Course 3, Module 1
On-policy Prediction with Approximation

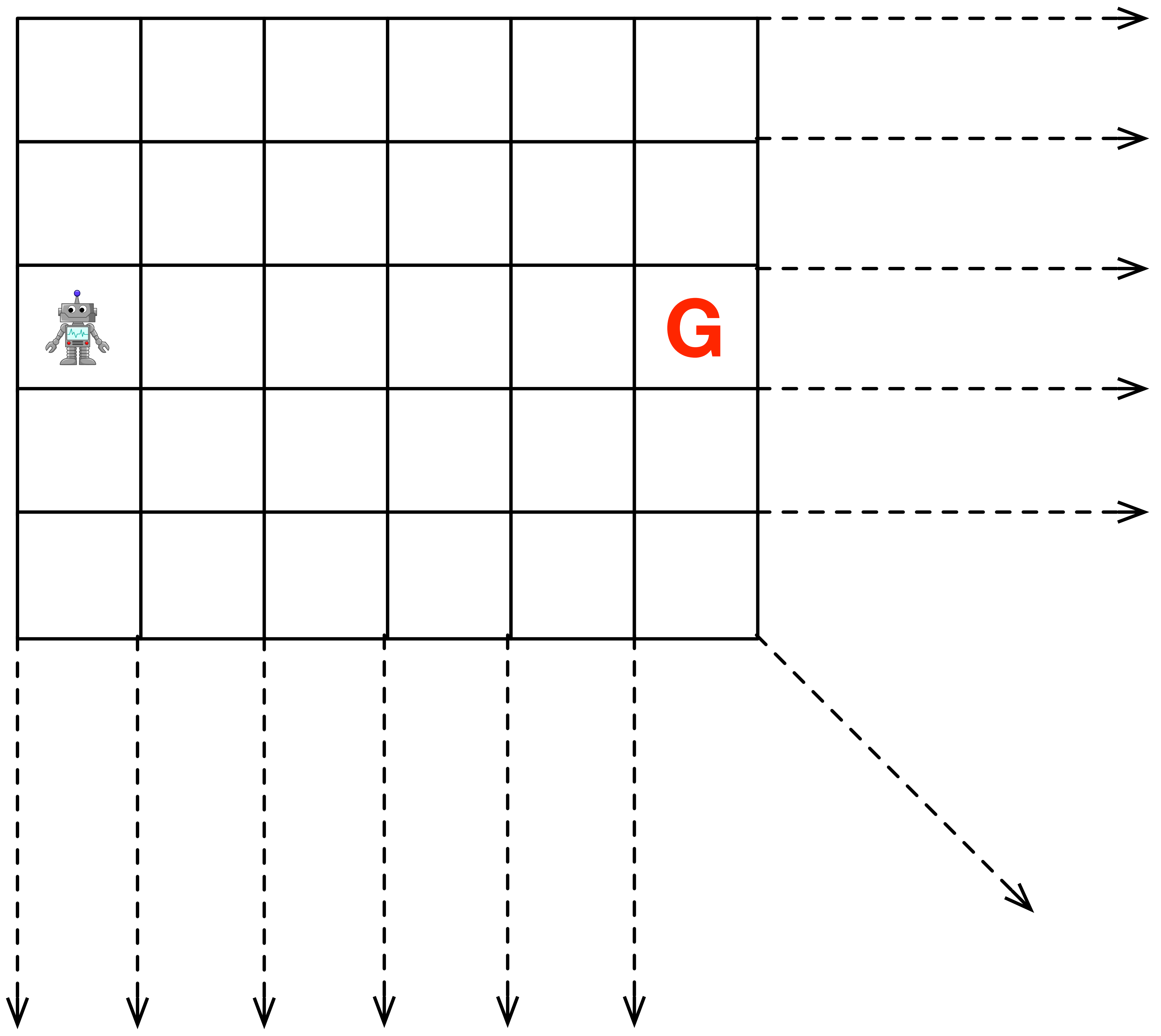
CMPUT 397
Fall 2019

Announcements

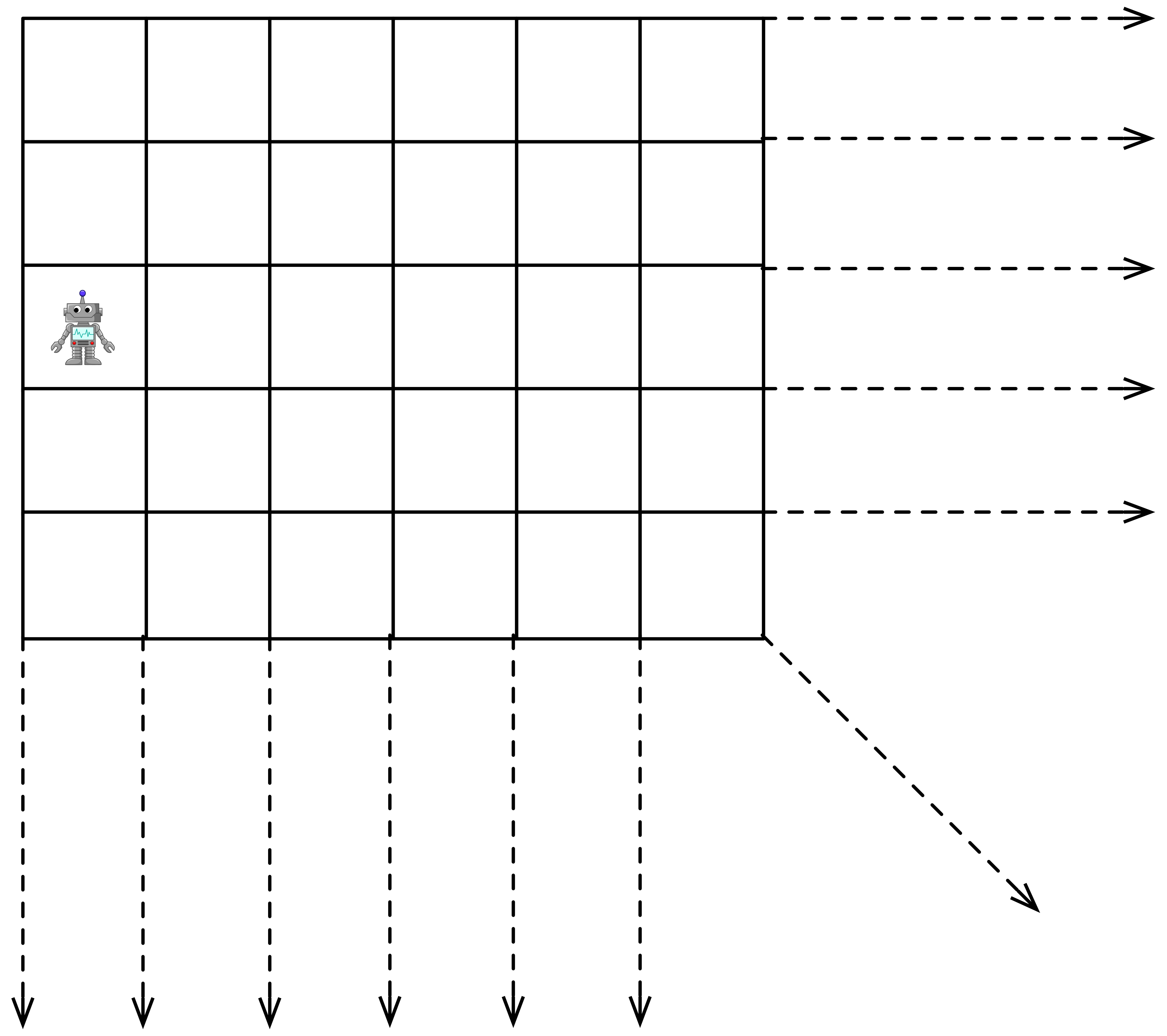
- If you **get zero on a participation mark** for one week there are two typical reasons:
  - you did not submit
  - you submitted a discussion topic that was not acceptable (major formatting & spelling problems, unclear, asked a question that was the topic of a video, asked for help etc)
  - the reasons will be noted in eclass
- Are you checking eclass? Do you get the announcements?

