Appendix B

Problem Sets

B.1 Problem Set I

(due Wed. Sept. 23)[1]

Problem B.1.1. Background reading: To start preparing for the block of lectures on partial information problems, please start read Rabiner's tutorial paper on Hidden Markov Models [Rab89], sections I-III (and the rest if you are interested). This paper considers only estimation problems, but we'll build on it to solve control problems. Moreover, it's also good to remind you about modeling using Markov chains.

Question: Look at displays (18),(19),(20) in that paper. Explain how these equations can be interpreted as a DP algorithm [hint: you might have to replace the operations max and + in the DP algorithm seen in class by some other operations].

Problem B.1.2. Do the exercises in the class notes.

Problem B.1.3. Consider a sequence of matrices $M_1, M_2, \ldots, M_N$, where M_k has n_k rows and n_{k+1} columns. The problem is to choose the order for multiplying the matrices that minimizes the number of scalar multiplications needed to compute the product $M_1 M_2 \ldots M_N$. Assume that the matrices are multiplied the usual way, so that the multiplication $M_1 M_2$ for example involves $n_1 n_2 n_3$ multiplications. So for example, if $n_1 = 1, n_2 = 10, n_3 = 1, n4 = 10$, computing $M_1 M_2 M_3$ in the order $((M_1 M_2) M_3)$ requires 20 scalar multiplications, but $(M_1 (M_2 M_3))$ requires 200 scalar multiplications.

1. Formulate the problem in a form suitable for the application of the DP algorithm (define a state, control, stage cost and terminal cost, etc.). Then use the DP algorithm to find the optimal order in which to multiply the matrices when $N = 4$ and $(n_1, n_2, n_3, n_4, n_5) = (10, 30, 70, 2, 100)$.

[1] this version: Sept. 11 2009

2. Using the same numerical values as in the previous question, solve the problem where the objective is instead to maximize the number of multiplications. What is the ratio of the maximum to minimum number of multiplications?

Problem B.1.4 (Markov chains basics)**.**

1. (gambler's ruin) A basic analysis tool for Markov chains is a technique called *first-step analysis*, which works recursively rather like dynamics programming. Consider the 1D-symmetric random walk. That is, we have a sequence $\{X_i\}_{i\geq 0}$ of i.i.d. random variables with $P(X_i = 1) = P(X_i = -1) = \frac{1}{2}$, and we define the partial sum $S_n = S_0 + X_1 + \ldots + X_n$, where S_0 is some arbitrary integer representing the initial position (or the initial wealth). Given two integers $A, B \geq 0$, consider the first time τ at which the partial sum S_n reaches level A or $-B$:

$$\tau = \min\{n \geq 0 : S_n = A \text{ or } S_n = B\}.$$

   We wish to determine the probability that a gambler starting with 0 dollars wins A dollars before losing B dollars. Since $S_\tau = A$ or $S_\tau = -B$, we want to find $p = P(S_\tau = A | S_0 = 0)$.

   a) Define $f(k) = P(S_\tau = A | S_0 = k)$ for $-B \leq k \leq A$. Now find a recurrence relation between $f(k-1), f(k)$, and $f(k+1)$ for $-B < k < A$.

   b) Determine the values $f(A)$ and $f(-B)$. Finally, compute p.

2. Consider a homogeneous Markov chain over a *finite* state space S. Let P be its transition matrix, i.e. $P_{ij} = P(X_{t+1} = j | X_t = i)$. Show that P has at least one eigenvalue equal to 1. Deduce from this that P has a stationary distribution.

3. Compute the stationary distribution of the Markov transition matrix

$$\begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix},$$

   with $(\alpha, \beta) \in (0, 1)$.

4. Give i) an example of a Markov chain with multiple stationary distributions; ii) an example of a Markov chain with no stationary distribution.

5. (random walk on a graph) Consider a finite undirected graph $G = (V, E)$. Let d_i be the degree of node i, i.e., the number of edges incident to node i or equivalently, the number of neighbor nodes of i. Assume that there is no isolated node, i.e., $d_i > 0$ for all i. Transform this graph into a directed graph by splitting each edge into two oriented edges of opposite directions. Now consider this directed graph as the transition graph of a Markov chain: node i represents state i, and the probability of transition

from state i to state j is equal to $1/d_i$, for all i, j. Show that a stationary distribution for the Markov chain defined by this transition graph is given by the vector π with coordinates

$$\pi_i = \frac{d_i}{\sum_{j \in V} d_j}.$$

Problem B.1.5 (the parking problem)**.** A driver is looking for parking on a one-way road toward his destination. Each parking place is free with probability p independently of whether other parking places are free or not. The driver cannot observe whether a parking space is vacant until he reaches it. If he reaches a vacant space, he can either park or continue driving to the next space. He cannot return to vacant spaces that he has passed. If he parks k places from his destination, he incurs a cost k. If he reaches the destination without having parked the cost is C.

