Further Considerations on Dynamic Programming

V. Leclère (ENPC)

October 16, 2014

Contents

Contents

Stochastic Controlled Dynamic System

A stochastic controlled dynamic system is defined by its *dynamic*

$$\mathbf{X}_{t+1} = f_t(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1})$$

and initial state

$$\mathbf{X}_0 = x_0$$

The variables

- $\mathbf{X}_t$ is the *state* of the system,
- $\mathbf{U}_t$ is the *control* applied to the system at time t,
- $\mathbf{W}_t$ is an exogeneous noise.

Optimization Problem

We want to solve the following optimization problem

$$
\begin{aligned}
\min \quad & \mathbb{E}\Big[\sum_{t=0}^{T-1} L_t\big(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1}\big) + K\big(\mathbf{X}_T\big)\Big] && (1) \\
s.t. \quad & \mathbf{X}_{t+1} = f_t(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1}), \qquad \mathbf{X}_0 = x_0 && (2) \\
& \mathbf{U}_t \in U_t(\mathbf{X}_t) && (3) \\
& \mathbf{U}_t \preceq \sigma\big(\mathbf{W}_0, \cdots, \mathbf{W}_t\big) && (4)
\end{aligned}
$$

Dynamic Programming Principle

Assume that the noises $\mathbf{W}_t$ are independent.

Then, there exists an optimal solution of the form $\mathbf{U}_t = \pi_t(\mathbf{X}_t)$, given by

$$\pi_t(x) = \arg\min_{u \in U_t(x)} \mathbb{E}\Big[\underbrace{L_t(x, u, \mathbf{W}_{t+1})}_{\text{current cost}} + \underbrace{V_{t+1} \circ f_t\big(x, u, \mathbf{W}_{t+1}\big)}_{\text{future costs}} \Big] ,$$

where

$$\begin{cases} V_T(x) & = K(x) \\ V_t(x) & = \min_{u \in U_t(x)} \mathbb{E}\Big[L_t(x, u, \mathbf{W}_{t+1}) + V_{t+1} \circ f_t\big(x, u, \mathbf{W}_{t+1}\big) \Big] \end{cases}$$

Interpretation of Bellman Value

The Bellman's value function $V_{t_0}(x)$ can be interpreted as the value of the problem starting at time t_0 from the state x. More precisely we have

$$
\begin{aligned}
V_{t_0}(x) = \min \quad & \mathbb{E}\Big[\sum_{t=t_0}^{T-1} L_t(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1}) + K(\mathbf{X}_T)\Big] && (5)\\
s.t. \quad & \mathbf{X}_{t+1} = f_t(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1}), \qquad \mathbf{X}_{t_0} = x && (6)\\
& \mathbf{U}_t \in U_t(\mathbf{X}_t) && (7)\\
& \mathbf{U}_t \preceq \sigma\big(\mathbf{W}_0, \cdots, \mathbf{W}_t\big) && (8)
\end{aligned}
$$

Contents

Dynamic Programming Algorithm

Data: Problem parameters
Result: optimal control and value;
$V_T \equiv K$;
for $t : T \to 0$ **do**
  for $x \in \mathbb{X}_t$ **do**
    $\underline{\mathrm{v}} = -\infty$;
    for $u \in U_t(x)$ **do**
      $\underline{\mathrm{v}} = \min \Big\{ \underline{\mathrm{v}}, \mathbb{E}\Big[L_t(x, u, \mathbf{W}_{t+1}) + V_{t+1} \circ f_t\big(x, u, \mathbf{W}_{t+1}\big) \Big] \Big\}$;
    $V_t(x) = \underline{\mathrm{v}}$;

Algorithm 1: Dynamic Programming Algorithm (discrete case)

Number of flops: $O(T \times |\mathbb{X}_t| \times |\mathbb{U}_t| \times |\mathbb{W}_t|)$.

3 curses of dimensionality

1. State. If we consider 3 independent states each taking 10 values, then $|\mathbb{X}_t| = 1000$. In practice DP is not applicable for states of dimension more than 5.
2. Decision. The decision are often vector decisions, that is a number of independent decision, hence leading to huge $|U_t(x)|$.
3. Expectation. In practice random information came from large data set. Without a proper statistical treatment computing an expectation is costly. Monte-Carlo approach are costly too, and unprecise.

