Lecture 22, Mar 3, 2026

Reinforcement Learning – Introduction

- We start by identifying some issues with policy and value iteration:
  1. We need to know the model $f(x,u)$ to iterate
  2. Both are offline algorithms: for value iteration, we need V^* in the future so it needs to run backwards in time; for policy iteration, we need policy evaluation which is also offline
  3. V^*, V^μ are arbitrary nonlinear functions, introducing inherent complexity problems
  4. Performing the optimization for policy improvement is computationally difficult
- To address these issues, we shift to *reinforcement learning*, which is based on policy iteration
- We use the following innovations:
  1. *Temporal difference error*: replacing $f(x,u) \to x(k+1)$, which we can measure
     - This represents a fundamental shift where we go from an entirely offline state space search to an online method, since now we have a time dependence
  2. *Value function approximation*: assume that $V^\mu(x) = \psi^T w(x)$, where $w(x)$ is a regressor and ψ is an unknown parameter vector
     - e.g. in the LQR problem, $V(x) = x^T P x$

$$\begin{aligned} &= \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= p_1 x_1^2 + p_3 x_2^2 + 2 p_2 x_1 x_2 \\ &= \begin{bmatrix} p_1 & 2p_2 & p_3 \end{bmatrix} \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix} \\ &= \psi^T w(x) \end{aligned}$$

     - The known regressor represents the assumptions about what types of functions we can have in the value function; e.g. in deep learning, the regressor is generated by the neural network
     - Now we can do adaptive control on ψ
  3. Use *Q-functions* (*quality functions*) for policy improvement to make the optimization computationally tractable
     - We will use calculus here similar to what we did for LQR when we differentiated with respect to u

Q-Functions

- Consider a nonlinear system $x(k+1) = f(x(k), u(k))$ with a stationary policy $\mu \in \mathcal{M}$
- Let the Q-function or quality function be $Q^\mu(x,u) = r(x,u) + \gamma V^\mu(f(x,u)), x \in \mathcal{X}, u \in \mathcal{U}(x)$
  - Consider the Bellman equation for V^μ: $V^\mu(x) = r(x, \mu(x)) + \gamma V^\mu(f(x, \mu(x)))$
    $= Q^\mu(x, \mu(x))$
  - Intuitively the Q-function is similar to V^μ, but we can choose the input that we apply in the current step, rather than being locked into the stationary policy μ; it contains more information than V^μ
  - Therefore the Q-function for μ also satisfies its own Bellman equation $Q^\mu(x,u) = r(x,u) + \gamma Q^\mu(f(x,u), \mu(f(x,u)))$
- Let the optimal Q-function $Q^*(x,u) = r(x,u) + \gamma V^*(f(x,u))$
  - Note that $V^*(x) = \min_{u \in \mathcal{U}(x)} Q^*(x,u)$ and $\mu^*(x) = \arg\min_{u \in \mathcal{U}(x)} Q^*(x,u)$
    * The first can be obtained by applying $\inf_{u \in \mathcal{U}(x)}$ to the definition for Q^*
  - Substituting $V^*(x)$, we get the HJB equation is $Q^*(x,u) = r(x,u) + \gamma \min_{u' \in \mathcal{U}(f(x,u))} Q^*(f(x,u), u')$
- Our main objective is to do the iterations online, but we can still do our offline searches with the Q-function:
  - Policy iteration with Q-functions:

1. Initialize: select any $\mu^0 \in \mathcal{M}$
2. Policy evaluation: $Q^{\mu^j}(x,u) = r(x,u) + \gamma Q^{\mu^j}(f(x,u), \mu^j(f(x,u))), \forall x \in \mathcal{X}, u \in \mathcal{U}(x)$
   * Note that compared to before now we need to do more work since we also need to evaluate over all values of u
3. Policy improvement: $\mu^{j+1}(x) = \arg\min_{u \in \mathcal{U}(x)} Q^{\mu^j}(x,u)$

– Value iteration with Q-functions:

1. Initialize: select any Q^0
2. Value iteration: $Q^{j+1}(x,u) = r(x,u) + \min_{u' \in \mathcal{U}(f(x,u))} \gamma Q^j(f(x,u), u')$