

Dynamic Programming

Guillaume Rabault

In this short post, I sketch the derivation of the dynamic programming principle in discrete time.

Optimization problem

- Intertemporal program:

$$\begin{aligned} V_t(x_t, W_t) &= \max_{(\boldsymbol{\pi}_{[t,T]}, \mathbf{C}_{[t,T]})} E_t \left[\sum_{k=t}^T \delta^k u_k(C_k) \right] \\ &\text{s.t.} \\ W_{k+1} &= Y_{k+1} + (W_k - C_k) \pi'_k R_{k+1}, \\ C_k &\geq 0, \\ W_T &\geq 0. \end{aligned}$$

Additive decomposition

- We'll define:

$$J(t, x_t, W_t, \boldsymbol{\alpha}_{[t,T]}) = E_t \left[\sum_{k=t}^T \delta^k u_k(C_k) \right],$$

where I have packed all decision variables over $[t, T]$ into the notation $\boldsymbol{\alpha}_{[t,T]}$:
 $\boldsymbol{\alpha}_{[t,T]} = (C_{[t,T]}, \boldsymbol{\pi}_{[t,T]})$ and similarly $\boldsymbol{\alpha}_t = (C_t, \boldsymbol{\pi}_t)$.

- We have:

$$J(t, x_t, W_t, \boldsymbol{\alpha}_{[t,T]}) = E_t \left[\sum_{k=t}^T \delta^k u_k(C_k) \right],$$

$$J(t, x_t, W_t, \boldsymbol{\alpha}_{[t,T]}) = \delta^t u_t(C_t) + E_t [J(t+1, x_{t+1}, W_{t+1}, \boldsymbol{\alpha}_{[t+1,T]})].$$

- The latter equation is the starting point of everything.

You cannot beat backward induction

- We have:

$$\begin{aligned}
 J(t, x_t, W_t, \boldsymbol{\alpha}_{[t,T]}) &\leq \delta^t u_t(C_t) + E_t[V_{t+1}(x_{t+1}, W_{t+1})], \\
 J(t, x_t, W_t, \boldsymbol{\alpha}_{[t,T]}) &\leq \max_{\boldsymbol{\alpha}_t} \{\delta^t u_t(C_t) + E_t[V_{t+1}(x_{t+1}, W_{t+1})]\}, \\
 V_t(x_t, W_t) &\leq \max_{\boldsymbol{\alpha}_t} \{\delta^t u_t(C_t) + E_t[V_{t+1}(x_{t+1}, W_{t+1})]\}.
 \end{aligned}$$

- This can be paraphrased as saying that carrying out the program $\max_{\boldsymbol{\alpha}_t} \{\delta^t u_t(c_t) + E_t[V_{t+1}(x_{t+1}, W_{t+1}, w_{t+1})]\}$, if feasible, is necessarily optimal.

You can do it (1)

- It remains to be shown that the value of the above program can be arbitrarily approached through an adequate choice of commands. For any $\varepsilon > 0$, for each (x_{t+1}, W_{t+1}) we can find a command $\boldsymbol{\alpha}^\varepsilon_{[t+1,T]}$ such that:

$$J(t+1, x_{t+1}, W_{t+1}, \boldsymbol{\alpha}^\varepsilon_{[t+1,T]}) \geq V_{t+1}(x_{t+1}, W_{t+1}) - \varepsilon.$$

The dependence of the command on the state variable is omitted.

- But then we can build a command¹ $\hat{\boldsymbol{\alpha}}^\varepsilon_{[t,T]} = (\boldsymbol{\alpha}_t, \boldsymbol{\alpha}^\varepsilon_{[t+1,T]})$ for which:

$$\begin{aligned}
 J(t, x_t, W_t, \hat{\boldsymbol{\alpha}}^\varepsilon_{[t,T]}) &= \\
 &\delta^t u_t(C_t) + E_t[J(t+1, x_{t+1}, W_{t+1}, \boldsymbol{\alpha}^\varepsilon_{[t+1,T]})]. \\
 J(t, x_t, W_t, \hat{\boldsymbol{\alpha}}^\varepsilon_{[t,T]}) &\geq \delta^t u_t(C_t) + E_t[V_{t+1}(x_{t+1}, W_{t+1})] - \varepsilon.
 \end{aligned}$$

You can do it (2)

This being true for any $\boldsymbol{\alpha}_t = (C_t, \boldsymbol{\pi}_t)$, we have:

$$\begin{aligned}
 J(t, x_t, W_t, \hat{\boldsymbol{\alpha}}^\varepsilon_{[t,T]}) &\geq \max_{\boldsymbol{\alpha}_t} \{\delta^t u_t(c_t) + E_t[V_{t+1}(x_{t+1}, W_{t+1})]\} - \varepsilon, \\
 V_t(x_t, W_t) &\geq \max_{\boldsymbol{\alpha}_t} \{\delta^t u_t(c_t) + E_t[V_{t+1}(x_{t+1}, W_{t+1})]\} - \varepsilon.
 \end{aligned}$$

and since $\varepsilon > 0$ is arbitrary:

$$V_t(x_t, W_t) \geq \max_{\boldsymbol{\alpha}_t} \{\delta^t u_t(c_t) + E_t[V_{t+1}(x_{t+1}, W_{t+1})]\}.$$

- This complete the proof of the dynamic programming principle.

¹Important technical details are omitted here. Indeed, one has to select an ε -optimal command for each value of the state variable, uniformly with respect to the state variable. One cannot hope to do that without strong assumptions regarding the data of the optimization problem.

Links

- [Link to pdf](#)