Contents

Introducing the Bellman operators

We define the Bellman operator associated to our optimisation problem

$$T_t(J) : x \mapsto \min_{u \in U_t(x)} \mathbb{E}\Big[L_t(x, u, \mathbf{W}_{t+1}) + J \circ f_t\big(x, u, \mathbf{W}_{t+1}\big)\Big] .$$

The Dynamic Programming equation can then be written

$$\begin{cases} V_T &= K \\ V_t &= T_t\Big(V_{t+1}\Big) \end{cases}$$

We also construct the *policy-dependent Bellman operator*

$$T_t^{\pi}(J) : x \mapsto \mathbb{E}\Big[L_t(x, \pi(x), \mathbf{W}_{t+1}) + J \circ f_t\big(x, \pi(x), \mathbf{W}_{t+1}\big)\Big] .$$

Discounted fixed cost case

We now consider the following specific case problem, where $\big(\mathbf{W}_t\big)_{t\in\mathbb{N}}$ is i.i.d.

$$\begin{aligned}
\min \quad & \mathbb{E}\Big[\sum_{t=0}^{T} \alpha^t L\big(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1}\big)\Big] && (9)\\
s.t. \quad & \mathbf{X}_{t+1} = f(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1}), \qquad \mathbf{X}_0 = x_0 && (10)\\
& \mathbf{U}_t \in U(\mathbf{X}_t) && (11)\\
& \mathbf{U}_t \preceq \sigma\big(\mathbf{W}_0, \cdots, \mathbf{W}_t\big) && (12)
\end{aligned}$$

where $\alpha \in]0,1]$. Note that the constraint and cost structure doesnot depend on t.

The Bellman operator is given by

$$T(J) : x \mapsto \min_{u\in U(x)} \mathbb{E}\Big[L(x,u,\mathbf{W}_{t+1}) + \alpha J \circ f(x,u,\mathbf{W}_{t+1})\Big]$$

Infinite horizon problems

There is different ways of considering the above problem in an "infinite horizon" setting.

1. Discounted case. This is the case where $\alpha < 1$. It is especially easy to treat if the cost L is bounded.
2. Stocastic shortest path. In this case $\alpha = 1$ but there is a "cemetary state" such that once reached the system remains there with null cost. Moreover, we assume that the system always reach the cemetary state in a finite time.
3. Average cost per stage problems. This approach is mainly taken if the infinite time cost isn't finite (for example $\alpha = 1$ and $L > 0$). We consider

$$\lim_{T \to \infty} \frac{1}{T} \quad \mathbb{E}\Big[\sum_{t=0}^{T-1} L(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_{t+1}) \Big] .$$

An overview of typical infinite horizon results

Here are the main results that can be shown in infinite horizon problems (under the right set of assumptions)

1. the sequence of value function $V_{n+1} = T\big(V_n\big)$, converges toward the value function of the infinite horizon problem: $\lim_{n\to\infty} V_n = V^\sharp$.
2. The optimal value of the infinite horizon problem is a fixed point of the Bellman operator: $V^\sharp = T\big(V^\sharp\big)$.
3. If π is such that $V^\sharp = T^\pi V^\sharp$ then the stationnary policy π is optimal.

Value iteration algorithm

Data: Initial value $V^{(0)}$
Result: optimal policy and value;
repeat
  for $x \in \mathbb{X}$ **do**

$$V^{(k+1)}(x) = T\big(V^{(k)}\big)(x)$$

until $\|V^{(k+1)} - V^{(k)}\|_\infty < \varepsilon$;

Algorithm 2: Value iteration algorithm

- Each step takes $O(|\mathbb{X}| \times |\mathbb{U}| \times |\Omega|)$ flops.
- The error $|V_n(x) - V^\sharp(x)|$ is bounded by $C\alpha^n$.

Policy iteration algorithm

Data: Initial policy $\pi^{(0)}$
Result: optimal policy and value;
repeat
  policy evaluation step: solve $V = T^{\pi^{(k)}}(V)$ which gives $V^{(k)}$;
  policy improvement step :
  for $x \in \mathbb{X}$ **do**

$$\pi^{(k+1)}(x) = \arg\min_{u \in U(x)} \mathbb{E}\Big[L(x,u,\mathbf{W}_{t+1}) + \alpha V^{(k)} \circ f\big(x,u,\mathbf{W}_{t+1}\big)\Big]$$

until $V^{(k)} = T\big(V^{(k)}\big)$;

Algorithm 3: Policy iteration algorithm

The policy iteration algorithm terminate in a finite number of step.