IV. Example: LQR

Last time: (1) TD error (2) Value Fcn approx

We have established:

- $V(x_k) = x_k^T P x_k$ is quadratic
- $u_k = K \cdot x_k$ is linear

Construct TD error: $e_k = c + \overset{1}{\cancel{\gamma}} \cdot V_\pi(x_{k+1}) - V_\pi(x_k)$

$$e_k = x_k^T Q x_k + u_k^T R u_k + x_{k+1}^T \underset{?}{P} x_{k+1} - x_k^T \underset{?}{P} x_k$$

Note this is linear in parameter matrix P.

Next, let's re-write $V(x_k) = x_k^T P x_k = \underbrace{W}_{\text{weights}}^T \overset{\text{basis vec}}{\phi(x)}$

Define $W = \mathrm{vec}(P)$ ← stacks columns of P into one big column vector

$\phi(x) = x_k \otimes x_k$ ← is a vector of monomials of x_k.

For example:

$$x_k = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad P = \begin{bmatrix} p_{11} & p_{12} \\ * & p_{22} \end{bmatrix}$$

then

$$x_k^T P x_k = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ * & p_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$= \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} p_{11} x_1 + p_{12} x_2 \\ p_{12} x_1 + p_{22} x_2 \end{bmatrix}$$

$$V = \underline{p_{11}} x_1^2 + \underline{p_{12}} x_2 x_1 + \underline{p_{12}} x_1 x_2 + \underline{p_{22}} x_2^2$$

$$= \underbrace{\begin{bmatrix} p_{11} & 2p_{12} & p_{22} \end{bmatrix}}_{= W^T} \underbrace{\begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix}}_{= \phi(x)} = W^T \phi(x)$$

Note P is symmetric, we have

$$\phi(x) : \mathbb{R}^n \to \mathbb{R}^{n(n+1)/2}.$$

We can re-write the TD error as:

$$e_k = \underbrace{x_k^T Q x_k + u_k^T R u_k}_{c(x_k, u_k)} + W^T \phi(x_{k+1}) - W^T \phi(x_k)$$

$$= c(x_k, u_k) + W^T \left[\phi(x_{k+1}) - \phi(x_k) \right]$$

The TD error can be computed for supervised learning of $V_\pi(\cdot)$ by collecting $(x_k, x_{k+1}, c(x_k, u_k))$ at each time step

<u>Remark</u>: Previously, DP algos required eval. of Bellman eqn at <u>all</u> $x_k \in X$. To achieve this computationally, we considered a <u>discrete-value</u> state space X. This results in a exponential increase in calculations as the state vector size increases, known as "curse of dimensionality".

Value fcn approx bypasses the curse of dimensionality!