- Link for questions:
  - **http://www.tricider.com/brainstorming/3D4V06mUv2V**

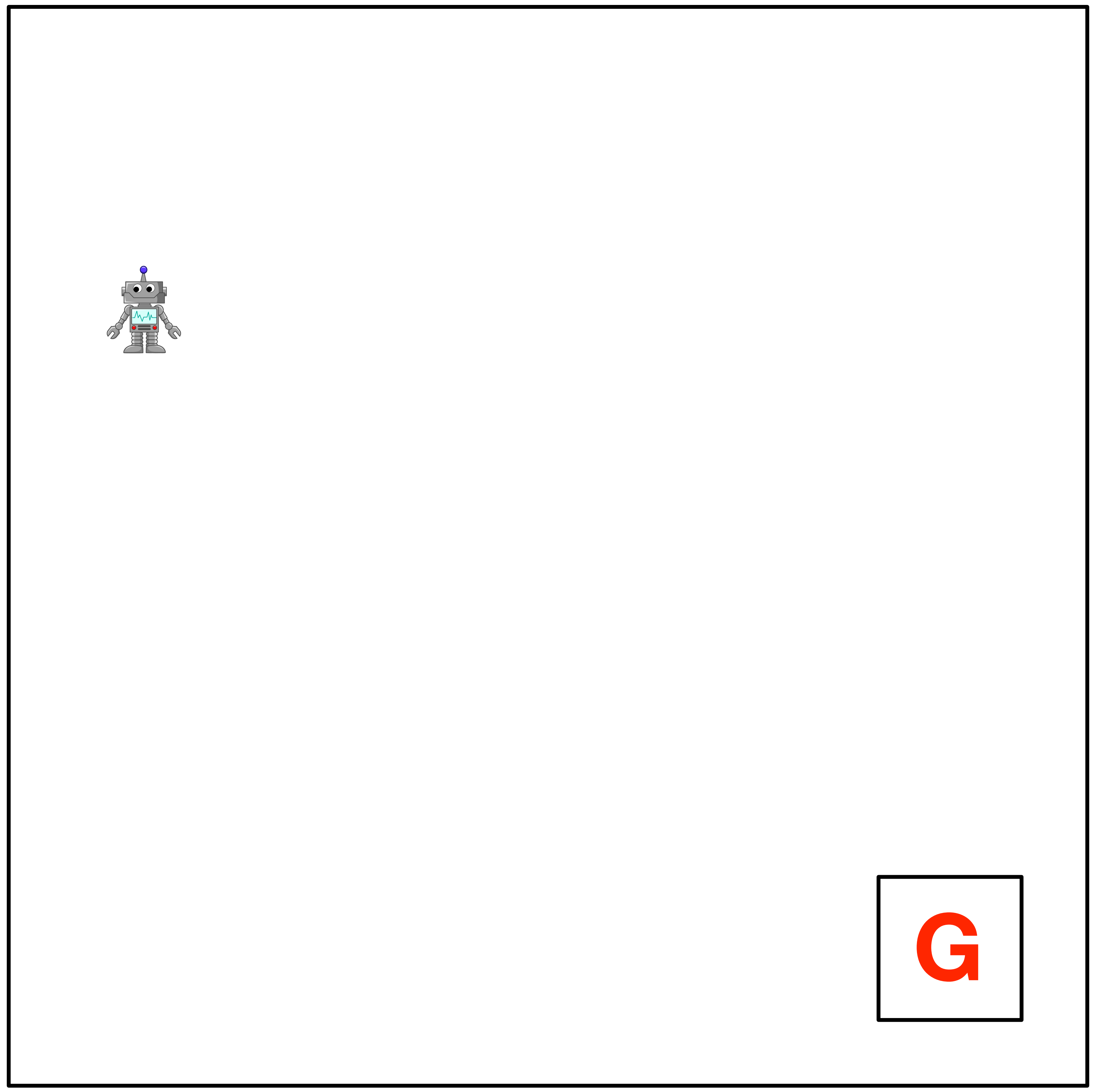
Imagine a huge state space



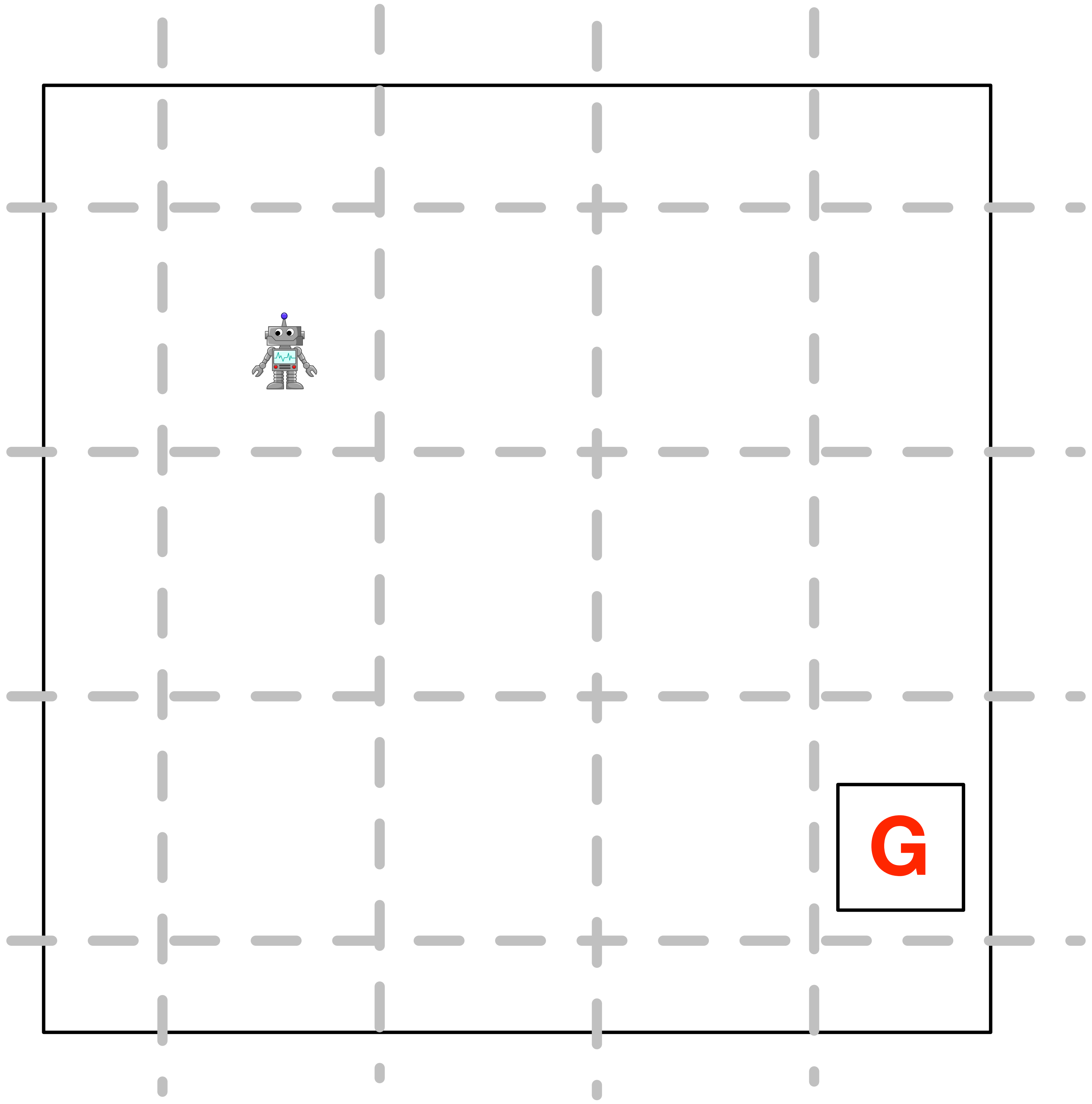
Imagine a huge state space



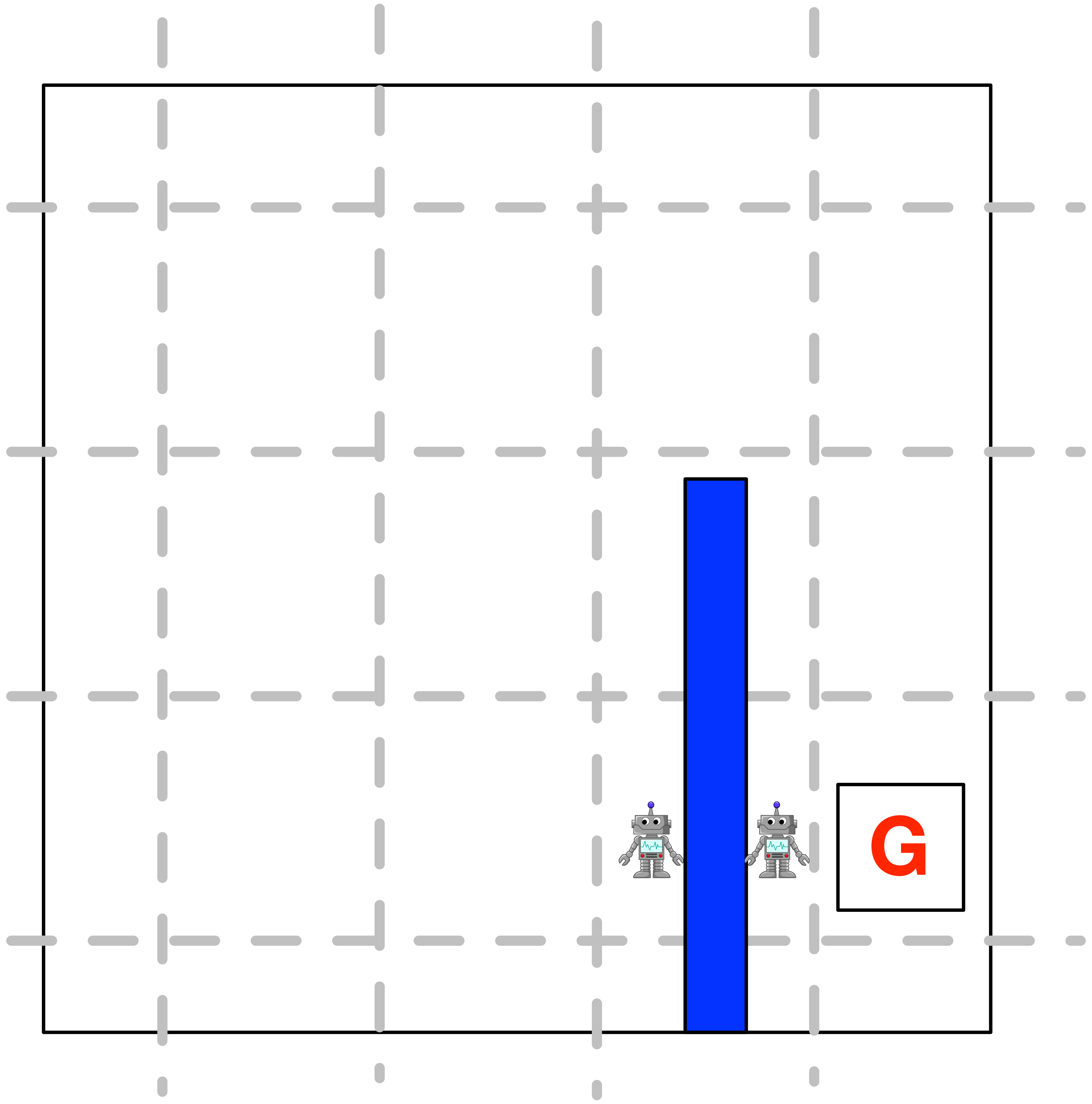
Imagine a continuous state space



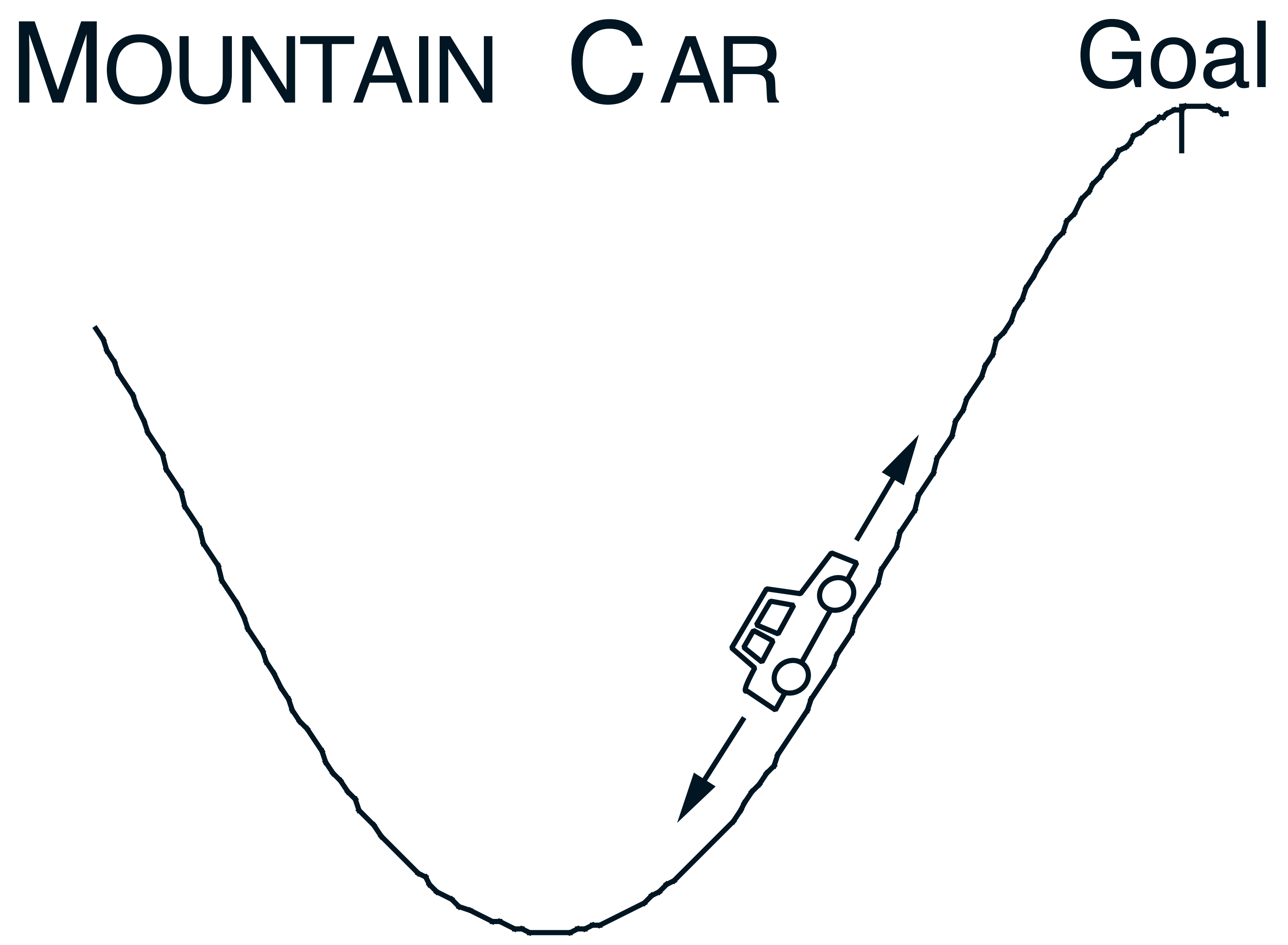
Imagine a continuous state space



Imagine a continuous state space



Another continuous state domain



$$p_{t+1} \doteq bound\big[p_t + \dot{p}_{t+1}\big]$$

