Lecture 4 and 5: Modern reinforcement learning—LSTD and policy gradients

Reinforcement Learning and Decision Making MLSALT7, Lent 2016

Matthew W. Hoffman, Zoubin Ghahramani, Carl Edward Rasmussen

Department of Engineering
University of Cambridge

http://mlg.eng.cam.ac.uk/teaching/mlsalt7/

The value function (again!)

Once again, the value function V can be defined as the unique fixed point of the Bellman operator, $\mathsf{V}^\pi = \mathsf{T}^\pi \mathsf{V}^\pi$

$$(\mathcal{T}^\pi V)(x) = r(x) + \gamma \int_{\mathcal{X}} P(x'|x, \pi(x))\, V(x')\, dx'$$

or more concisely, as

$$\mathcal{T}^\pi V = r + \gamma P^\pi V.$$

Linear function approximation for the value function

Instead of computing V^π for every state (which may not even be possible!) we will approximate this with a weighted combination of features $\phi : \mathcal{X} \to \mathbb{R}^k$, i.e.

$$V^\pi(x) \approx \phi(x)^\top \theta$$

Let's return to the definition of the value function, by way of the Bellman operator, and apply it to the approximator:

$$\mathcal{T}^\pi\left[\phi(x)^\top \theta\right] = r(x) + \gamma P^\pi\left[\phi(x)^\top \theta\right]$$

What's wrong with applying the Bellman operator to this approximation?

Linear function approximation for the value function

Instead of computing V^{π} for every state (which may not even be possible!) we will approximate this with a weighted combination of features $\phi : \mathcal{X} \to \mathbb{R}^{k}$, i.e.

$$V^{\pi}(x) \approx \phi(x)^{\top}\theta$$

Let's return to the definition of the value function, by way of the Bellman operator, and apply it to the approximator:

$$\mathcal{T}^{\pi}\left[\phi(x)^{\top}\theta\right] = r(x) + \gamma P^{\pi}\left[\phi(x)^{\top}\theta\right]$$

What's wrong with applying the Bellman operator to this approximation?

More concretely: does there exist a θ^{*} such that $\mathcal{T}^{\pi}\left[\phi(x)^{\top}\theta\right] = \phi(x)^{\top}\theta^{*}$?

Linear function approximation in general

Let's go back to what it means to approximate some function. We want to find the best approximation under some norm,

$$\mathsf{f}(x) \approx \phi(x)^\top \theta^* \text{ where } \theta^* = \arg\min_{\theta} \|\phi(x)^\top \theta - \mathsf{f}(x)\|$$

Linear function approximation

However $\mathcal{T}^{\pi}\Phi w$ may not necessarily lie in the span of Φ. Instead, we will introduce a projection operator Π such that

$$\Pi v = \Phi \operatorname*{arg\,min}_{u \in \mathbb{R}^k} \|\Phi u - v\|^2.$$

Linear function approximation

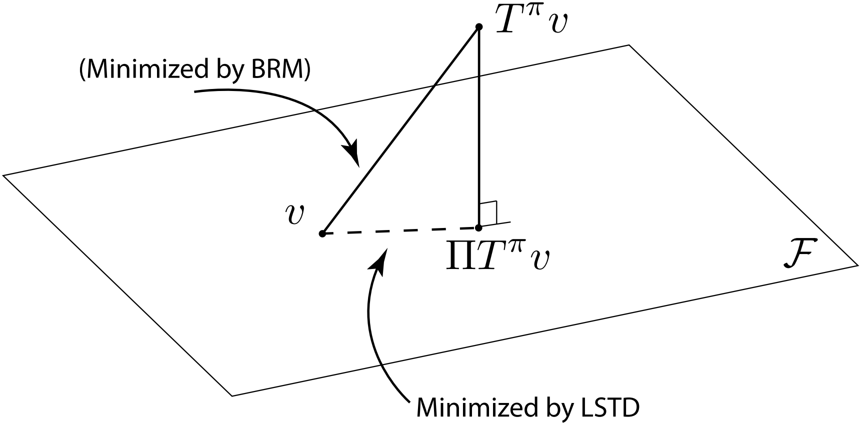
However $\mathcal{T}^\pi \Phi w$ may not necessarily lie in the span of Φ. Instead, we will introduce a projection operator Π such that

$$\Pi v = \Phi \arg\min_{u \in \mathbb{R}^k} \|\Phi u - v\|^2.$$

We can then define the following fixed point:

$$\Phi w = \Pi \mathcal{T}^\pi \Phi w.$$

A bit of geometry



Existence and uniqueness of the solution

First, the Bellman operator is a γ-contraction, i.e. for any y, z

$$\|\mathcal{T}^\pi \mathsf{y} - \mathcal{T}^\pi z\| \leqslant \gamma \|\mathsf{y} - z\|$$

and the projection operator is non-expansive, i.e. for any y

$$\|\Pi \mathsf{y}\| \leqslant \|\mathsf{y}\|.$$

Combining these two means that $\Pi\mathcal{T}^\pi$ is a γ-contraction, and due to the Banach fixed-point theorem there exists a unique fixed point $\hat{v} = \Pi\mathcal{T}^\pi\hat{v}$.

