VALUE FUNCTION APPROXIMATION

So far we have taken a tabular view of estimation and control; value functions and policies are represented as tables, indexed by states (and actions).

But what if the state space S is huge, and possibly infinite? What are some examples?

In such situations, we can attempt to characterize value functions as members of a restricted family of functions.

For example, we might assume a parametric model for the value function

$$V(s) = V(s; \theta)$$

Example The most important example is the class of linear functions,

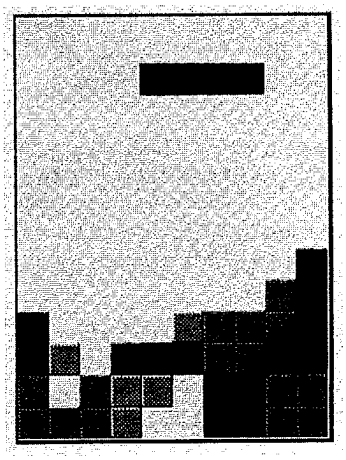
(A) $$V(s;\theta) = \langle \Phi(s), \theta \rangle =$$

where

$$\Phi(s) =$$

Each $\phi^{(j)}(s)$ is a *feature* that hopefully conveys some useful information about the value of a policy in state s.

Example Tetris



(B) # of states $\approx$

where

w = width
h = height
m = # of shapes

For $w=10$, $h=20$, $m=7$

$\rightarrow 10^{61}$ states

reward = +1 each time a row is completed

Possibe features include

- h_k = height of k^{th} column, $k = 1, \ldots, w$
- $|h_k - h_{k+1}|$, $k = 1, \ldots, w-1$
- height of highest "hole" in each column
- the current falling piece
- ⋮

Function Approximation / Regression

We want to find the best θ. To measure performance we can use the MSE

$$MSE(\theta) = E_\pi \left\{ (V^\pi(s) - V(s;\theta))^2 \right\}$$

$$= \sum_s p(s) \left(V^\pi(s) - V(s;\theta) \right)^2$$

would be an integral for continuous S

where $p(s)$ = probability of being in state s when following π.

Of course $p(s)$ is probably unknown. But suppose we can generate data $s_0, a_0, r_1, s_1, a_1, r_2, s_2, a_2, \ldots, s_n$

Then we can approximate

(c) $MSE(\theta) \approx$

In addition, $V^{\pi}(s)$ is unknown, so we may instead substitute one of our usual "targets", e.g.

- R_t = return after 1st visit to s_t
- $r_{t+1} + \gamma V(s_{t+1}; \theta)$
- R_t^{λ}

(D) Therefore, estimating V^{π} reduces to

In the case where $V(s;\theta) = \theta^T I(s)$, we need to solve

Implementation

For efficiency, it is desirable that we can perform incremental updates to the regression solution.

For this reason, gradient-descent algorithms are often favored. Regression models that can be fit using gradient descent include

- linear models
- neural networks.

The general form of gradient descent is

$$\theta_{t+1} = \theta_t + \alpha \left[v_t - V(s_t; \theta_t) \right] \nabla_\theta V(s_t; \theta_t)$$

where v_t is the target (i.e., the surrogate for $V^\pi(s_t)$).

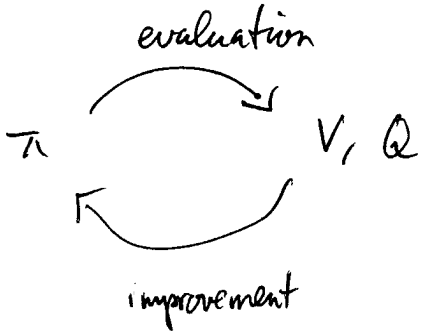
If v_t is an unbiased estimated of $V^\pi(s_t)$, then G.D. converges to a <u>local</u> optimum. In the case of a linear model, this implies <u>global</u> optimality.

For TD(λ) type updates, v_t is biased, but for linear models it has been shown that

(E) $MSE(\theta_\infty) \leq$

Control

One again, to improve a policy we may use the framework of generalized policy iteration:



- Evaluation: $Q^{\pi}(s,a) \approx Q(s,a;\theta)$

$$\theta_{t+1} = \theta_t + \alpha \left[v_t - Q(s_t, a_t; \theta_t) \right] \nabla_\theta Q(s_t, a_t; \theta_t)$$

- Improvement:
  - if $\mathcal{A}$ is finite:

$$\pi_t(s) = \arg\max_a Q(s,a;\theta_t)$$

  - if $\mathcal{A}$ is infinite:

numerical maximization of $Q(s,a;\theta_t)$ w.r.t a?

Extensions to Sarsa (λ) and $Q(\lambda)$ for on-policy and off-policy exploration.
↳ challenging if $\mathcal{A}$ large or infinite

Key A. $\sum_{j=1}^{m} \theta^{(j)} \varphi^{(j)}(s)$, $\underline{\phi}(s) = \left[\varphi^{(1)}(s), \ldots, \varphi^{(m)}(s) \right]^T$

B. $m 2^{wh}$ C. $MSE(\theta) \approx \frac{1}{n} \sum_{t=1}^{n} \left(V^{\pi}(s_t) - V(s_t;\theta) \right)^2$

D. least squares regression, least squares linear regression

E. $MSE(\theta_\infty) \leq \frac{1-\gamma\lambda}{1-\gamma} MSE(\theta^*)$