Lecture 7 — February 6

Lecturer: Venkat Anantharam *Scribe: Ching-Ling Huang*

7.1 DP is not a panacea

Here we given an example of a problem where the DP approach is not smart. There are N jobs waiting for service at a processor, and we have to decide the order in which to serve the jobs. The objective to minimize is

$$E[\sum_{i=1}^{N} \alpha^{-t_i} R_i] \tag{7.1}$$

where $\alpha > 0$, t_i is the time at which job#i gets finished, and R_i is the reward associated with job#i. Hence $(t_1, t_2, ..., t_N)$ depends on the order of service. Let $T_1, T_2, ..., T_N$ be independent random variables, where T_i represents the time required to serve job#i. For example, for $N = 3$ and service order (1,2,3):

$$t_1 = T_1 \tag{7.2}$$

$$t_2 = T_1 + T_2 \tag{7.3}$$

$$t_3 = T_1 + T_2 + T_3 \ . \tag{7.4}$$

However, for service order (2,3,1):

$$t_1 = T_2 \tag{7.5}$$

$$t_2 = T_2 + T_3 \tag{7.6}$$

$$t_3 = T_2 + T_3 + T_1 \ . \tag{7.7}$$

We think of the rewards R_i as being deterministic.

This problem can be solved by Dynamic Programming. Define the state as the identities of jobs already served and the total time elapsed. Use the DP recursion to find the optimal order.

However, in this example we observe that using DP is unnecessary complicated. A simple trick is to use an interchange argument. In this simple example, given the service order

$$i_1, ..., i_{k-1}, i, j, i_{k+1}, ..., i_N \ , \tag{7.8}$$

let's consider the service order (switching i and j)

$$i_1, ..., i_{k-1}, j, i, i_{k+1}, ..., i_N \ . \tag{7.9}$$

The only difference between the two orders is the realized departure times of job#i and job#j. The difference in net reward between the former and the latter service orders is

$$\alpha^{-(T+T_i)} R_i + \alpha^{-(T+T_i+T_j)} R_j - \left[\alpha^{-(T+T_j)} R_j + \alpha^{-(T+T_j+T_i)} R_i\right] \ , \tag{7.10}$$

where T is the time at which job$\#i_{k-1}$ departs. We would pick i before j if and only if the expectation of this difference in rewards is nonnegative. Thus the optimal strategy is to pick i before j if

$$\frac{E[\alpha^{-T_i}]R_i}{1-E[\alpha^{-T_i}]} > \frac{E[\alpha^{-T_j}]R_j}{1-E[\alpha^{-T_j}]} \tag{7.11}$$

Such a rule is called an index rule.

Message of this example: Dynamic Programming can be the dumb way sometimes.

7.2 Linear Dynamical Systems

In the next few lectures, we introduce some major results with many applications: the solution of the linear Quadratic (LQ) control problem, the Kalman filter, the separation principle, etc. A finite horizon discrete time finite dimensional stochastic linear dynamical system model is given by

$$x_{k+1} = A_k x_k + B_k u_k + w_k. \quad k = 0, 1, \ldots N-1 \ . \tag{7.12}$$

where $x_k \in R^n$, $u_k \in R^m$, and $w_k \in R^n$. x_k is the state, u_k is the control, and w_k is the randomness, representing uncertainty or noise. Matrices A_k and B_k are deterministic matrices of appropriate dimensions. In cases where the state x_k is not directly available, we typically model the observations y_k by

$$y_k = C_k x_k + v_k \tag{7.13}$$

where $y_k \in R^p$ and v_k is random, representing the measurement noise. The noise terms $w_k, k = 0, 1, \ldots, N-1$, and $v_k, k = 0, 1, \ldots, N-1$, are assumed to be independent.

Now consider the deterministic state evolution equation

$$x_{k+1} = A_k x_k + B_k u_k \ , \tag{7.14}$$

with some initial condition x_0. Then x_k can be expressed in terms of x_0 and the controls by writing

$$x_1 = A_0 x_0 + B_0 u_0 \tag{7.15}$$

$$x_2 = A_1 x_1 + B_1 u_1 = A_1 A_0 x_0 + A_1 B_0 u_0 + B_1 u_1 \ , \tag{7.16}$$

and so on. An important special case is the time-invariant case, i.e. $A_k = A$, $B_k = B$ for all k. Then

$$x_k = A^k x_0 + \sum_{i=0}^{k-1} A^{k-i-1} B u_i \ . \tag{7.17}$$

Similarly, for the observations, in the absence of measurement noise, in the time-invariant case, ($A_k = A$, $B_k = B$, $C_k = C$ for all k), we have

