Dynamic Programming and Optimal Control HS18

Sean Bone
http://weblog.zumguy.com/

January 24, 2019

1 General problem

Dynamics

$$x_{k+1} = f_k(x_k, u_k, w_k), \quad k = 0, ..., N-1$$

- k: discrete time index, or stage;
- N: given time horizon;
- $x_k \in \mathcal{S}_k$: system state vector at time k;
- $u_k \in \mathcal{U}_k$: control input vector at time k;
- w_k: random disturbance vector at time k, conditionally independent with all prior $x_l, u_l, w_l, l < k$. The conditional probability distribution of w_k is known given x_k, u_k;
- $f_k(\cdot,\cdot,\cdot)$: function capturing system evolution at time k.

Cost function

$$\underbrace{g_N(x_N)}_{\text{terminal cost}} + \underbrace{\sum_{k=0}^{N-1} \underbrace{g_k(x_k, u_k, w_k)}_{\text{stage cost}}}_{\text{accumulated cost}}$$

1.1 Control strategies

Open-loop

Given an initial state x_0, find a *fixed* sequence of control inputs $U = (u_0, ..., u_{N-1})$ that minimizes the expected cost:

$$\mathop{\mathbb{E}}_{(X_1, W_0 | x_0)} \left[g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k) \right]$$

Closed-loop

Define

$$u_k = \mu_k(x_k), \quad u_k \in \mathcal{U}_k, k = 0, ..., N-1,$$
$$\pi := (\mu_0(\cdot), ..., \mu_{N-1}(\cdot)),$$

where π is called *admissible policy*. Given an initial state x_0, the expected cost is now:

$$J_\pi := \mathop{\mathbb{E}}_{(X_1, W_0 | x_0)} \left[g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k), w_k) \right]$$

Let Π be the set of all admissible policies. We seek the the *optimal policy* π^* with the *optimal cost*:

$$J^* := J_{\pi^*}(x) \leq J_\pi(x) \quad \forall \pi \in \Pi, \forall x \in \mathcal{S}_0.$$

1.2 Discrete states

If x_k takes on discrete values or is finite, we usually express the dynamics in terms of *transition probabilities*:

$$P_{ij}(u, k) := \Pr(x_{k+1} = j | x_k = i, u_k = u) = p_{x_{k+1}|x_k, u_k}(j | i, u),$$

where $p_{x_k+1|x_k,u_k}(\cdot|\cdot,\cdot)$ denotes the PDF of x_{k+1} given x_k and u_k. This is equivalent to the dynamics:

$$x_{k+1} = w_k$$

where w_k has the following probability distribution:

$$p_{w_k|x_k,u_k}(j|i,u) = P_{ij}(u,k)$$

Conversely, given $x_{k+1} = f_k(x_k, u_k, w_k)$ and $p_{w_k|x_k,u_k}(\cdot|\cdot,\cdot)$, then

$$P_{ij}(u,k) = \sum_{\{\bar{w}_k | f_k(i,u,\bar{w}_k) = j\}} p_{w_k|x_k,u_k}(\bar{w}_k | i, u),$$

that is, $P_{ij}(u,k)$ is equal to the sum over the probabilities of all possible disturbances $\bar{w}_k$ that get us to state j from state i with control input u at time k.

1.3 DPA

Principle of optimality

If $\pi^* = (\mu_0^*(\cdot), ..., \mu_{N-1}^*(\cdot))$ is an optimal policy, then the truncated policy $\pi = (\mu_i(\cdot), ..., \mu_{N-1}(\cdot))$ minimizes the cost from stage i to N.
This provides the intuition as to why the control inputs selected by the following algorithm constitute the optimal policy:

1. Initialization

$$J_N(x) = g_N(x), \quad \forall x \in \mathcal{S}_N$$

2. Recursion

$$J_k(x) := \min_{u \in \mathcal{U}_k(x)} \mathop{\mathbb{E}}_{(w_k | x_k = x, u_k = u)} \left[g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)) \right], \quad \forall y \in \mathcal{S}_k, k = N-1, ..., 0.$$

2 Infinite Horizon Problems

Consider the same problem as before, but with time-invariant state evolution and stage costs:

$$x_{k+1} = f(x_k, u_k, w_k), \forall x_k \in \mathcal{S}, u_k \in \mathcal{U},$$
$$\text{cost } = \sum_{k=0}^{N-1} g(x_k, u_k, w_k).$$

