Introduction

In the previous lecture, we have seen how to represent a multistage decision process and also the concept of suboptimization. In order to solve this problem in sequence, we make use of recursive equations. These equations are fundamental to the dynamic programming. In this lecture, we will learn how to formulate recursive equations for a multistage decision process in a backward manner and also in a forward manner.

Recursive equations

Recursive equations are used to structure a multistage decision problem as a sequential process. Each recursive equation represents a stage at which a decision is required. In this, a series of equations are successively solved, each equation depending on the output values of the previous equations. Thus, through recursion, a multistage problem is solved by breaking it into a number of single stage problems. A multistage problem can be approached in a backward manner or in a forward manner.

Backward recursion

In this, the problem is solved by writing equations first for the final stage and then proceeding backwards to the first stages. Consider the serial multistage problem discussed in the previous lecture.

\[ \begin{align*}
  \text{Stage 1} & : S_1 \rightarrow S_2 \rightarrow \ldots \rightarrow S_t \rightarrow S_{t+1} \\
  \text{Stage t} & : S_t \rightarrow S_{t+1} \\
  \text{Stage T} & : S_T \rightarrow S_{T+1}
\end{align*} \]
Suppose the objective function for this problem is

\[ f = \sum_{t=1}^{T} NB_t = \sum_{t=1}^{T} h_t(X_t, S_t) \]

\[ = h_1(X_1, S_1) + h_2(X_2, S_2) + \ldots + h_t(X_t, S_t) + \ldots + h_{T-1}(X_{T-1}, S_{T-1}) + h_T(X_T, S_T) \]  \hspace{1cm} \ldots(1)

and the relation between the stage variables and decision variables are given as

\[ S_{t+1} = g(X_t, S_t), \quad t = 1, 2, \ldots, T. \]  \hspace{1cm} \ldots(2)

Consider the final stage as the first subproblem. The input variable to this stage is \( S_T \).

According to the principle of optimality, no matter what happens in other stages, the decision variable \( X_T \) should be selected such that \( h_T(X_T, S_T) \) is optimum for the input \( S_T \). Let the optimum value be denoted as \( f_T^* \). Then,

\[ f_T^*(S_T) = \text{opt} \left[ h_T(X_T, S_T) \right] \]  \hspace{1cm} \ldots(3)

Next, group the last two stages together as the second subproblem. Let \( f_{T-1}^* \) be the optimum objective value of this subproblem. Then, we have

\[ f_{T-1}^*(S_{T-1}) = \text{opt} \left[ h_{T-1}(X_{T-1}, S_{T-1}) + h_T(X_T, S_T) \right] \]  \hspace{1cm} \ldots(4)

From the principle of optimality, the value of \( X_T \) should be to optimize \( h_T \) for a given \( S_T \).

For obtaining \( S_T \), we need \( S_{T-1} \) and \( X_{T-1} \). Thus, \( f_{T-1}^*(S_{T-1}) \) can be written as,

\[ f_{T-1}^*(S_{T-1}) = \text{opt} \left[ h_{T-1}(X_{T-1}, S_{T-1}) + f_T^*(S_T) \right] \]  \hspace{1cm} \ldots(5)

By using the stage transformation equation, \( f_{T-1}^*(S_{T-1}) \) can be rewritten as,

\[ f_{T-1}^*(S_{T-1}) = \text{opt} \left[ h_{T-1}(X_{T-1}, S_{T-1}) + f_T^*(g_{T-1}(X_{T-1}, S_{T-1})) \right] \]  \hspace{1cm} \ldots(6)
Thus, here the optimum is determined by choosing the decision variable $X_{T-1}$ for a given input $S_{T-1}$. Eqn (4) which is a multivariate problem (second sub problem) is divided into two single variable problems as shown in eqns (3) and (6). In general, the $i+1^{th}$ subproblem ($T-i^{th}$ stage) can be expressed as,

$$f^*_T(S_{T-i}) = \text{opt}_{X_{T-i},...,X_{T-i}} \left[ h_{T-i}(X_{T-i},S_{T-i}) + \ldots + h_{T-1}(X_{T-1},S_{T-1}) + h_T(X_T,S_T) \right] \ldots(7)$$

Converting this to a single variable problem,

$$f^*_T(S_{T-i}) = \text{opt}_{X_{T-i}} \left[ h_{T-i}(X_{T-i},S_{T-i}) + f^*_{T-(i-1)}(g_{T-i}(X_{T-i},S_{T-i})) \right] \ldots(8)$$

where $f^*_{T-(i-1)}$ denotes the optimal value of the objective function for the last $i$ stages. Thus for backward recursion, the principle of optimality can be stated as, no matter in what state of stage one may be, in order for a policy to be optimal, one must proceed from that state and stage in an optimal manner.

**Forward recursion**

In this approach, the problem is solved by starting from the stage 1 and proceeding towards the last stage. Consider the serial multistage problem with the objective function as given below

$$f = \sum_{t=1}^{T} NB_t = \sum_{t=1}^{T} h_t(X_t, S_t)$$

$$= h_1(X_1, S_1) + h_2(X_2, S_2) + \ldots + h_t(X_t, S_t) + \ldots + h_{T-1}(X_{T-1}, S_{T-1}) + h_T(X_T, S_T) \ldots(9)$$

and the relation between the stage variables and decision variables are gives as

$$S_t = g'(X_{t+1}, S_{t+1}) \quad t = 1, 2, \ldots, T \ldots(10)$$

where $S_t$ is the input available to the stages $1$ to $t$.

Consider stage 1 as the first subproblem. The input variable to this stage is $S_1$. The decision variable $X_1$ should be selected such that $h_1(X_1, S_1)$ is optimum for the input $S_1$. 

D Nagesh Kumar, IISc, Bangalore

M5L2
The optimum value $f^*_1$ can be written as

$$f^*_1(S_1) = \text{opt}[h_1(X_1, S_1)] \quad \text{...(11)}$$

Now, group the first and second stages together as the second subproblem. The objective function $f^*_2$ for this subproblem can be expressed as,

$$f^*_2(S_2) = \text{opt} \left[ h_2(X_2, S_2) + h_1(X_1, S_1) \right] \quad \text{...(12)}$$

But for calculating the value of $S_2$, we need $S_1$ and $X_1$. Thus,

$$f^*_2(S_2) = \text{opt} \left[ h_2(X_2, S_2) + f^*_1(S_1) \right] \quad \text{...(13)}$$

By using the stage transformation equation, $f^*_2(S_2)$ can be rewritten as,

$$f^*_2(S_2) = \text{opt} \left[ h_2(X_2, S_2) + f^*_1(g^*_2(X_2, S_2)) \right] \quad \text{...(14)}$$

Thus, here through the principle of optimality the dimensionality of the problem is reduced from two to one. In general, the $i^{th}$ subproblem can be expressed as,

$$f^*_i(S_i) = \text{opt} \left[ h_i(X_i, S_i) + \ldots + h_2(X_2, S_2) + h_1(X_1, S_1) \right] \quad \text{...(15)}$$

Converting this to a single variable problem,

$$f^*_i(S_i) = \text{opt} \left[ h_i(X_i, S_i) + f^*_{i-1}(g^*_i(X_i, S_i)) \right] \quad \text{...(16)}$$

where $f^*_i$ denotes the optimal value of the objective function for the first $i$ stages. The principle of optimality for forward recursion is that no matter in what state of stage one may be, in order for a policy to be optimal, one had to get to that state and stage in an optimal manner.