$$\dot{p}_{t+1} \doteq bound\big[\dot{p}_t + 0.001A_t - 0.0025\cos(3p_t)\big]$$

Review of Course 3, Module 1
Prediction with Function Approximation

Video 1: Moving to Parameterized Functions

- **Using parameterized functions to represent value functions.** From tables of values to more general functions over states
- Goals:
  - Understand how we can use parameterized functions to approximate values.
  - Explain linear value function approximation.
  - Recognize that the tabular case is a special case of linear value function approximation
  - Understand that there are many ways to parameterize an approximate value function.

$$\cancel{V(s)} \approx v_\pi(s)$$ 1.71

$$\mathbf{w} \in \mathbb{R}^d, \quad e.g., \quad \mathbf{w} = \begin{bmatrix} 2.1 \\ 0.01 \\ -1.1 \\ 1.2 \\ -0.1 \\ 0.01 \\ 4.93 \\ 0.5 \end{bmatrix}, \qquad \mathbf{x}(s) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{x} : \mathcal{S} \to \mathbb{R}^d$$

parameter vector

feature vector

Video 2: Generalization and Discrimination

- A key concept in machine learning. We cannot learn all the values separately (in fact we wouldn't want to), so we have to make choices.
- Goals:
  - Understand what is meant by generalization and discrimination
  - Understand how generalization can be beneficial
  - Explain why we want both generalization and discrimination from our function approximation

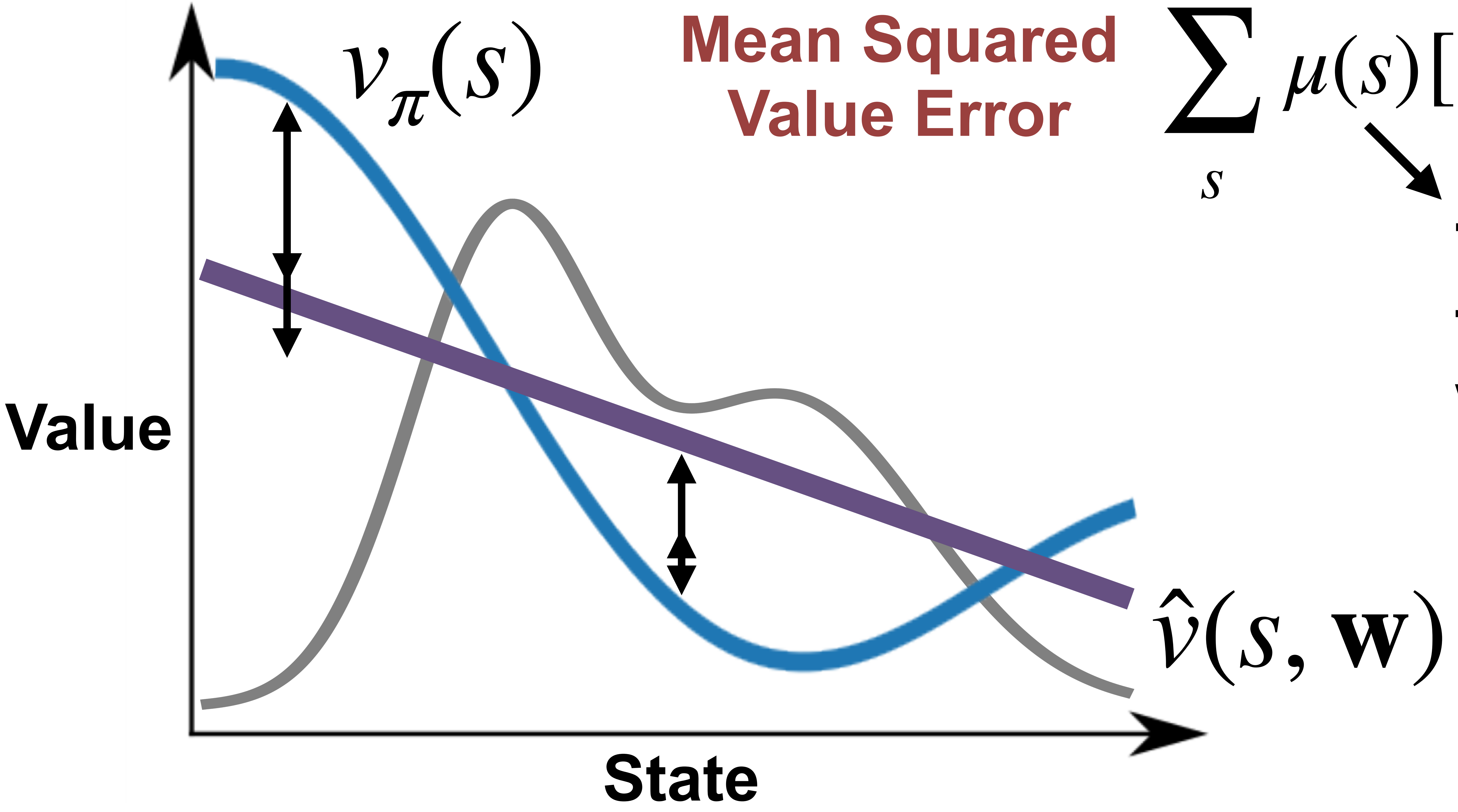
Video 3: Framing Value Estimation as Supervised Learning

