Lecture 23, Mar 5, 2026

Temporal Difference Learning and Value Function Approximation

- Consider again the system $x(k+1) = f(x(k), u(k))$ and some stationary policy $\mu \in \mathcal{M}$
- Define the temporal difference (TD) error $e(k) = r(x(k), \mu(x(k))) + \gamma V^\mu(x(k+1)) - V^\mu(x(k))$
  - This comes from the Bellman equation, but we introduce time
  - This can be interpreted as a prediction error: $e(k) = \hat{V}^\mu(x(k)) - V^\mu(x(k))$ where the "estimate" is generated based on the Bellman equation, $\hat{V}^\mu(x(k)) = r(x(k), \mu(x(k))) + \gamma V^\mu(x(k+1))$
  - Since V^μ satisfies its own Bellman equation, $e(k) = 0, \forall k$ along all solutions of the system
- The preliminary idea is that we set $r(x(k), \mu(x(k))) + \gamma V^\mu(x(k+1)) - V^\mu(x(k)) = 0$, and as we apply some policy we collect data about the system states, and in the end use a batch algorithm (e.g. least squares) to solve for V^μ
- To solve for V^μ we need value function approximation; consider the LQR problem with a stationary policy $u(x) = -Kx$, $V^\mu(x) = x^T P x$ and $r(x, u) = x^T Q x + u^T R u$
  - Substitute: $e(k) = x^T(k) Q x(k) + u^T(k) R u(k) + x^T(k+1) P x(k+1) - x^T(k) P x(k)$
    - Note that since we know everything except P, this can be easily solved since the equation is linear in P
  - We can use the Kronecker product to rewrite the TD error in a form amenable to batch least squares
    - $V^\mu(x) = x^T P x = (\text{vec}(P))^T (x \otimes x) = \psi^T w(x)$
    - Note $\text{vec}(P)$ stacks all entries of P into one long vector, and $x \otimes x = \begin{bmatrix} x_1 x \\ \vdots \\ x_n x \end{bmatrix} \in \mathbb{R}^{n^2}$
    - The Kronecker product will give us duplicate terms, which we can eliminate
- In general we approximate the value function as $V^\mu(x) = \psi^T w(x)$, then the TD error becomes $e(k) = r(x(k), \mu(x(k))) + \gamma \psi^T w(x(k+1)) - \psi^T w(x(k))$, which is linear in ψ