Assuming the cost converges as $N \to \infty$, problem simplifies to finding the optimal *stationary* policy by means of the *Bellman Equation* (BE):

$$J(x) = \min_{u \in \mathcal{U}(x)} \mathop{\mathbb{E}}_{(w | x = x, u = u)} \left[g(x, u, w) + J(f(x, u, w)) \right], \quad \forall x \in \mathcal{S}$$

2.1 SSP

The Stochastic Shortest Path problem is a class of problems for which solving the BE yields the optimal cost-to-go and stationary policy.

Dynamics

Given a finite set $\mathcal{S}$ and $\mathcal{U}(x)$ for all $x \in \mathcal{S}$, consider the finite state, time-invariant system:

$$x_{k+1} = w_k, \quad x_k \in \mathcal{S},$$
$$\Pr(w_k = j | x_k = i, u_k = u) = P_{ij}(u), \; u \in \mathcal{U}(i)$$

Terminal state

In order for the cost to be meaningful, there must be a *terminal state*, designated as 0, with:

$$P_{00}(u) = 1 \text{ and } g(0, u, 0) = 0, \quad \forall u \in \mathcal{U}(0).$$

We assume there is at least one proper policy, and that improper policies will have infinite cost for at least one starting state. For a policy to be *proper*, there must be at least one m for which

$$\Pr(x_m = 0 | x_0 = i) > 0, \quad \forall i \in \mathcal{S}.$$

2.2 Discounted problems

Consider an SSP with no explicit termination state and a cost discount factor $\alpha \in]0,1[$:

$$\tilde{J}_{\tilde{\pi}}(i) = \mathop{\mathbb{E}}_{(\tilde{X}_1, \tilde{W}_0 | \tilde{x}_0 = i)} \left[\sum_{k=0}^{N-1} \alpha^k \tilde{g}(\tilde{x}_k, \tilde{\mu}_k, \tilde{w}_k) \right]$$

We can convert this problem to an SSP by adding a virtual termination state.
State: $x_k \in \mathcal{S} = \tilde{\mathcal{S}} \cup \{0\}$
Control: $\mathcal{U}(x_k) = \tilde{\mathcal{U}}(x_k), \mathcal{U}(0) = \{\texttt{stay}\}$
Dynamics: $x_{k+1} = w_k$ where $\forall u, \forall i, j$:

$$p_{w|x,u}(j|i,u) = P_{ij}(u) = \alpha \tilde{P}_{ij}(u),$$
$$p_{w|x,u}(0|i,u) = P_{i0}(u) = 1 - \alpha,$$
$$p_{w|x,u}(j|0,\texttt{stay}) = P_{0j}(\texttt{stay}) = 0,$$
$$p_{w|x,u}(j|i,u) = P_{00}(u) = 1.$$

Cost: $\forall u_k, \forall x_k, w_k$

$$g(x_k, u_k, w_k) = \frac{1}{\alpha} \tilde{g}(x_k, u_k, w_k),$$
$$g(x_k, u_k, 0) = 0,$$
$$g(0, \texttt{stay}, 0) = 0.$$

From this we can derive the Bellman Equation for the original problem:

$$J^*(i) = \min_{u \in \tilde{\mathcal{U}}(i)} \left(q(i,u) + \alpha \sum_{j=1}^n \tilde{P}_{ij}(u) J^*(j) \right), \forall i \in \tilde{\mathcal{S}}$$

$$q(i,u) = \sum_{j=1}^n \tilde{P}_{ij}(u) \tilde{g}(i, \mu_k(i), j)$$

2.3 Solving the BE

Value Iteration (VI)

Given any initial conditions $V_0(\cdot)$, the following sequence converges to the optimal cost $J^*(\cdot)$ which uniquely solves the BE, and the corresponding u for each i constitute the optimal policy:

$$V_{l+1} = \min_{u \in \mathcal{U}(i)} \left[q(i,u) + \sum_{j=1}^n P_{ij}(u) V_l(j) \right] \quad \forall i \in \mathcal{S}^+ = \mathcal{S} \setminus \{0\}$$

$$q(i,u) := \mathop{\mathbb{E}}_{(w | x = i, u = u)} [g(i,u,w)]$$

Policy Iteration (PI)