Note! This does not mean that there is a unique solution $\Phi w = \hat{v}$.

Finding the fixed point

We can write the fixed point as the following:

$$
\begin{aligned}
w &= \arg\min_{u \in \mathbb{R}^k} \|\Phi u - (r + \gamma P^\pi \Phi w)\|^2 \\
&= (\Phi^\top \Phi)^{-1} \Phi^\top (r + \gamma P^\pi \Phi w) \\
&\ \ \vdots \\
&= \underbrace{(\Phi^\top(\Phi - \gamma P^\pi \Phi))^{-1}}_{A^{-1}} \underbrace{\Phi^\top r}_{b}
\end{aligned}
$$

A "model-free" approach

The previous approach required forming the entire feature matrix Φ and also required the transition model P^π.

Instead we will assume samples (x_i, a_i, x_i') generated on-policy and form

$$\hat{\Phi} = \begin{bmatrix} \phi(x_1)^T \\ \vdots \\ \phi(x_m)^T \end{bmatrix}, \ \hat{\Phi}' = \begin{bmatrix} \phi(x_1')^T \\ \vdots \\ \phi(x_m')^T \end{bmatrix}, \ \hat{r} = \begin{bmatrix} r(x_1) \\ \vdots \\ r(x_m) \end{bmatrix}.$$

Solving for the fixed point is then given by

$$w = (\hat{\Phi}^T(\hat{\Phi} - \gamma\hat{\Phi}'))^{-1}\hat{\Phi}^T\hat{r}$$

Moving on to policy improvement

The previous method is model-free for policy evaluation, but in order to improve the policy (moving from LSTD to LSPI) we would need a model.

Instead, learn the Q-function,

$$Q^\pi(x, a) = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t r(x_t) \Big| x_0 = x, a_0 = a, \pi\Big]$$

$$\pi^{\text{new}}(x) = \arg\max_a Q^\pi(x, a)$$

Learning the Q-function

Define the Bellman operator as

$$\mathsf{T}^\pi \mathsf{Q} = r + \gamma \mathsf{P}\mathsf{H}^\pi \mathsf{Q}$$

where H^π is called Π^π in (Lagoudakis and Parr), and $\mathsf{P}\mathsf{H}^\pi$ basically describes the probability of transitioning from $(x, a) \rightarrow (x', a')$.

Learning the Q-function

Define the Bellman operator as

$$T^{\pi}Q = r + \gamma P H^{\pi} Q$$

where H^{π} is called Π^{π} in (Lagoudakis and Parr), and PH^{π} basically describes the probability of transitioning from $(x, a) \to (x', a')$.

Now, when we move to the empirical version we have samples (x, a, x') not necessarily drawn on-policy. We construct

$$\hat{\Phi} = \begin{bmatrix} \phi(x_1, a_1)^T \\ \vdots \\ \phi(x_m, a_m)^T \end{bmatrix}, \ \hat{\Phi}' = \begin{bmatrix} \phi(x'_1, \pi(x'_1))^T \\ \vdots \\ \phi(x'_m, \pi(x'_m))^T \end{bmatrix}, \ \hat{r} = \begin{bmatrix} r(x_1) \\ \vdots \\ r(x_m) \end{bmatrix}.$$

The solution for w is the same as before.

Parameterized policies

Rather than indirectly parameterizing a policy via its value function we can directly parameterize the policy

$$\pi_\theta(a|x)$$

with parameters θ. Note that now we'll go back to assuming a stochastic policy (we'll see why shortly) and return to the value of a full trajectory,

$$J(\theta) = \mathbb{E}_\theta\Big[\sum_{t=0}^{k} \gamma^t r(X_t)\Big]$$

Example policies

For a discrete space we can define policies of the form:

$$\pi_\theta(a|x) = \frac{\exp(\theta_{ax})}{\sum_{a'} \exp(\theta_{a'x})}$$

Or a common policy for continuous spaces is:

$$\pi_\theta(a|x) = Kx + m \quad \text{for } \theta = (K, m)$$

A little notation

Let's let $\tau = (x_0, \dots, x_k, a_0, \dots, a_k)$ denote a single trajectory. How can we write the probability of τ?

$$p_\theta(\tau) = p(x_0)\pi_\theta(a_0|x_0) \prod_{n=1}^{k} p(x_n|x_{n-1}, a_{n-1})\pi_\theta(a_n|x_n)$$