$$y_{k+1} = C x_k \ , \tag{7.18}$$

and so

$$y_k = CA^k x_0 + \sum_{i=0}^{k-1} CA^{k-i-1} B u_i \ . \tag{7.19}$$

Recall that every $n \times n$ matrix A can be written, as complex matrix, as

$$A = T^{-1} J_A T \tag{7.20}$$

where J_A is block diagonal, with the typical block of J_A looking like

$$J = \begin{pmatrix} \lambda & 1 & 0 & \dots & 0 \\ 0 & \lambda & 1 & \dots & 0 \\ 0 & 0 & \lambda & \ddots & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & \ddots & \lambda \end{pmatrix} ,$$

for λ (complex) an eigenvalue of A. Each such component block matrix J has some size $r \geq 1$ and it has diagonal entries the corresponding eigenvalue and each entry just above the main diagonal equal to 1. Then, for every $k \geq 1$ we have

$$A^k = (T^{-1} J_A T)^k = T^{-1} J_A^k T \tag{7.21}$$

and J_A^k is the block diagonal matrix with blocks.

$$J^k = \begin{pmatrix} \lambda^k & k\lambda^{k-1} & \binom{k}{2}\lambda^{k-2} & \dots & \binom{k}{r-1}\lambda^{k-r+1} \\ 0 & \lambda^k & k\lambda^{k-1} & \ddots & \binom{k}{r-2}\lambda^{k-r+2} \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & ... & \lambda^k \end{pmatrix} .$$

This fact is called the Jordan decomposition.

The system is stable iff $|\lambda| < 1$ for all eigenvalues λ of A. The Jordan decomposition gives us a conceptual picture of the intrinsic dynamics of the system (i.e. the deterministic system with the controls are set equal to zero). However, if we want to influence the system through the control, we have to work through B.

7.3 Definition of Controllability

For the state evolution equation over a large enough horizon (bigger than the dimension of the state space), the deterministic dynamical system

$$x_{k+1} = A_k x_k + B_k u_k \tag{7.22}$$

is called controllable if for any x_0 we can move the state to any x_N via appropriate control.

Recall that the eigenvalues of A are the roots of the degree n polynomial in s

$$det(sI - A), \tag{7.23}$$

which is called the characteristic polynomial of A. For example, for

$$A = \begin{pmatrix} 2 & 1 \\ -1 & 0 \end{pmatrix}$$

the characteristic polynomial of A is,

$$det(sI - A) = det \begin{pmatrix} s-2 & 1 \\ -1 & s-0 \end{pmatrix} = (s-1)^2 \ .$$

The Cayley-Hamilton theorem tells us that A satisfies its own characteristic polynomial. That is, plugging A into its characteristic polynomial gives the zero matrix. In the example this means that $(A - I)^2 = 0$.

In general, the Cayley-Hamilton theorem implies that A^n is a linear combination of $I, A, A^2, A^3, ..., A^{n-1}$ (where A is an $n \times n$ matrix). For the time-invariant linear system,

$$x_N = A^N x_0 + \sum_{i=0}^{N-1} A^{N-i-1} B u_i \tag{7.24}$$

we have

$$x_N - A^N x_0 = (B \ AB \ \ldots \ A^{N-1}B) \begin{pmatrix} u_{N-1} \\ ... \\ u_0 \end{pmatrix} \tag{7.25}$$

On the right-hand side, if $N \geq n$, the range of this matrix is all of R^n precisely if the range space of $(B \ AB \ \ldots \ A^{n-1}B)$ is all of R^n, which is the necessary and sufficient condition for controllability.

Now consider only the observations in the time-invariant deterministic linear system model (with no control term)

$$x_{k+1} = Ax_k \tag{7.26}$$

$$y_k = Cx_k \ . \tag{7.27}$$

Therefore,

$$y_k = CA^k x_0 \tag{7.28}$$

Given the horizon of the observation is N, can we infer the state sequence from the observations? If so we call the system observable. We see that

$$y_0 = Cx_0, y_1 = CAx_0, ..., y_N = CA^{N-1}x_0 \ . \tag{7.29}$$

To infer the state sequence from $y_0, y_1, ..., y_N$ is equivalent to inferring x_0. So we want

$$\begin{pmatrix} C \\ CA \\ ... \\ CA^{N-1} \end{pmatrix}$$

to have null space $=\{0\}$. If $N \geq n$, then by the Cayley-Hamilton theorem, we equivalently want

$$\begin{pmatrix} C \\ CA \\ ... \\ CA^{n-1} \end{pmatrix}$$

to have null space $=\{0\}$. This is the necessary and sufficient condition for observability.

In the next lecture, we will discuss the finite horizon, fully observed, linear quadratic stochastic control problem. Informally speaking, we want to choose control strategy to minimize

$$E[\sum_{k=0}^{N-1}(X_k^T Q_k X_k + U_k^T R_k U_k) + X_N^T Q_N X_N] \tag{7.30}$$

where Q_k is positive semi-definite symmetric matrix, and R_k is positive definite symmetric matrix. Here the state evolves according to

$$x_{k+1} = A_k x_k + B_k u_k + w_k, \tag{7.31}$$

for $k = 0, 1, ..., N-1$, starting from some initial condition x_0.