- If we can setup the problem of learning a value function (policy evaluation) as a supervised learning problem, then we can borrow methods from supervised learning to do reinforcement learning with function approximation.
- Goals:
  - Understand how value estimation can be framed as a supervised learning problem
  - Recognize that not all function approximation methods are well suited for reinforcement learning.

Video 4: Value Error

- We want to change the parameters of our function to estimate the value. We need an objective function!
- Goals:
  - Understand the mean-squared value error objective for policy evaluation
  - Explain the role of the state distribution in the objective

The Mean Squared Value Error Objective



Mean Squared Value Error

$$\sum_s \mu(s)[v_\pi(s) - \hat{v}(s, \mathbf{w})]^2$$

The fraction of time we spend in S when following policy π

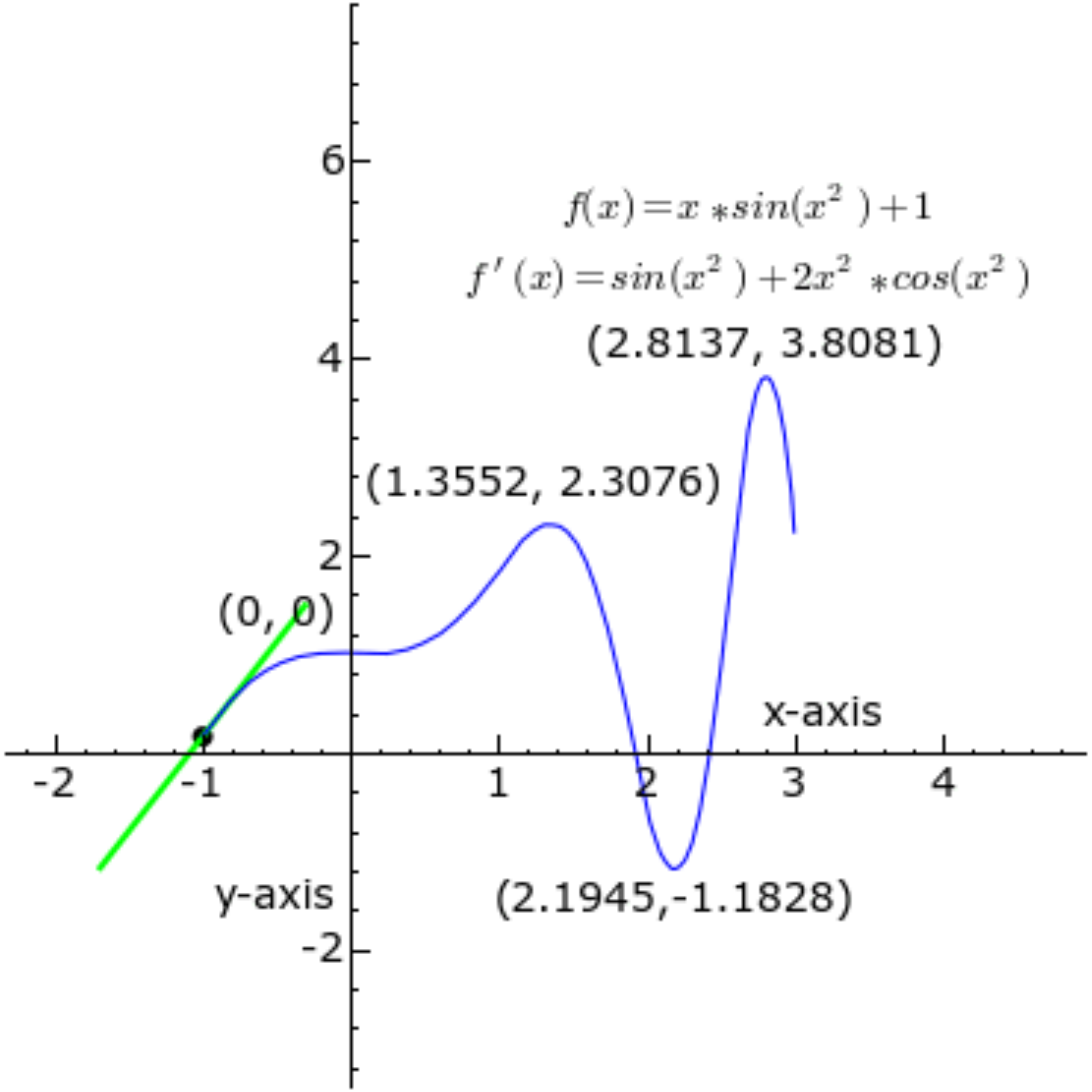
Question: Why didn't we use the Value Error in the tabular setting?

