1 Dynamic programming for a controlled Markov chain

Problem 1. We discuss the case of a linear system with quadratic cost and Gaussian noise, which is called the LQG problem. For simplicity, we consider only the scalar case. The system is

\[ X(n+1) = aX(n) + U(n) + V(n), \quad n \geq 0 \]

Here, \( X(n) \) is the state, \( U(n) \) is a control value, and \( V(n) \) is the noise. We assume that the random variables \( V(n) \) are i.i.d. and \( N(0, \sigma^2) \).

The problem is to choose, at each time \( n \), the control value \( U(n) \) in \( \mathbb{R} \) based on the observed state values up to time \( N \) to minimize the expected cost

\[
E\left[ \sum_{n=0}^{N} (X(n)^2 + \beta U(n)^2) \mid X(0) = x \right]
\]

Thus, the goal of the control is to keep the state value close to zero, and one pays a cost for the control. Then the stochastic dynamic programming equations are

\[
V_m(x) = \min_u \left\{ x^2 + \beta u^2 + E[V_{m-1}(ax + u + V)] \right\}, \quad m \geq 0
\]

Show that \( V_m \) is of the form \( V_m(x) = c(m) + d(m)x^2 \).

Solution. We prove by induction.

We have the basis \( V_0(x) = x^2 + \beta U(0)^2 \) which is of the right form.

Now suppose that \( V_{m-1}(x) = c(m-1) + d(m-1)x^2 \). Then

\[
V_m(x) = \min_u \left\{ x^2 + \beta u^2 + E[c(m-1) + d(m-1)(ax + u + V)^2] \right\}
\]

\[
= c(m-1) + x^2 + \min_u \left\{ \beta u^2 + d(m-1)E[(ax + u + V)^2] \right\}
\]

Since \( ax + u + V \sim N(ax + u, \sigma^2) \), we have

\[
E[(ax + u + V)^2] = \sigma^2 + (ax + u)^2 = \sigma^2 + a^2x^2 + 2axu + u^2
\]

hence

\[
V_m(x) = c(m-1) + x^2 + \min_u \left\{ \beta u^2 + d(m-1)(\sigma^2 + a^2x^2 + 2axu + u^2) \right\}
\]

\[
= c(m-1) + x^2 + d(m-1)(\sigma^2 + a^2x^2) + \min_u \left\{ (\beta + d(m-1))u^2 + 2d(m-1)axu \right\}
\]

The minimum is obtained for \( u = -\frac{d(m-1)a}{\beta + d(m-1)} \). Substituting in \( V_m(x) \), we see that it is of the form \( V_m(x) = c(m) + d(m)x^2 \).
Problem 2. There are two coins. One is fair and the other one has a probability of head equal to 0.6. You cannot tell which is which by looking at the coins. At each step \( n \geq 1 \), you must choose which coin to flip. The goal is to maximize the expected number of heads.

(a) Formulate the problem as a partially observed Markov decision problem (POMDP).

(b) Discretize the state of the system as we did in the searching for your keys example and write the SDPEs.

(c) Implement the SDPEs in Matlab and simulate the resulting system.

Solution.

(a) The state of the system is \( X(n) \), the probability that coin \( A \) is fair. The control actions are \( U(n) \in \{ A, B \} \) where \( U(n) = A \) means that one flips coin \( A \). The observation at time \( n \) is \( Y(n) = 1 \) if the coin flips yields head and \( Y(n) = 0 \) otherwise.

Say that you choose \( U(n) = A \) and observe \( Y(n + 1) = 1 \). Then

\[
X(n + 1) = \frac{P[A \text{ is fair}|X(n), Y(n + 1) = 1, U(n) = A]}{P[Y(n + 1) = 1, A \text{ is fair}|X(n), U(n) = A]}.
\]

Now, the numerator is equal to

\[
P[Y(n + 1) = 1|A \text{ is fair}]P[A \text{ is fair}|X(n)] = 0.5X(n).
\]

Also, the denominator is

\[
P[Y(n + 1) = 1, A \text{ is fair}|X(n), U(n) = A] + P[Y(n + 1) = 1, A \text{ is biased}|X(n), U(n) = A].
\]

Moreover,

\[
P[Y(n + 1) = 1, A \text{ is biased}|X(n), U(n) = A] = 0.6(1 - X(n)).
\]

Hence,

\[
X(n + 1) = \frac{0.5X(n)}{0.5X(n) + 0.6(1 - X(n))}.
\]

Similarly, one finds all the update equations:

\[
X(n + 1) = \begin{cases}
0.5X(n) & \text{if } Y(n) = 1, U(n) = A; \\
0.5X(n) + 0.4(1 - X(n)) & \text{if } Y(n) = 0, U(n) = A; \\
0.5(1 - X(n)) + 0.4X(n) & \text{if } Y(n) = 1, U(n) = B; \\
0.5(1 - X(n)) + 0.4X(n) & \text{if } Y(n) = 0, U(n) = B;
\end{cases}
\]