0. Initialize with a proper policy.
1. Policy evaluation: $\forall i \in \mathcal{S}^+$,

$$J_{\mu^h}(i) = q(i, \mu^h(i)) + \sum_{j=1}^n P_{ij}(\mu^h(i)) J_{\mu^h}(j)$$

2. Policy improvement: $\forall i \in \mathcal{S}^+$,

$$\mu^{h+1}(i) = \arg\min_{u \in \mathcal{U}(i)} \left(q(i,u) + \sum_{j=1}^n P_{ij}(u) J_{\mu^h}(j) \right)$$

Repeat 1 and 2 until $J_{\mu^{h+1}}(i) = J_{\mu^h}(i) \; \forall i$

Linear Programming (LP)

The solution to the following optimization program solves the BE:

$$\textit{maximize} \sum_{i \in \mathcal{S}^+} V(i)$$
$$\textit{subject to } V(i) \leq q(i,u) + \sum_{j=1}^n P_{ij}(u) V(j), \quad \forall u \in \mathcal{U}(i), \forall i \in \mathcal{S}^+$$

3 Shortest paths

SP problem

Consider a graph with vertices $\mathcal{V}$ and weighted edges $\mathcal{C}$. $\mathbb{Q}_{s,t}$ is the set of possible s,t paths, and a path Q has cost J_Q. We seek the path with lowest cost $Q^* = \arg\min_{Q\in\mathbb{Q}_{s,t}} J_Q$. For this problem to have a solution the graph may not contain negatives cycles: $\forall i \in \mathcal{V}, \forall Q \in \mathbb{Q}_{i,i} : J_Q \geq 0$.

DFS problem

A Deterministic Finite State is a problem like the general case, but without w_k and where each $\mathcal{S}_k$ is finite. A closed loop approach offers no advantage in a deterministic problem, but we can still solve it with DPA.

DFS to SP

A DFS problem can be converted to an SP problem by creating a "layered" graph with layers $k = 0, ..., N+1$. The first layer only contains the node $(0, x_0)$ and is the starting position. Nodes in layers $k = 1, ..., N$ have nodes (k, x_k), $x_k \in \mathcal{S}_k$. Connections are between consecutive layers $k \to k+1$ and have weights

$$c = \min_{u\in\mathcal{U}_k(x_k)|x_{k+1}=f_k(x_k,u_k)} g_k(x_k, u)$$

The final layer contains just a virtual termination state t, and connections $N \to N+1$ are weighted with terminal costs $g_N(x_N)$.

SP to DFS

Since there are no negative cycles, an optimal s,t path will have at most $|\mathcal{V}|$ elements. We set $c_{i,i} = 0$ and formulate the problem as an $N = |\mathcal{V}| - 1$ stage DFS:

- $\mathcal{S}_0 = \{s\}$, $\mathcal{S}_k = \mathcal{V} \setminus \{t\}$, $\mathcal{S}_N = \{t\}$
- $\mathcal{U}_k = \mathcal{V} \setminus \{t\}$, $\mathcal{U}_{N-1} = \{t\}$
- $x_{k+1} = u_k$, $u_k \in \mathcal{U}_k$, $k = 0, ..., N-1$
- $g_k(x_k, u_k) = c_{x_k,u_k}$, $g_N(t) = 0$

3.1 LCA

The SP problem can be solved more efficiently with the Label Correcting Algorithm for a single starting node.

0. Place node s in `OPEN`, set $d_s = 0$, $d_j = \infty \, \forall j \in \mathcal{V} \setminus \{s\}$.
1. Remove node i from `OPEN` and run step 2 for every child j of i.
2. If $d_i + c_{i,j} < d_j$ and $d_i + c_{i,j} < d_t$ set $d_j = d_i + c_{i,j}$ and set i as the parent of j. If $j \neq t$, put j in `OPEN`.
3. Repeat from step 1 while `OPEN` $\neq \emptyset$.

Traversal order

- Depth-first: always remove the newest element of `OPEN`;
- Breadth-first: always remove the oldest element of `OPEN`;
- Best-first: always remove the element with lowest cost d_i.

