Elements of Optimal Control: ICDNS

MSci/MSc

Nick Jones

nick.jones@imperial.ac.uk

Optimal Control

What we will cover.
Why this is biologically relevant.

Reviews by Todorov and Kappen [1, 2] are short-ish introductions and the Bechhoefer review [3] also includes a brief treatment of Optimal control - I'll be drawing on these treatments in the following. The text on Reinforcement learning by Sutton and Barto [4] covers similar material and is available online in a browsable format. The books [5, 6] cover optimal control but not in the stochastic setting.

The simplest case

Optimal control is devoted to finding a set of controls which minimize an appropriately chosen cost function C.

We will consider a control problem which must be completed in finite, discrete, time T (finite horizon) and in which there is no noise (though we will consider the effect of unanticipated perturbations to the trajectory).

The simplest case II

Discrete time dynamics: $x_{t+1} = x_t + f(u_t, x_t, t)$. We'll switch to calling our state variable x (not y used previously).

Local cost of action u at time t and location x: $w(x, u, t)$. This might be the petrol cost of flying between two locations x and x' *in one timestep* $t \rightarrow t+1$: I start at x at t and arrive at x' at time $t+1$ and do so under control u (and under dynamics $f(u, x, t)$).

Final cost of destination location: $W(x_T)$. Measures how good/costly my final location is: does being in John o'Groats have a higher cost than being in Honolulu? It says nothing about the cost of getting to these locations.

Cost function

Consider starting at x_0 with a sequence of T controls: $u_{0:T-1} = u_0, u_1 ... u_{T-1}$.
If I know the controls and the dynamics $(f(u,x,t))$ I know the corresponding sequence of states $x_{1:T}$.
Cost: $C(x_0, u_{0:T-1}) = \sum_{t=0}^{T-1} w(u_t, x_t, t) + W(x_T)$.
What is u? As well as being an external control signal, r, it could be constrained to be a function of the existing state $g(x, \mathbf{f})$: in this case (for given g) optimizing the control means finding the best values of the parameters $\mathbf{f}$ to minimize our cost.

Cost function II

Cost: $C(x_0, u_{0:T-1}) = \sum_{t=0}^{T-1} w(u_t, x_t, t) + W(x_T)$.
(the petrol cost of flying between successive points $x_{0:T-1}$ + how much my final destination is desirable).
Minimum Cost:

$$J(0, x_0) = \min_{u_{0:T-1}} [C(x_0, u_{0:T-1})]$$

Optimal control:

$$u^*_{0:T-1} = \arg\min_{u_{0:T-1}} [C(x_0, u_{0:T-1})]$$

Optimal cost-to-go

Cost: $C(x_0, u_{0:T-1}) = \sum_{t=0}^{T-1} w(u_t, x_t, t) + W(x_T)$.

Minimum Cost:

$$\min_{u_{0:T-1}} [C(x_0, u_{0:T-1})]$$

Optimal control:

$$u^*_{0:T-1} = \arg\min_{u_{0:T-1}} [C(x_0, u_{0:T-1})]$$

Cost-to-go:

$$C(x_t, u_{t:T-1}) = \sum_{t}^{T-1} w(u_t, x_t, t) + W(x_T)$$

Optimal cost-to-go:

$$J(t, x) = \min_{u_{t:T-1}} [C(x_t, u_{t:T-1})]$$

This expression (sometimes called the optimal value function) will help us use Dynamic Programming to find the optimal control.

Approach to finding the optimal control

We want to find the T controls $u^*_{0:T-1}$ which minimize $C(x_0, u_{0:T-1})$.

It turns out that it is easiest to calculate the following vector field: $u'(t,x)$. This is the optimal control for time $t \to t+1$ *at all points in space x and time t.*

If we have $u'(t,x)$ then we can find $u^*_{0:T-1}$ by finding out our initial condition x_0 then finding $u'(t=0,x_0)$. We then use $u'(t=0,x_0)$ to calculate the new co-ordinate x_1 (given knowledge of the dynamics $f(u,x,t)$). We then set $t=1$ and plug x_1 into $u'(t,x)$ and continue until we've found the full vector $u^*_{0:T-1}$.