We can also write the reward for a trajectory as

$$R(\tau) = \sum_{t=0}^{k} \gamma^t r(x_t)$$

for which our objective becomes

$$J(\theta) = \mathbb{E}_\theta[R(\tau)]$$

The policy gradient

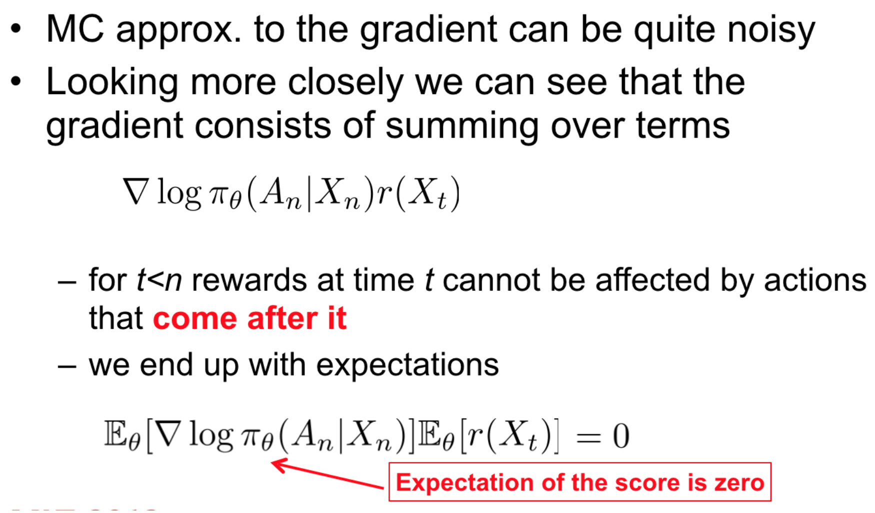
Now we can expand our gradient as follows:

$$\begin{aligned}\nabla J(\theta) &= \nabla \int R(\tau)\, p_\theta(\tau) d\tau \\ &= \int R(\tau)\, \nabla p_\theta(\tau) d\tau \\ &= \int R(\tau)\, p_\theta(\tau)\, \nabla \log p_\theta(\tau) d\tau && \text{because } \nabla \log f = \nabla f / f \\ &= \int R(\tau)\, p_\theta(\tau) \Big[\sum_{n=0}^{k} \nabla \log \pi_\theta(a_n | x_n) \Big]\end{aligned}$$

What does this mean?

- We need only sample trajectories
- We need to know the policy and its gradient, but that's fine because we decide on that

Exploting independence



- MC approx. to the gradient can be quite noisy
- Looking more closely we can see that the gradient consists of summing over terms

$$\nabla \log \pi_\theta(A_n|X_n)r(X_t)$$

  - for *t*<*n* rewards at time *t* cannot be affected by actions that **come after it**
  - we end up with expectations

$$\mathbb{E}_\theta[\nabla \log \pi_\theta(A_n|X_n)]\mathbb{E}_\theta[r(X_t)] = 0$$

Expectation of the score is zero

Exploting independence cont'd

$$R(\tau)\left[\sum_{n=0}^{k}\nabla\log\pi_\theta(A_n|X_n)\right]$$

$$=\sum_{n=0}^{k}\nabla\log\pi_\theta(A_n|X_n)\sum_{t=0}^{k}\gamma^t r(X_t)$$

Expand the term inside the expectation…

Under expectation we can eliminate rewards for *t<n*

$$=\sum_{n=0}^{k}\nabla\log\pi_\theta(A_n|X_n)\underbrace{\sum_{t=n}^{k}\gamma^t r(X_t)}_{\text{Call this } R_n(\tau)}$$

Including those terms would only add noise!

Baselines

- Let $R_n(\cdot)$be the sum of rewards **after** step *n*
- We can now write

$$\nabla J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{n=0}^{k} R_n(\tau^{(i)}) \nabla \log \pi_\theta(a_n^{(i)} | x_n^{(i)})$$

for *N* sample trajectories $\tau^{(i)}$

- Following this gradient coincides with
  - REINFORCE [Williams, 1992]
  - GPOMDP [Baxter and Bartlett, 2001]

The policy gradient

- For the same reason that we eliminated rewards with *t*<*n*, we can write the reward as

$$R_n(\tau^{(i)}) = \sum_{t=n}^{k} r(x_t^{(i)}) - b_t$$

Variance of gradient depends on magnitude of rewards.

  for **baseline** quantities independent of the *n*th action
  - can depend on previous rewards! (say the average)
  - this baseline can then be selected to reduce variance
  - see [Greensmith et al., 2001], [Riedmiller, et al, 2008]