3.2 A* algorithm

Perform LCA, and at each step 2, formulate some lower bound $h_j \geq 0$ of the j,t distance. Then change the condition $d_i + c_{i,j} < d_t$ to $d_i + c_{i,j} + h_j < d_t$.

3.3 HMMs and Viterbi algorithm

Consider a *Markov chain*: $\mathcal{S} = \{1, ..., n\}$ finite and $p_{w_k|x_k}$ given,

$$x_{k+1} = w_k, \quad x_k \in \mathcal{S},$$
$$P_{ij} := p_{w|x}(j|i), \quad \forall i, j \in \mathcal{S}.$$

When a state transition occurs, we obtain measurements

$$M_{ij}(z) := p_{z|x,w}(z|i, j), \quad z \in \mathcal{Z}.$$

Note that, given x_k and x_{k-1}, z_k is independent of all prior variables.

Defining $X_i = (x_i, ..., x_N)$ and $Z_i = (z_i, ..., z_N)$, we wish to find:

$$\hat{X}_0 = \arg\max_{X_0} p(X_0|Z_1)$$

$$p(X_0, Z_1) = ... = p(x_0) \prod_{k=1}^{N} P_{x_{k-1}x_k} M_{x_{k-1}x_k}(z_k)$$

This problem is solved by the SP problem:

$$\min_{X_0} \left(c_{(\mathrm{s},x_0)} + \sum_{k=1}^{N} c_{(k-1,x_{k-1}),(k,x_k)} \right)$$

where

$$c_{(\mathrm{s},x_0)} = \begin{cases} -\ln(p(x_0)) & \text{if } p(x_0) > 0 \\ \infty & \text{if } p(x_0) = 0 \end{cases}$$

$$c_{(k-1,x_{k-1}),(k,x_k)} = \begin{cases} -\ln(P_{x_{k-1}x_k} M_{x_{k-1}x_k}(z_k)) \\ \text{if } P_{x_{k-1}x_k} M_{x_{k-1}x_k}(z_k) > 0 \\ \infty \quad \text{otherwise} \end{cases}$$

This is a "layered" graph where each node represents a state x at time k. An artificial terminal node is added, connected to by layer $k = N$ at zero cost.

4 Deterministic continuous time

$$\dot{x}(t) = f(x(t), u(t)), \quad 0 \leq t \leq T$$
$$u(t) = \mu(\mathrm{x}, t), \quad u(t) \in \mathcal{U}, \forall t \in [0, T], \forall \mathrm{x} \in \mathcal{S}$$
$$J_\mu(t, \mathrm{x}) = h(x(T)) + \int_0^T g(x(\tau), u(\tau)) \mathrm{d}\tau$$

Assumption: for any admissible control law μ, initial time t and initial condition $x(t) \in \mathcal{S}$, there exists a unique state trajectory $x(\tau)$ that satisfies

$$\dot{x}(\tau) = f(x(\tau), u(\tau)), \quad t \leq \tau \leq T$$

4.1 HJB Equation

The Hamilton-Jacobi-Bellman equation is a sufficient but not necessary condition for optimality. If $V(t,x)$ is a solution to

$$\min_{u\in\mathcal{U}} \left[g(x,u) + \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(x,u) \right] = 0$$
$$\forall x \in \mathcal{S}, 0 \leq t \leq T$$
$$\text{s.t. } V(T,x) = h(x) \quad \forall x \in \mathcal{S}$$

then it is the optimal cost-to-go function, and the minimizing $\mu(t,x)$ is an optimal feedback law.

4.2 Minimum principle

Assumption: $f, g, h \in C^1$ in x
Pontryagin's minimum principle: for a given i.c. $x(0) = \mathrm{x} \in \mathcal{S}$, let $u(t)$ be an optimal control trajectory with associated state trajectory $x(t)$. Then there exists a trajectory $p(t)$ such that with $H(\mathrm{x}, \mathrm{u}, \mathrm{p}) := g(\mathrm{x}, \mathrm{u}) + \mathrm{p}^\top f(\mathrm{x}, \mathrm{u})$:

$$\dot{p}(t) = -\left.\frac{\partial H}{\partial x}\right|^\top_{x(t),u(t),p(t)}, \ p(T) = \left.\frac{\partial h(\mathrm{x})}{\partial x}\right|^\top_{x(T)}$$
$$u(t) = \arg\min_{\mathrm{u}\in\mathcal{U}} H(x(t), \mathrm{u}, p(t))$$
$$H(x(t), u(t), p(t)) = const. \forall t \in [0, T]$$

Fixed terminal state: Replace $p(T) = \left.\frac{\partial h(\mathrm{x})}{\partial x}\right|^\top_{x(T)}$ with $x(T) = x_T$.

