Lecture 24, Mar 6, 2026

Q-Learning – Policy Evaluation

- Recall the Bellman equation for the Q-function: $Q^\mu(x,u) = r(x,u) + \gamma Q^\mu(f(x,u), \mu(f(x,u)))$
  - Along a solution, this becomes $Q^\mu(x(k),u(k)) = r(x(k),u(k)) + \gamma Q^\mu(x(k+1), \mu(x(k+1)))$
    - It's important to note that we use $u(k)$ in Q and r for time k, but $\mu(x(k+1))$ for time $k+1$; for the current time step we are allowed to explore and apply any input we want (so $u(k) \neq \mu(x(k))$), but for all future time steps we must follow the policy
- The TD error is $e(k) = Q^\mu(x(k),u(k)) - \gamma Q^\mu(x(k+1), \mu(x(k+1))) - r(x(k),u(k))$, zero along solutions
- Introduce the Q-function approximation, $Q^\mu(x,u) = \psi^T w(x,u)$ for some known regressor and unknown (but constant) parameters
- Let $v(k) = w(x(k)) - \gamma w(x(k+1))$, a regressor which is fully known at time $k+1$, then using the TD error we can write $\psi^T v(k) = r(x(k),u(k))$, since $e(k) = 0$ along a solution
  - This allows us to estimate $\hat{Q}^\mu(x,u) = \hat{\psi}^T(k) w(x,u)$ for some parameter estimate $\hat{\psi}(k)$
  - Define the new error $e_1(k) = \hat{\psi}^T v(k) - r(x(k),u(k)) = \tilde{\psi}^T(k) v(k)$, where $\tilde{\psi}(k) = \hat{\psi}(k) - \psi$, giving us a static error model which we can solve using standard techniques
- Now we can from an algorithm for online, model-free policy iteration:
  1. Initialization: select any admissible policy $\mu^0 \in \mathcal{M}$
  2. Policy evaluation: compute ψ^{j+1} satisfying $(\psi^{j+1})^T (w(x(k)) - \gamma w(x(k+1))) = r(x(k), \mu^j(x(k)))$
     - Note we need to wait until time $k+1$ to know the value of $w(x(k+1))$
  3. Policy improvement: $\mu^{j+1}(x(k)) \in \underset{u \in \mathcal{U}(x(k))}{\arg\min} \{ r(x(k),u) + \gamma (\psi^{j+1})^T w(x(k+1)) \}$
     - Note that the k here is different, since to do policy evaluation we had to run the system forward in time
     - This is however not yet feasible since we need to have $\mu^{j+1}(x(k))$ for all $x(k) \in \mathcal{X}$, but since we must follow solutions in time it's not guaranteed that we reach all those states
     - These issues will be addressed next time
- During policy evaluation we must find $\psi^{j+1} \in \mathbb{R}^q$ such that $(\psi^{j+1})^T v(k) = r(x(k), \mu^j(x(k)))$
- We can do this via batch least squares: run the process for q steps, yielding q equations for $x(k), \ldots, x(k+q-1)$; now we can solve for the q unknowns in ψ with least squares
  - Can also be done using recursive least squares (not covered in this course)
- We can also use adaptive control like we did before
  - Let $\hat{\psi}^{j+1}(k)$ be an estimate of ψ^{j+1}
  - Let the prediction error
$$\begin{aligned} e_1(k) &= (\hat{\psi}^{j+1}(k))^T v(k) - r(x(k), \mu^j(x(k))) \\ &= (\hat{\psi}^{j+1}(k))^T v(k) - (\psi^{j+1})^T v(k) \\ &= (\tilde{\psi}^{j+1}(k))^T v(k) \end{aligned}$$
    - The first equation can be computed
    - This results in a static error model
  - Now we can use the typical gradient law $\hat{\psi}^{j+1}(k+1) = \hat{\psi}(k) - \gamma_1(k) e_1(k) v(k), \gamma_1(k) = \frac{\bar{\gamma}_1}{1 + \|v(k)\|^2}, \bar{\gamma}_1 \in (0,2)$
- We need to run the policy evaluation step long enough for $\hat{\psi}^{j+1}$ to converge before starting policy improvement
  - Practically we need a way to know when it's good enough to move on