EE 223: Stochastic Estimation and Control **Spring 2007**

Lecture 6 - February 1

Lecturer: Venkat Anantharam *Scribe: Xianzhi Wang*

In the *Inventory Control* problem,

$$x_{k+1} = x_k + u_k - w_k,\ k = 0, 1, \cdots, N-1,\ u_k \geq 0,$$

where x_k is the inventory at time k, u_k is the amount restocked and w_k is the random demand. We want to minimize

$$\boldsymbol{E}[\sum_{k=0}^{N-1}(cu_k + r(x_k)) + r(x_N)].$$

Assume that $r(x)$ is piecewise linear with slope $h > 0$ when $x > 0$ and slope $-p < 0$ when $x < 0$, and $r(0) = 0$. Here, negative inventory means borrowing. We also assume that $p > c$, otherwise an optimal strategy will be to set the restocking amount to zero at each time.

Since $\boldsymbol{E}[r(x_0)]$ cannot be influenced, we can subtract it and rewrite the cost modified as

$$\boldsymbol{E}[\sum_{k=0}^{N-1}(cu_k + r(x_k + u_k - w_k))].$$

Then we take $g_k(x_k, u_k, w_k)$ to be $cu_k + r(x_k + u_k - w_k)$ and $g_N(x_N) = 0$ to fit this problem into our general set-up.

We can write the DP recursion as follows:

$$J_N(x_N) = 0;$$

$$J_k(x_k) = \min_{u_k \geq 0}\{cu_k + \boldsymbol{E}[r(x_k + u_k - w_k) + J_{k+1}(x_k + u_k - w_k)]\},\ k = N-1, \cdots, 0.$$

Write $\boldsymbol{E}[r(x_k + u_k - w_k)]$ as $H_k(x_k + u_k)$, where $H_k(y) = \boldsymbol{E}[r(y - w_k)]$.

Observe that since $r(x)$ is a convex function, so it $H_k(x)$. Further the slope of $H_k(x)$ approaches h as $x \to \infty$ and approaches $-p$ as $x \to -\infty$.

Recall: A function $f(x)$ is called *convex* if for $x_0,\ x_1$ and $\lambda \in [0, 1]$,

$$f(\lambda x_1 + (1-\lambda)x_0) \leq \lambda f(x_1) + (1-\lambda)f(x_0),$$

and *strictly convex* if the inequality is strict for all $x_0 \neq x_1$ and $\lambda \neq 0, 1$.
Fact: (easy) If f_1 and f_0 are convex function and $\lambda \in [0, 1]$, then $\lambda f_1 + (1-\lambda)f_0$ is convex.

$J_k(x_k)$ can be written as

$$J_k(x_k) = \min_{u_k \geq 0} \{cu_k + H_k(x_k + u_k) + \boldsymbol{E}[J_{k+1}(x_k + u_k - w_k)]\}.$$

Let y_k denote $x_k + u_k$. Then the constraint for the minimization $u_k \geq 0$ becomes $y_k \geq x_k$, and we may write cu_k as $cy_k - cx_k$. Define

$$G_k(y) = cy + H_k(y) + \boldsymbol{E}[J_{k+1}(y - w_k)].$$

Then,

$$J_k(x_k) = \min_{y_k \geq x_k} [G_k(y_k) - cx_k] = \min_{y_k \geq x_k} [G_k(y_k)] - cx_k.$$

Suppose we could show $G_k(y_k)$ is convex with strictly positive slope as $y_k \to \infty$ and strictly negative slope as $y_k \to -\infty$. Let's define $s_k = \text{argmin}_{y_k}[G_k(y_k)]$, where s_k can be any minimizer. Then $J_k(x_k) = G_k(y_k^*) - cx_k$, where

$$y_k^* = \begin{cases} s_k, & \text{if } s_k \geq x_k \\ x_k, & \text{else} \end{cases}$$

We will show that $J_k(x_k)$ is convex with positive slope as $x_k \to \infty$, and negative slope approaching $-c$ as $x_k \to -\infty$ for $k = 0, 1, \cdots, N-1$.