Free initial state: instead of $x(0) = x_0$, a cost $l(x(0))$ is given. Add the condition: $p(0) = \left.\frac{\partial l}{\partial x}\right|^\top_{x(0)}$.

Free terminal time: we get $H(x(t), u(t), p(t)) = 0 \, \forall t \in [0, T]$

Time-varying system and cost: if f and/or g depend on t, we lose that $H = const.$

5 Non-standard problems

Some problems are not in the general (discrete-time) form, but can be reformulated as such.

5.1 Time lags

If the dynamics have a similar form: $x_{k+1} = f_k(x_k, x_{k-1}, u_k, u_{k-1}, w_k)$ we can contruct a state vector $\tilde{x}_k = (x_k, y_k, s_k)$ and modify the dynamics:

$$\tilde{x}_{k+1} = \tilde{f}_k(\tilde{x}_k, u_k, w_k) := \begin{bmatrix} f_k(x_k, y_k, u_k, s_k, w_k) \\ x_k \\ u_k \end{bmatrix}$$

5.2 Correlated Disturbances

Disturbances w_k correlated over time can sometimes be modeled as

$$w_k = C_k y_{k+1}, \quad y_{k+1} = A_k y_k + \xi_k$$

Where A_k and C_k are given and ξ_k, $k = 0, ..., N-1$ are independent random variables. Then we can augment the state as $\tilde{x}_k := (x_k, y_k)$ and update the dynamics:

$$\tilde{x}_{k+1} = \tilde{f}_k(\tilde{x}_k, u_k, \xi_k) := \begin{bmatrix} f_k(x_k, u_k, C_k(A_k y_k + \xi_k)) \\ A_k y_k + \xi_k \end{bmatrix}$$

5.3 Forecasts

w_k is independent of x_k and u_k, and we get a forecast y_k that w_k will attain a distribution from a given family $\{p_{w_k|y_k}(\cdot|1), ... p_{w_k|y_k}(\cdot|m)\}$. If $y_k = i$, then $w_k \sim p_{w_k|y_k}(\cdot|i)$. The forecast itself has a given *a priori* distribution:

$$y_{k+1} = \xi_k$$

where $\xi_k \sim p_{\xi_k}(i)$ are independent random variables.

Augmented state vector: $\tilde{x}_k := (x_k, y_k)$, disturbance $\tilde{w}_k := (w_k, \xi_k)$ with distribution

$$p(\tilde{w}_k|\tilde{x}_k, u_k) = p(w_k|y_k)p(\xi_k)$$

Dynamics:

$$\tilde{x}_{k+1} = \tilde{f}_k(\tilde{x}_k, u_k, \tilde{w}_k) := \begin{bmatrix} f_k(x_k, u_k, w_k) \\ \xi_k \end{bmatrix}$$

The DPA then becomes:

$$J_N(\tilde{\mathrm{x}}) = J_N(\mathrm{x}, \mathrm{y}) = g_N(\mathrm{x}), \quad \mathrm{x} \in \mathcal{S}_N, \mathrm{y} \in \{1, ..., m\}$$
$$J_k(\tilde{\mathrm{x}}) = J_k(\mathrm{x}, \mathrm{y}) = \min_{\mathrm{u}\in\mathcal{U}_k(x_k)} \underset{(w_k|y_k=\mathrm{y})}{\mathbb{E}}$$
$$\left[g_k(\mathrm{x}, \mathrm{u}, w_k) + \sum_{i=1}^{m} p_{\xi_k}(i) J_{k+1}(f_k(\mathrm{x}, \mathrm{u}, w_k), i) \right]$$
$$\forall \mathrm{x} \in \mathcal{S}_k, \ \forall \mathrm{y} \in \{1, ..., m\}, \ \forall k = N-1, ..., 0.$$