Lecture 12 — Dynamic programming

- Closed loop formulation of optimal control
- Intuitive methods of solution
- Guarantees global optimality
- Iteratively solves the problem starting at the end-time

"Life can only be understood backwards; but it must be lived forwards"

Kierkegaard

Example: Shortest path

As an example we try to find the shortest path to "0" in the above graph.

Example: Shortest path

Goal

To be able to
- to understand the idea of Dynamic programming
- to derive optimal feedback laws in simple cases
Basic problem formulation: The system

- First we assume that the system is in discrete time
  \[ x_{k+1} = f_k(x_k, u_k), \quad k = 0, 1, \ldots, N - 1 \]
  where \( x_k \) is the state \( u_k \in U(x_k) \) is the control.
- Feedback-control implies \( u_k = \mu_k(x_k) \)
- In closed-loop form the system can thus be written
  \[ x_{k+1} = f_k(x_k, \mu_k(x_k)) \]

Example: Shortest path

![Example: Shortest path](image)

Basic problem formulation: The cost

- We let \( \mu = \{\mu_0, \mu_1, \ldots, \mu_{N-1}\} \) and assume that we have an additive cost
  \[ J_\mu(x_0) = g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k)) \]
- Total cost \( J_\mu(x_0) \) is a function of both initial state \( x_0 \) and feedback law \( \mu \)
- \( N \) is the horizon of the problem
  - Finite-horizon: \( N < \infty \)
  - Infinite-horizon: \( N = \infty \)

The principle of optimality

Let \( \mu^* = \{\mu_0^*, \mu_1^*, \ldots, \mu_{N-1}^*\} \) be an optimal policy for the basic problem and assume that when applying \( \mu^* \), a given state \( x_i \) occurs at time \( i \), when starting at \( x_0 \). Consider the subproblem whereby we are in state \( x_i \) at time \( i \) and wish to minimize the “cost-to-go” from time \( i \) to time \( N \)

\[ g_N(x_N) + \sum_{k=i}^{N-1} g_k(x_k, \mu_k(x_k)) \]

Principle of optimality

The truncated policy \( \{\mu_i^*, \mu^*_{i+1}, \ldots, \mu^*_{N-1}\} \) is optimal for the subproblem starting from \( x_i \) at time \( i \).

The dynamic programming algorithm

Let

\[ V_k(x_k) = g_N(x_N) + \sum_{j=k}^{N-1} g_j(x_j, \mu_j^*(x_j)) \]

so that \( V_k(x_k) \) is the optimal “cost-to-go” from time \( k \) to time \( N \)

The Bellman equation

For every initial state \( x_0 \), the optimal cost \( J^*(x_0) \) is given by the last step in the following backward-recursion.

\[ V_0(x_0) = g(x_0, u_0) \]

We get the optimal control “for-free”

\[ \mu^*_0(x_0) = \arg \min_{u_0 \in U(x_0)} [g(x_0, u_0) + V_1(f_0(x_0, u_0))] \]
Managing spending and saving

Example
An investor holds a capital sum in a building society, which gives an interest rate of $\theta \times 100\%$ on the sum held at each time $k = 0, 1, \ldots, N - 1$. The investor can choose to reinvest a portion of the interest paid which then itself attracts interest. No amounts invested can ever be withdrawn. How should the investor act so as to maximize total reward by time $N - 1$?

- We take as the state $x_k$ the present income at time $k = 0, 1, \ldots, N - 1$ and let $u_k \in [0, 1]$ be the fraction of reinvested interest, hence
  
  $\dot{x}_{k+1}= x_k + \theta u_k x_k = f(x_k, u_k)$

- The reward is $g_k(x, u) = (1 - u)x$ and $g_N(x, u) \equiv 0$.

Managing spending and saving

- We have thus verified that $V(N - s, x) = \rho_s x$, and found the recursion
  
  $\rho_s = \rho_{s-1} + \max \{1, \theta \rho_{s-1}\}$

- Together with $\rho_1 = 1$ this gives
  
  $\rho_s = \left\{ \begin{array}{ll} s & \text{for } s \leq s^* \\ s^{*} (1 + \theta)^{s-s^*} & \text{otherwise.} \end{array} \right. \quad s^* = \lceil 1/\theta \rceil$

- The optimal policy is then
  
  $u_k = \left\{ \begin{array}{ll} 1 & \text{for } k < N - s^* \\ 0 & \text{for } k \geq N - s^*. \end{array} \right.$

Continuous time problem formulation

- In continuous time the system is given by
  
  $\dot{x}(t) = f(x(t), u(t)), \quad t \in [0, T]$ with $x(0) = x_0$ and $u(t) \in U(x(t))$, for all $t \in [0, T]$.

- We define the cost as
  
  $J(x_0) = \phi(x(T)) + \int_0^T L(x(t), u(t)) dt$

- With optimal "cost-to-go" from $(t, x)$
  
  $V(t, x) = \min_u \{ \phi(x(T)) + \int_t^T L(x(t), u(t)) dt \}$

The HJB-equation: Informal derivation

- Dynamic programming now yields
  
  $V(k\delta, x) = \min_u \{ L(x, u) \delta + V((k+1)\delta, x + f(x, u)\delta) \}$

  $V(N\delta, x) = \phi(x)$. 