This works if we have the vector field $u'(t,x)$. How do we find $u'(t,x)$?

It'll turn out that we get $u'(t,x)$ for free when we recursively construct the optimal cost-to-go scalar field: $J(t,x)$.

Interpreting $u'(t,x)$

$\diamond$ The field $u'(t,x)$ is a magical object (and this hints at how computationally challenging some optimal control problems can be). It says 'if you are at location x, t I know what you should do next. If you follow my advice for this timestep t and refer to me at all successive timesteps then I guarantee you'll minimize your cost *from now on*.'

$\diamond$ As such, even if at the last time step $t-1$ you did something stupid, were buffeted by a random force, and you are at a suboptimal x, t you don't need to worry: referring to $u'(t,x)$ and recursively calculating your controls from now on will take you to your destination at minimum total expense. Thus, although we are not considering a noisy scenario, if we think that randomizing events are rare (and we don't have a model for them, or they have a very simple structure) using $u'(t,x)$ will be useful.

$\diamond$ We might think that $u'(t,x)$ is somehow an 'open-loop control' vector field since we have precomputed it based on knowledge of our dynamics. However $u'(t,x)$ specifies the optimal control under all circumstances (x,t) and, as noted above, it might be that $u = g(x, \mathbf{f})$ involves state-feedback. $u'(t,x)$ is precomputed but it could be either open or closed loop control.

Finding the optimal cost-to-go field yields the optimal control field

We know our optimization has T steps and that $J(T,x) = W(x)$. We further know that the optimal cost-to-go, at time T-1 and location x, is the cheapest combination of a) the cost of moving to point $x + f(t,x,u)$ (or deploying control u) and b) the optimal cost-to-go from point $x + f(t,x,u)$:

$$J(T-1,x) = \min_{u_{T-1}} \left[(w(u_{T-1},x,t) + J(T, x + f(t,x,u)) \right]$$

$$u'(x, T-1) = \arg\min_{u_{T-1}} \left[(w(u_{T-1},x,t) + J(T, x + f(t,x,u)) \right]$$

Here $J(T, x + f(T-1,x,u)) = W(x + f(T-1,x,u))$.
Thus for all points $T-1, x$ we calculate the optimal cost-to-go to the (horizon) time T

Finding the optimal cost-to-go field yields the optimal control field II

We can find $\forall x$:

$$J(T-1,x) = \min_{u_{T-1}} \left[(w(u_{T-1},x,t) + J(T,x+f(t,x,u)) \right]$$

Given this we can then back-up a time-step and find:

$$J(T-2,x) = \min_{u_{T-2}} \left[(w(u_{T-2},x,t) + J(T-1,x+f(t,x,u)) \right].$$

We proceed in this manner moving back in time and calculate the scalar optimal cost-to-go field $J(t,x)$. To find $J(t,x)$ we are also calculating $u'(x,t)$ (that's why I said we get the optimal control field for free).
The trick is that we are keeping track of $J(t,x)$ as we progress back-in time - we only need to refer to $J(t,x)$ when we calculate $J(t-1,x)$.

Dynamic programming

Dynamic programming

1. Initialise with $J(T, x) = W(x)$
2. Backwards: for $t = T - 1, ..., 0$ and all x find

$$u'(x,t) = \underset{u}{\operatorname{argmin}} \left[\left(w(u,x,t) + J(t+1, x + f(t,x,u)) \right) \right]$$

$$J(t,x) = \left[w(u'(x,t), x, t) + J(t+1, x + f(t, x, u'(x,t))) \right]$$

3. Forwards: For $t = 0, ..., T$ and using i.c. $t = 0$, $x = x_0$.

$$x^*_{t+1} = x^*_t + f(t, x^*_t, u'(x^*_t, t))$$

This is after Ref. [2] and will be used in our practical.