First of all, $G_{N-1}(y) = cy + H_{N-1}(y)$ is convex because $H_{N-1}(y)$ is convex, and its slope approaches $h + c > 0$ as $y \to \infty$, and approaches $-p + c < 0$ as $y \to -\infty$. So, $J_{N-1}(x_{N-1}) = c(s_{N-1} - x_{N-1})^+ + H_{N-1}(\max(s_{N-1}, x_{N-1}))$ where z^+ denotes $\max\{z, 0\}$ for a real number z. This function is convex with slope > 0 as $x_{N-1} \to \infty$ and slope approaching $-c < 0$ as $x_{N-1} \to -\infty$. Suppose that we have shown this for $J_{k+1}(x_{k+1})$ (i.e., convexity with these slope properties), then the same is true for $y \mapsto \boldsymbol{E}[J_{k+1}(y - w_k)]$, then $y \mapsto G_k(y)$ has the property that it is convex with strictly positive slope as $y \to \infty$ and strictly negative slope as $y \to -\infty$, hence $x_k \mapsto J_k(x_k)$ has the same convexity and slope properties as $x_{k+1} \mapsto J_{k+1}(x_{k+1})$, so the induction propagates backwards. QED.

Why is this interesting?
Reason 1: This methodology (finding some properties of the cost-to-go function that propagate under backwards induction and using these properties to conclude properties about the optimal Markov strategies) underlies a large number of papers in the technical literature and often provides nice engineering insights.
Reason 2: We learn in this specific example that restocking should be done according to a time varying threshold policy (i.e. restock to s_k iff $x_k \leq s_k$), which is quite nice and appealing. This can be seen from the minimizers in the DP recursion above.

Looking ahead we will be discussing some of the literature on estimation and identification for both Markov chains and linear systems.

Soon we will discuss partially observed stochastic control problems. In this kind of situation the controller needs to run an estimator for the states. Another issue is that the system model may be uncertain (e.g. parameters within the model may be unknown). "Identification" of the system may need to be done (estimating the underlying parameters). Identification of system parameters is often combined with control strategies based on the estimated system parameters (adaptive control). Then control plays a *dual role*: it is being used both to elicit information about the underlying model parameters and to achieve the performance objective.

The DP approach has a very big role to play in practice for problems of estimation and identification.

Example [Estimating the States of a Hidden Markov Model based on observations]

(Applications in communications/speech processing/artificial intelligence/expert systems, etc.)

Suppose a system is modeled as evolving as a Markov chain, with $x_0, x_1, \cdots, x_N$ the state sequence, but this is hidden. We get to observe functions of the states $y_1, y_2, \cdots, y_N$. Let's assume the joint distribution of the state of observation is

$$p(x_0) \prod_{k=0}^{N-1} p(x_{k+1}, y_{k+1} | x_k).$$

We'd like to estimate the underlying state sequence given the observations (we seek the maximum a posteriori probability estimate). We want to find

$$(\hat{x}_0, \cdots, \hat{x}_N) = \text{argmax}_{x_0, \cdots, x_N} P(x_0, \cdots, x_N \mid y_1, \cdots, y_N)$$

i.e.,

$$\min \{-\text{log} P(x_0) - \sum_{k=0}^{N} \text{log} P(x_{k+1}, y_{k+1} | y_k)\}$$

over all state sequences $x_0, \ldots, x_N$. This is a minimum weight path problem on the trellis associated to the transition matrix of the underlying Markov chain with weight $-\text{log} P(x_{k+1}, y_{k+1} | x_k)$ given to the edge from x_k to x_{k+1} (the weight depends on the actual value of the corresponding observation y_{k+1}). We already saw that finding the minimum weight path on a trellis can be done using a DP based algorithm. In communications applications where we are estimating an input sequence to a channel (or a code) described by a finite state machine, based on noisy observations at the output, the corresponding DP based algorithm is called the *Viterbi* algorithm.