- For small $\delta$ we get (with $t = k\delta$)
  
  $V(t + \delta, x + f(x, u)\delta) \approx V(t, x) + V(t, x)\delta + \nabla_x V(t, x) \cdot f(x, u)\delta$

- Inserting this in the DP equation gives
  
  $V(t, x) \approx \min_u \{ L(x, u) \delta + V(t, x) + V(t, x)\delta + \nabla_x V(t, x) \cdot f(x, u)\delta \}$

Continuous time optimal control: The HJB-equation

- The optimality equation is $V(N, x) = 0,$

  $V(k, x) = \max_{0 \leq u \leq 1} \{ (1-u)x + V(k+1, (1+\theta u)x) \}$, \quad $k = 0, 1, \ldots, N-1$

- We get
  
  $V(N - 1, x) = \max_{0 \leq u \leq 1} \{ (1-u)x + 0 \} = x$

  $V(N - 2, x) = \max_{0 \leq u \leq 1} \{ (1-u)x + (1 + \theta u)x \} = \max \{ 2x + (\theta - 1)ux \} = \max \{ 2, 1 + \theta \} x = \rho_2 x$

- Guess: $V(N - s + 1, x) = \rho_{s-1} x$, then
  
  $V(N - s, x) = \max_{0 \leq u \leq 1} \{ (1-u)x + \rho_{s-1}(1 + u\theta)x \} = \max \{ 1 + \rho_{s-1}, (1 + \theta)\rho_{s-1} \} x = \rho_s x$

The HJB-equation: Informal derivation

- divide $[0, T]$ into $N$ subintervals of length $\delta = T/N$

- Let $x_k = x(k\delta)$ and $u_k = u(k\delta)$, for $k = 0, 1, \ldots, N$ and approximate the system by
  
  $x_{k+1} = x_k + f(x_k, u_k)\delta, \quad k = 0, 1, \ldots, N$.

- The optimal "cost-to-go" is approximated by
  
  $V(k\delta, x) = \min_{x_0, \ldots, x_{N-1}} \{ \phi(x_N) + \sum_{k=0}^{N-1} L(x_k, u_k)\delta \}$

The Hamilton-Jacobi-Bellman equation

For every initial state $x_0$, the optimal cost is given by

$J^*(x_0) = V(0, x_0)$ where $V(t, x)$ is the solution to the PDE

$V_t(t, x) = -\min_{u \in U} \{ L(x, u) + \nabla_x V(t, x) \cdot f(x, u) \}$

$V(T, x) = \phi(x)$

As before the optimal control is given in feedback form by

$\mu^*(t, x) = \arg\min_{u \in U} \{ L(x, u) + \nabla_x V(t, x) \cdot f(x, u) \}$
Example: The HJB-equation

Consider the simple example involving the scalar system
\[ x(t) = u(t), \]
with the constraint \(|u(t)| \leq 1\) for all \(t \in [0, T]\) and the cost
\[ J(x_0) = \frac{1}{2} (x(T))^2. \]

The HJB equation for this problem is
\[ V_t = \min_{u} \{ V_x(x) u \}, \]
with terminal condition \(V(T, x) = x^2/2\).

Example: The HJB-equation

\[ V_t(t, x) = -\min_{u} [V_x(t, x) u] \]
with terminal condition \(V(T, x) = x^2/2\).

Infinite horizon problem: example

Assume that the final cost is \(\phi(x(T)) = 0\) and the final time \(T \to +\infty\), and that there exists some control such that the total cost remains bounded in the limit. Hence, we want to solve
\[ \min_{u} \int_0^T L(x(t), u(t)) dt, \quad x(0) = x_0 \]
It is intuitive that the cost-to-go from \((x, t)\)
\[ V(x, t) = \min_{u} \int_t^T L(x(t), u(t)) dt = V(x) \]
does not depend on the initial time but only on the initial state. The HJB equation then becomes
\[ 0 = \min_u \{ L(x, u) + \nabla_x V(x) \cdot f(x, u) \} \]
(Observable that, for scalar problems, this is an ODE!)

Example: The HJB-equation

\[ \begin{cases} 
1 & \text{for } x < 0 
\end{cases} \]
\[ \begin{cases} 
0 & \text{for } x = 0 
\end{cases} \]
\[ \begin{cases} 
-1 & \text{for } x > 0 \]

The optimal "cost-to-go" with this control is
\[ V(t, x) = \frac{1}{2} (\max\{0, |x| - (T - t)\})^2 \]

Dynamics Programming for LQ control

Consider the optimal feedback control problem for an LTI system \(\dot{x} = Ax + Bu\) with cost
\[ J = \int_0^T \left( x'(t)Qx(t) + u'(t)Ru(t) \right) dt + x(T)'Mx(T) \]
where \(Q, R, M\) are symmetric positive definite. The HJB eqn reads
\[ 0 = \min_u \{ x'Qx + u'Ru + \nabla_x V(x) \cdot f(x, u) \} \]
with final time condition \(V(T, x) = x'Mx\).

Summary — Dynamic programming

- Closed loop formulation of optimal control
- Intuitive methods of solution
- Guarantees global optimality
- Iteratively solves the problem starting at the end-time