The problem is to maximize the expected number of heads

\[
\sum_{n=1}^{N} c(X(n), U(n))
\]

where

\[
c(X(n), U(n)) = \begin{cases}
1 & \text{if } U(n) = A \\
0.5X(n) + 0.6(1 - X(n)) & \text{if } U(n) = B;
\end{cases}
\]

\[
+ \sum_{n=1}^{N} 1 & \text{if } U(n) = B \text{ and } X(n) = 0;
\]

\[
0.5(1 - X(n)) + 0.6X(n) & \text{if } U(n) = B \text{ and } X(n) = 1;
\]
Let $V_n(x)$ be the maximum expected number of heads in $n$ flips, salting with a probability $x$ that coin $A$ is fair. The DPEs are

$$V_{n+1}(x) = \max_{u \in \{A,B\}} \{c(x,u) + p(x,u)V_n(g(x,u,1)) + (1 - p(x,u))V_n(g(x,u,0))\}$$

where

$$p(x,u) = P[Y(n + 1) = 1|X(n) = x, U(n) = u].$$

We find

$$p(x,A) = 0.5x + 0.6(1 - x) \text{ and } p(x,B) = 0.5(1 - x) + 0.6x.$$ (b) The code is

```matlab
X = 3000; % discretization points of x
N = 100; % number of steps
x = 0;
V = zeros(X, N);
U = V;
z = zeros(1,X);
for m =1:X
    z(m) = m/X;
end
for n = 1:N-1
    for m = 1:X
        x = m/X;
        GA1 = min(max(round(X*(0.5*x/(0.5*x + 0.6*(1 - x)))),1),X);
        GA0 = min(max(round(X*(0.5*x/(0.5*x + 0.4*(1 - x)))),1),X);
        A = 0.5*x + 0.6*(1 - x) + (0.5*x + 0.6*(1 - x))*V(GA1,n);
        A = A + (1 - 0.5*x - 0.6*(1 - x))*V(GA0,n);
        GB1 = min(max(round(X*(0.5*(1 - x)/(0.5*(1 - x) + 0.6*x)))),1,X);
        GB0 = min(max(round(X*0.5*(1 - x)/(0.5*(1 - x) + 0.4*x)),1),X);
        B = 0.5*(1 - x) + 0.6*x + (0.5*(1 - x) + 0.6*x)*V(GB1,n);
        B = B + (1 - 0.5*(1 - x) - 0.6*x)*V(GB0,n);
        V(m,n+1) = max(A,B);
        U(m, n) = (A > B);
    end
end
plot(z,U(:,99))
```

Figure 1 shows the results. One should flip the coin that is most likely to be biased. Thus, the optimal strategy is myopic.
Figure 1: Problem 12.3: One should flip coin A if it is more likely to be biased than coin B.

2 Sufficient statistics

Problem 3. Let $X = (X_1, \cdots, X_n)$ be a sequence of i.i.d. r.v. and $h(X) = X_1 + \cdots + X_n$.
(a) Show that $h(X)$ is sufficient for $p$ when $X_1, \cdots, X_n \sim \text{Bernoulli}(p)$, and find $\text{MLE}[p|X]$
(b) Show that $h(X)$ is sufficient for $\theta$ when $X_1, \cdots, X_n \sim N(\theta, 1)$, and find $\text{MLE}[\theta|X]$

Solution.
(a) We compute $P[X = x | p]$ and show it has the right form:

$$P[X = x | p] = p^{x_1} (1 - p)^{1-x_1} \cdots p^{x_n} (1 - p)^{1-x_n} = p^{x_1 + \cdots + x_n} (1 - p)^{n-(x_1 + \cdots + x_n)} = p^{h(x)} (1 - p)^{n-h(x)} = f(p, h(x))$$

The MLE is

$$\text{MLE}[p | X = x] = \arg\max_p p^{h(x)} (1 - p)^{n-h(x)}$$

$$= \arg\max_p p^{h(x)} \log p + (n - h(x)) \log(1 - p)$$

The optimality condition is given by

$$0 = \partial_p (p^{h(x)} \log p + (n - h(x)) \log(1 - p)) = \frac{h(x)}{p} - \frac{n - h(x)}{1 - p}$$

with solution $p = h(x)/n$. Hence $\text{MLE}[p | X = x] = \frac{h(x)}{n} = \frac{x_1 + \cdots + x_n}{n}$

(b) We have

$$P[X = x | \theta] = f_{X_1}(x_1) \cdots f_{X_n}(x_n)$$

$$= K e^{-(x_1 - \theta)^2/2} \cdots e^{(x_n - \theta)^2/2}$$

$$= K \exp\left(-\frac{1}{2} \sum_i (x_i - \theta)^2\right)$$

$$= K \exp\left(-\frac{1}{2} \sum_i x_i^2 + \theta \sum_i x_i - \frac{n}{2} \theta^2\right)$$

$$= K \exp(-\|x\|^2/2) \exp(\theta h(x) - n\theta^2/2)$$

The MLE is $\arg\max_{\theta} \{\theta h(x) - n\theta^2/2\} = h(x)/n$