Video 5: Introducing Gradient Descent

- An algorithm for adapting the parameters of our estimate of the value function.
- Goals:
  - Understand the idea of gradient descent
  - Understand that gradient descent converges to stationary points

Question

- Why do we care about finding stationary points? i.e., point w where the gradient is zero



Video 6: Gradient Monte Carlo for Policy Evaluation

- We use gradient descent idea to get an online algorithm to adjust the parameters of our value function estimate
- Goals:
  - Understand how to use gradient descent and stochastic gradient descent to minimize value error
  - Outline the gradient Monte Carlo algorithm for value estimation

Video 7: State Aggregation with Monte Carlo

- So far we have said the value function could be any parametric function. Here we use a particular one---state aggregation. Simple and effective. And we run an experiment on a big Random Walk Problem

- Goals:

  - Understand how state aggregation can be used to approximate the value function

  - Apply Gradient Monte-Carlo with state aggregation

Video 8: Semi-gradient TD for Policy Evaluation

- TD with function approximation. Now we can learn value functions, in continuous state spaces AND update the value function parameters on every time-step!!
- Goals:
    - Understand the TD-update for function approximation
    - Outline the Semi-gradient TD algorithm for value estimation.

Semi-gradient TD(0) for estimating $\hat{v} \approx v_\pi$

Input: the policy π to be evaluated
Input: a differentiable function $\hat{v} : \mathcal{S}^+ \times \mathbb{R}^d \rightarrow \mathbb{R}$ such that $\hat{v}(\text{terminal},\cdot) = 0$
Algorithm parameter: step size $\alpha > 0$
Initialize value-function weights $\mathbf{w} \in \mathbb{R}^d$ arbitrarily (e.g., $\mathbf{w} = \mathbf{0}$)

Loop for each episode:
    Initialize S
    Loop for each step of episode:
        Choose $A \sim \pi(\cdot|S)$
        Take action A, observe R, S'
        $\mathbf{w} \leftarrow \mathbf{w} + \alpha\big[R + \gamma\hat{v}(S',\mathbf{w}) - \hat{v}(S,\mathbf{w})\big]\nabla\hat{v}(S,\mathbf{w})$
        $S \leftarrow S'$
    until S is terminal

Question: What is different compared to Tabular TD(0)?

Video 9: Comparing TD and MC with State Aggregation

- An experiment comparing TD and MC with a simple function approximation.
- Goals:
  - Understand that TD converges to biased value estimates
  - Understand that TD can learn faster than Gradient Monte Carlo.

Video 10: The Linear TD Algorithm

- Linear function functions are special. Most of the theory in RL is for the case of linear function approximation. The algorithms can work well, if we have good features.
- Goals:
  - Derive the TD-update with linear function approximation
  - Understand that tabular TD is a special case of linear semi-gradient TD
  - Understand why we care about linear TD as a special case.

Video 11: The True Objective for TD

- A bit of theory about TD with function approximation. What does the algorithm converge to?
- Goals:
  - Understand the fixed point of linear TD
  - Describe a theoretical guarantee on the mean squared value error at the TD fixed point

Terminology

- We will do this on Wednesday

Any questions about the practice quiz?

Exercise Questions

$$\min_{\mathbf{w}\in\mathbb{R}^d} \sum_s \mu(s)[v_\pi(s) - \hat{v}(s, \mathbf{w})]^2$$

- Why can't we directly optimize the MSVE? We know the stochastic gradient descent update would be the following

$$\mathbf{w}_t + \alpha[v_\pi(S_t) - \hat{v}(S_t, \mathbf{w})]\,\nabla\,\hat{v}(S_t, \mathbf{w})$$

- Further, why doesn't the TD fixed point minimize the MSVE?