What is dynamic programming?

An informal answer:

Break the problem into sub-problems (find $J(t,x)$ given $J(t+1,x)$) and store the results as you progress through the algorithm (store $J(t,x)$ to help find $J(t-1,x)$).
In our case if we had m points in space, we solved the optimization problem by performing $T \times m$ optimizations. You might have thought that you'd need something like m^T timesteps as we test each possible trajectory for optimality. Note that the DP solution is still disgusting if we have our space being d dimensional since we then have $T \times m^d$ steps.

Continuous time setting

Dynamics:

$$x_{t+dt} = x_t + f(u_t, x_t, t)dt$$

Cost:

$$C(x_0, u(0 \to T)) = W(x_T) + \int_0^T d\tau w(x(\tau), u(\tau), \tau)$$

where $u(0 \to T)$ is a function (of time) (we had $u_{0:T-1}$ previously). Optimal cost-to-go:

$$J(t, x) = \min_u (w(x, t, u)dt + J(t + dt, x + f(x, u, t)dt))$$

Where our min is over the control at time t and location x. This is the same as previously: our optimal cost-to-go from (x, t) is the best combination of a control that moves us to $(x + f(x, u, t)dt, t + dt)$ and then the optimal cost-to-go from $(x + f(x, u, t)dt, t + dt)$.

Continuous time setting II

Optimal cost-to-go:

$$J(t,x) = \min_u (w(x,t,u)dt + J(t+dt, x + f(x,u,t)dt))$$

Taylor expanding:

$$J(t,x) \simeq \min_u (w(x,t,u)dt + J(t,x) + \partial_t J(t,x)dt + \partial_x J(t,x) f(x,u,t)dt))$$

Setting $dt \to 0$ yields the below.

Hamilton-Jacobi-Bellman Equation

$$-\partial_t J(t,x) = \min_u (w(x,t,u) + f(x,u,t)\partial_x J(t,x))$$

with boundary condition
$J(T,x) = W(x)$.
The corresponding optimal control field is

$$u'(x,t) = \operatorname{argmin}_u (w(x,t,u) + f(x,u,t)\partial_x J(t,x))$$

Interpreting HJB

Hamilton-Jacobi-Bellman Equation

$$-\partial_t J(t,x) = \min_u (w(x,t,u) + f(x,u,t)\partial_x J(t,x))$$

with boundary condition
$J(T,x) = W(x)$.

$$\min_u (w(x,t,u)dt + \{\partial_t J(t,x) + f(x,u,t)\partial_x J(t,x)\}dt) = 0$$

Can be interpreted as 'for optimal controls, the cost for being at x, t and applying u for time dt should be balanced by the corresponding change in cost-to-go': i.e. the cost of my move must yield an exactly balancing decrement in my remaining cost to go *if my move* $f(x,u,t)dt$ *(and thus my control,* u*) is optimal*. We can solve the HJB equation by initializing at $J(T,x) = W(x)$ and evolving this function back in time. When we perform this we will generate the field $u'(x,t)$.

Bibliography

[1] E. Todorov, Optimal control theory, In Bayesian Brain: Probabilistic Approaches to Neural Coding, Doya K at al (eds), chap 12, 269, MIT Press 2006. Free online.

[2] H.J. Kappen, Optimal control theory and the linear Bellman Equation. Book chapter. Free online.

[3] J. Bechhoefer, Feedback for physicists: A tutorial essay on control, Reviews of Modern Physics, 77, 783, 2005. Free to you online.

[4] R.S. Sutton and A.G. Barto, Reinforcement learning, MIT Press, 1998. Free online.

[5] J. Zabczyk, Mathematical control theory, Birkauser, 1995. Free to IC students.

[6] E. Sontag, Mathematical control theory, Springer, 1998. Free online.