Approximate Dynamic Programming for Linear Convex Stochastic Control

Brendan O'Donoghue and Stephen Boyd
Information Systems Laboratory, Electrical Engineering, Stanford University

The stochastic control problem

- consider a linear dynamical system
  - with state $x_t \in \mathcal{X} \subseteq \mathbf{R}^n$ and action $u_t \in \mathcal{U} \subseteq \mathbf{R}^m$
  - which propagates over time according to
    $$x_{t+1} = A(w_t)x_t + B(w_t)u_t + c(w_t), \quad t = 0, 1, \ldots,$$
  - where $w_t \in \mathcal{W}$ is the noise and A, B, c are known functions
- with time-invariant stage cost function of the state and action
  - $\ell : \mathbf{R}^n \times \mathbf{R}^m \times \mathcal{W} \to \mathbf{R} \cup \{\infty\}$
  - which we assume is convex
  - we encode any constraints on x or u into ℓ, $i.e.$
    $$\ell(x,u) = \infty, \quad \forall (x,u) \notin \mathcal{X} \times \mathcal{U}$$
- the stochastic control problem is to find a state feedback control policy, $\phi : \mathcal{X} \to \mathcal{U}$
  - which we assume it is causal and time-invariant
  - which maps the system state to an action
    $$u(t) = \phi(x(t)), \quad t = 0, 1, \ldots,$$
- in order to minimize the average cost over time
  $$J_\phi = \limsup_{T\to\infty} (1/T)\, \mathbf{E} \sum_{t=0}^{T-1} \ell(x_t, u_t, w_t)$$
- where the optimal average cost $J^\star = \inf_\phi J_\phi$

Dynamic Programming

- we define a modified Bellman operator
  $$(\mathcal{S}f)(x,u) = \mathop{\mathbf{E}}_w \left[\ell(x,u) + f(Ax + Bu + c)\right]$$
  for any $f : \mathbf{R}^n \to \mathbf{R}$
- if we can find a function $V^\star : \mathbf{R}^n \to \mathbf{R}$ and a constant $\alpha^\star \in \mathbf{R}$ that satisfy
  $$\alpha^\star + V^\star = \min_u \mathcal{S}V^\star, \quad \forall x$$
- then it can be shown that $J^\star = \alpha^\star$
- and the optimal control policy is given by
  $$\phi^\star(x) = \operatorname*{argmin}_u \mathcal{S}V^\star(x,u)$$
- $V^\star$ is known as the *value function* of the dynamical system
- finding $V^\star$ and $\alpha^\star$ is hard in general
- they are the solutions to the following linear program
  $$\begin{array}{ll} \text{maximize} & \alpha \\ \text{subject to} & \alpha + V \le \mathcal{S}V, \quad \forall (x,u) \\ \text{variables} & V : \mathbf{R}^n \to \mathbf{R} \text{ and } \alpha \in \mathbf{R} \end{array} \qquad (1)$$
- this problem is convex, but computationally intractable in most cases
  - we are optimizing over an infinite number of variables
    solution - approximate dynamic programming
  - we have an infinite number of constraints over infinite indices x and u
    solution - cutting set method

Approximate dynamic programming

- we restrict the class of functions we are interested in to obtain an *approximate value function* $\hat{V}$, we restrict $\hat{V}$ to be
  - linear in some parameter $r = [r_1, \ldots, r_K]$
  - convex (equivalent to $r \in \mathcal{C}$ for some convex set $\mathcal{C}$)
  $$\hat{V}(x) = \sum_{k=1}^{K} r_k \phi_k(x),$$
  where ϕ_k, $k = 0, \ldots, K$, are fixed basis functions
- with this restriction the problem (1) has $K+1$ variables
- resultant $\hat{V}$ and $\hat{\alpha}$ are guaranteed lower bounds on true values if we can solve (1) exactly
- we can evaluate the approximate control policy using Monte Carlo methods
  $$\hat{\phi}(x) = \operatorname*{argmin}_u \mathcal{S}\hat{V}(x,u)$$

Cutting set method

an iterative method to solve problems with infinite constraints

- *Optimization*
  solve the problem with a finite subset of constraints, $\hat{\mathcal{Z}}$
- *Pessimization*
  invoke an oracle to identify violated constraints, append them to $\hat{\mathcal{Z}}$
- repeat until convergence by some measure

Optimization - sampled problem

- let $\hat{\mathcal{Z}} \subset (\mathbf{R}^n \times \mathbf{R}^m)^l$ be a collection of $l < \infty$ state-action pairs
- solve an approximate version of (1)
  $$\begin{array}{ll} \text{maximize} & \hat{\alpha} \\ \text{subject to} & \hat{V} + \hat{\alpha} \le \mathcal{S}\hat{V}, \quad \forall (x,u) \in \hat{\mathcal{Z}} \\ & r \in \mathcal{C} \\ \text{variables} & r \in \mathbf{R}^K \text{ and } \alpha \in \mathbf{R} \end{array}$$