1. Formulate this problem as an MDP and write the DP algorithm. Let $F(k)$ be the minimal expected cost if he is k parking places from his destination (but has not yet observed if the space is vacant or not), where $F(0) = C$. Show that

$$F(k) = p \min\{k, F(k-1)\} + qF(k-1), k = 1, 2, \ldots,$$

   where $q = 1 - p$.

2. Show that an optimal policy is of the form: never park if $k \geq k^*$, but take the first free place if $k < k^*$, where k is the number of parking places from the destination and k^* is the smallest integer i satisfying $q^{i-1} < 1/(pC + q)$.

Problem B.1.6. Consider the inventory problem for the case where the cost has the general form

$$E\left\{\sum_{k=0}^{N} r_k(x_k)\right\},$$

where the functions r_k are convex and $r_k(x) \to \infty$ as $|x| \to \infty$, for $k = 0, \ldots, N$.

1. Assume that the fixed cost is zero. Write the DP algorithm for this problem and show that the optimal ordering policy has the same form as the one derived in class.

2. Suppose there is a one-period time lag between the order and the delivery of inventory. That is, the system equation is of the form

$$x_{k+1} = x_k + u_{k-1} - w_k, \ k = 0, \ldots, N-1,$$

   where u_{-1} is given. Reformulate the problem so that it has the form of the problem of question 1 [hint: make a change of variable $y_k = x_k + u_{k-1}$].

Problem B.1.7 (Output Tracking with Constrained Control)**.** Consider the linear dynamics system

$$x_{k+1} = Ax_k + Bu_k,\ k = 0, 1, \ldots, N-1,$$
$$y_k = Cx_k,\ k = 0, 1, \ldots, N,$$

with state $x_k \in \mathbb{R}^n$, input $u_k \in \mathbb{R}^m$, and output $y_k \in \mathbb{R}^p$. Let $x_0 = 0$. We want to choose the input sequence $u_0, \ldots, u_{N-1}$ to minimize the output tracking cost

$$J = \sum_{k=1}^{N} \|y_k - \hat{y}_k\|^2,$$

subject to the constraint $\|u_k\|_\infty \leq U^{\max},\ k = 0, \ldots, N-1$. Here $\hat{y}_1, \ldots \hat{y}_N$ is a given desired trajectory to track. Take the following data

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0.5 \\ 1 \end{bmatrix}, C = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix},$$

$N = 100$, and $U^{\max} = 0.1$. The desired output trajectory is

$$\hat{y}_k = \begin{cases} 0 & \text{for } 0 \leq k < 30, \\ 10 & \text{for } 30 \leq k < 70, \\ 0 & \text{for } 70 \leq k \leq 100. \end{cases}$$

1. Since this is a deterministic optimal control problem, we can use optimization techniques to find an open-loop policy, and the best open-loop and closed-loop policies have the same performance. Formulate the problem of finding the inputs $u_0, \ldots, u_{N-1}$ (open-loop) in the particular problem above as a *convex* optimization problem, discussing briefly why it is convex. Then find numerically the optimal inputs $u^* = \{u_0^*, \ldots, u_{N-1}^*\}$ and the optimal cost $J^*(0)$ [use an optimization package, e.g. CVX]. Plot the trajectories $\{\hat{y}_k\}_{0 \leq k \leq N}$, and $\{y_k\}_{0 \leq k \leq N}$.

2. Change the dynamics equation to

$$x_{k+1} = Ax_k + Bu_k + w_k,$$

keeping the same numerical data as above, and with w_k i.i.d. zero-mean Gaussian with covariance matrix $10^{-6} I_3$, where I_3 is the 3×3 identity matrix. Using the same inputs u^* as computed in the previous question (which ignored any process noise), run 50 simulations of the system and plot the trajectories $\{y_k\}_{0 \leq k \leq N}$. Discuss qualitatively the performance of the open-loop policy for the system with disturbances.

3. Discuss briefly how you would try to solve the problem of question 2 using dynamic programming (note: recall that we have a control constraint here, this is not a standard LQR problem).