Advanced Topics in Reinforcement Learning

Lecture 13: Linear Function Approximation and On-policy Control

Josiah Hanna
University of Wisconsin — Madison
Announcements

• Homework 3 due Thursday of next week.

• Begin reading chapter 11 for next week.

• Midterm survey and evaluation.

• Looking ahead: https://pages.cs.wisc.edu/~jphanna/teaching/2022fall_cs839/schedule.html
Interest and Emphasis

• So far, assumed we are updating states equally (same learning rate) but according to the on-policy state distribution, $\mu$.

• We may wish to emphasize some states more.

• State interest, $I_t$, represents how much we care about accurate estimation in state $S_t$.

• Emphasis is a learned multiplier on the learning rate.

  • $M_t \leftarrow I_t + \gamma M_{t-1}$

  • $w_{t+1} \leftarrow w_t + \alpha M_t[R_t - \hat{v}(S_{t+1}, w) - \hat{v}(S_t, w)] \nabla \hat{v}(S_t, w)$
Interest and Emphasis

- Interest is \((1, 0, 1, 0)\)

- Semi-gradient 2-step TD converges to weight vector \((3.5, 1.5)\)

- Emphatic 2-step TD converges to weight vector \((4, 2)\)
On-Policy Control

- As usual, for control we will estimate action-values, $\hat{q}(s, a, w)$.

- For linear function approximation, features are now a function of (s,a) pairs, $x(s, a)$.

- Function approximation often inherently means that making $\hat{q}(s, a, w)$ more accurate at one state will make it less accurate at another state.

- Now making $\pi$ greedy w.r.t. $\hat{q}(s, a, w)$ is no longer guaranteed to improve $\pi$ — no more policy improvement theorem.
Semi-Gradient Sarsa

Episodic Semi-gradient Sarsa for Estimating $\hat{q} \approx q_*$

Input: a differentiable action-value function parameterization $\hat{q} : S \times A \times \mathbb{R}^d \to \mathbb{R}$
Algorithm parameters: step size $\alpha > 0$, small $\varepsilon > 0$
Initialize value-function weights $w \in \mathbb{R}^d$ arbitrarily (e.g., $w = 0$)

Loop for each episode:
  $S, A \leftarrow$ initial state and action of episode (e.g., $\varepsilon$-greedy)
  Loop for each step of episode:
    Take action $A$, observe $R, S'$
    If $S'$ is terminal:
      $w \leftarrow w + \alpha \left[ R - \hat{q}(S, A, w) \right] \nabla \hat{q}(S, A, w)$
    Go to next episode
    Choose $A'$ as a function of $\hat{q}(S', \cdot, w)$ (e.g., $\varepsilon$-greedy)
    $w \leftarrow w + \alpha \left[ R + \gamma \hat{q}(S', A', w) - \hat{q}(S, A, w) \right] \nabla \hat{q}(S, A, w)$
    $S \leftarrow S'$
    $A \leftarrow A'$

Handle termination
Linear Function Approximation

• Assume value estimate is a linear function of state features.

\[ \hat{v}(s, w) = w^\top x(s) = \sum_{i=1}^{d} w_i x_i(s) \]

• The features, \( x_i(s) \), can be non-linear functions of state variables.

• Expressive choices for \( x(s) \) make linear methods more powerful than they first appear.
1-Hot Features / State Aggregation

• For a finite state-space, partition state-space into \(d\) mutually exclusive groups.

• Let \(i\) be the group to which state \(s\) belongs.

• The 1-Hot feature encoding sets \(x_i(s) = 1\) and \(x_j(s) = 0\) for \(j \neq i\).

• What does generalization look like?

• Special case is \(d = |\mathcal{S}|\) in which case we recover the tabular setting.
  • Useful tip for debugging RL implementations!
  • Easily switch between easy to understand tabular experiments and more complex function approximation within same implementation.
Polynomial Features

• Suppose the state is represented as \((s_1, s_2) \in \mathbb{R}^2\).

• Polynomial representation: \((1, s_1, s_2, s_1s_2, s_1^2, s_2^2, \ldots)\).

• What is the advantage of the polynomial representation?
  
  • Can represent any function with sufficiently high order polynomials.
Fourier Features

Figure 9.3: One-dimensional Fourier cosine-basis features $x_i, i = 1, 2, 3, 4$, for approximating functions over the interval $[0, 1]$. After Konidaris et al. (2011).

$$x_i(s) = \cos(\pi s^T c^i)$$
Fourier Features
Coarse Coding

Narrow generalization

Broad generalization

Asymmetric generalization
Tile Coding

- Intuitively, multiple state aggregation mappings at the same time.
Radial Basis Functions

- Coarse coding with continuous features.

\[ x_i(s) = \exp \left( - \frac{||s - c_i||^2}{2\sigma_i^2} \right) \]
Neural Networks
Memory-Based Learning

- Non-parametric methods avoid need to fix a functional form for $\hat{v}(s, w)$.
- Instead, keep around all observed states and their value estimates.
- When need to compute $\hat{v}(s, w)$, find closest previously seen states to $s$ and use their value estimates.
- (+) Capacity grows with amount of data, focus approximation resources on states the agent is actually visiting.
- (-) Computationally expensive to find closest states, notion of “closest” is problem dependent.
Summary

• Approximate on-policy control with semi-gradient Sarsa parallels tabular Sarsa but we no longer have guaranteed policy improvement.

• Linear function approximation can be powerful with the right choice of features.

• Many good options to choose from but the most practical might be to simply learn the features with a neural network.
Action Items

- Homework 3.
- Begin literature review.
- Begin reading Chapter 11.
- Midterm survey and evaluation.