and edges E that form a graph G. You are also given a set of node subsets T_i for $i = 1, \dots, N-1$ such that $T_i \subseteq V$. You wish to find the shortest path $p = (p_0, p_1, \dots, p_N)$ from a source $p_0 = s \in V$ to a target $p_N = t \in V$, where for simplicity we assume that each edge has unit cost. However, we impose the constraint that $p_i \in T_i$ for $i = 1, \dots, N-1$.

   (a) Formulate this constrained shortest path problem as a dynamic programming problem. In other words, specify the dynamics, actions and cost function.

   (b) Show that the solution of the constrained shortest path problem can be found by solving a sequence of unconstrained shortest path problems each of which has a single source and multiple destinations.

4. **Linear Quadratic Regulators.** You will probably want to use MATLAB's optimal control toolbox to answer the following questions. Be sure to give title your plots, label their axes, and distinguish the lines in plots with multiple lines. Some visualization commands that you might find useful are `plot`, `surf`, `title`, `xlabel`, `ylabel`, and `legend`.

   (a) Consider the discrete time system

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} +1 & -h \\ +h & +1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \frac{h}{2} \begin{bmatrix} \sqrt{3} \\ 1 \end{bmatrix} u \tag{1}$$

   with cost matrices

$$Q = Ih, \qquad Q_f = Ih, \qquad R = Ih,$$

   where I is the identity matrix of appropriate size. Using discrete time LQR with $h = 0.25$ and over time interval $[0, 10]$, find the optimal $P(t)$ and $K(t)$ and plot their components as separate lines with respect to time (similar to what appears in Boyd's LQR notes on page 1-27). Using a surface mesh plot, show $V(x, 0)$. For initial state $\begin{bmatrix} 0 & 1 \end{bmatrix}^T$, plot the open and closed loop trajectories in phase space. You should have a total of at least four separate plots.

   (b) Repeat the process above, showing the same plots, but with the continuous time system:

$$\frac{d}{dt} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ +1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \sqrt{3} \\ 1 \end{bmatrix} u \tag{2}$$

   and cost matrices

$$Q = I, \qquad Q_f = I, \qquad R = I.$$

   For your interest, the system (1) is generated by discretizing (2) using the simple Forward Euler scheme (with stepsize h) for numerical solution of ODEs. Unfortunately, your plots should show that the results are not exactly comparable—the open loop trajectory for the discretization is unstable, but the open loop trajectory for the continuous system is stable (it stays the same distance from the origin)—thus demonstrating once again the dangers of naive discretizations of continuous systems.

   It turns out that for (2) we can generate a discretization that maintains the stability properties of the open loop trajectories using the implicit trapezoidal scheme, which can be written as

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} +1 & +\frac{h}{2} \\ -\frac{h}{2} & +1 \end{bmatrix}^{-1} \left(\begin{bmatrix} +1 & -\frac{h}{2} \\ +\frac{h}{2} & +1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \frac{h}{2} \begin{bmatrix} \sqrt{3} \\ 1 \end{bmatrix} u \right).$$