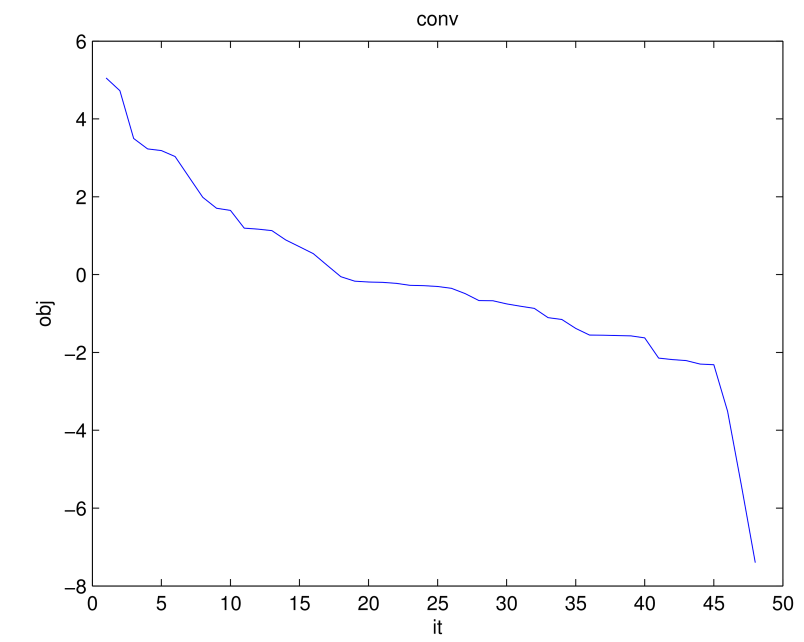
Pessimization - convex-concave procedure

- to find violated constraints in (1) we want to solve
  $$\begin{array}{ll} \text{minimize} & \mathcal{S}\hat{V} - \hat{V}, \\ \text{variables} & x \in \mathbf{R}^n \text{ and } u \in \mathbf{R}^m \end{array}$$
- this is a difference of convex functions, therefore hard in general
- we can identify local minima using the *convex-concave procedure*
  - select starting point $\bar{x}$ and some $\epsilon > 0$
  - solve
    $$\begin{array}{ll} \text{minimize} & \mathcal{S}\hat{V}(x,u) - \nabla\hat{V}(\bar{x})^T(x - \bar{x}) \\ \text{variables} & x \in \mathbf{R}^n \text{ and } u \in \mathbf{R}^m \end{array}$$
  - set $\bar{x} := x$, repeat until $\|x - \bar{x}\| \le \epsilon$
- we approximate the concave function $-\hat{V}$ by its gradient at $\bar{x}$ which is everywhere an upper bound on $-\hat{V}$

Example: Dynamic portfolio optimization

- $x_t \in \mathbf{R}^n$ dollar value of $n = 20$ assets at time t
- $u_t \in \mathbf{R}^n$ dollar amount we buy or sell of each asset at time t
  - the portfolio propagates over time according to
    $$x_{t+1} = A_t(x_t + u_t)$$
  - where $A_t = \mathbf{diag}(s_t)$ and $(s_t)_i > 0$ is the return of asset i at time t
  - let $\mathbf{E}[s_t] = \bar{s}$ and $\mathbf{E}[s_t s_t^T] = \Sigma$ for all t
  - at each time-step we incur a transaction fee, given by $\kappa\|u_t\|_1$
  - the stage cost is a risk-revenue trade-off of the form
    $$\ell(x,u) = \mathbf{1}^T u + \kappa\|u\|_1 + \gamma x^T \Sigma x$$
    where we set $\gamma = 0.1$
- we restrict $\hat{V}$ to be quadratic, $i.e.$ to have the form
  $$\hat{V}(x) = x^T P x + 2p^T x$$

Figure 1: Convergence rate of $\hat{\alpha}^{(k)}$



- results:
  - we generated 5 random time traces from 5 random starting portfolios
  - average cost of the approximate policy, $\hat{J}$ is compared to cost from model predictive control (mpc) with horizon $T = 10$, J^{mpc}
  - difference between $\hat{\alpha}$ and $\hat{J}$ is a rough sub-optimality gap

$\hat{\alpha}$	$\hat{J}$	J^{mpc}
-19.1	-14.6	43.3

Figure 2: Sample portfolio trajectories