CS 4/5789 7 Feb 2022 Prof Sarah Dean

Lecture 5: Continuous State-Space

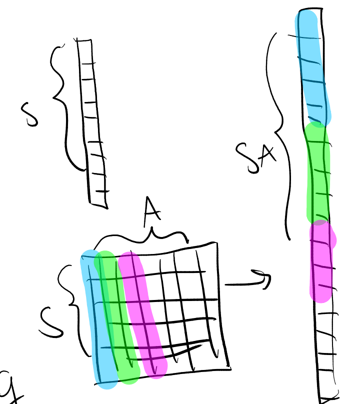
1) State distribution & transition matrix

Before we move onto continuous state spaces, let's reveiw how we've made use of discreteness so far.

A) Functions can be represented as finite dimensional vectors (or arrays)

e.g. the value function $V^\pi(s)$ $\forall s$ can be written $V^\pi \in \mathbb{R}^S$.

the Q function $Q^\pi(s,a)$ $\forall s,a$ can be written $Q^\pi \in \mathbb{R}^{SA}$.

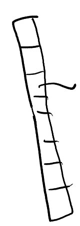


sometimes the finite state space setting is referred to as the "tabular setting"

B) Probability distributions can be represented as finite dimensional vectors

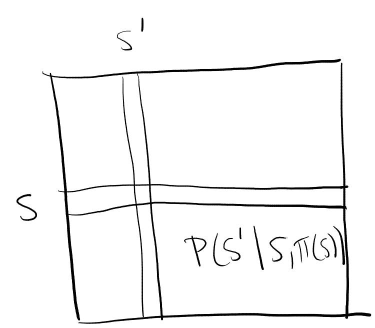
e.g. any distribution over states $\Delta(\mathcal{S})$ can be written as an S-dimensional vector

$d \in \mathbb{R}^S$



c) Transition Probabilities can be written as matrices

e.g. $P^\pi \in \mathbb{R}^{S\times S}$



D) Expectations can be written as dot products or matrix-vector multiplication

e.g. in policy evaluation, we wrote

$$\mathbb{E}_{s' \sim P(s,\pi(s))}(V^\pi(s')) = \sum_{s' \in \mathcal{S}} P(s' | s, \pi(s)) V^\pi(s')$$

$$= \langle P^\pi_s, V^\pi \rangle$$

stacking so that $\mathbb{E}(V^\pi) \in \mathbb{R}^S$, $s \to \mathbb{E}_{s' \sim P(s)}[V(s')]$

$$\mathbb{E}[V^\pi] = P^\pi V^\pi$$

One fact that we haven't made use of is that for a fixed policy, state distributions update according to repeated multiplication by the transition matrix.

suppose $s_0 \sim \mu_0$ and let $d_0 \in \mathbb{R}^S$ be the vector representation of μ_0

under policy π, define transition matrix P^π as above.

Then $d_1 = (P^\pi)^T d_0$ represents the state distribution at $t=1$.

$$\mathbb{P}^\pi_1(s ; \mu_0) = \sum_{s' \in \mathcal{S}} P(s | s', \pi(s')) \mu_0(s')$$

Similarly, $d_k = ((P^\pi)^k)^T d_0$ represents the state distribution at $t=k$.

Powers of matrix P^π determine the trajectory (HW1)

2) Continuous Control

Motivated by applications where states, actions are real-valued, we now consider MDPs with continuous state and action spaces. The historical terminology for this setting is an "optimal control problem" as continuous state/action spaces are considered when designing controllers for many physical systems.

<u>Setting</u>: Finite horizon optimal control

$$\mathcal{M} = \{\mathcal{S}, \mathcal{A}, (f, \mathcal{D}), c, H, \mu_0\}$$

<u>States & actions</u>

We consider state space $\mathcal{S} = \mathbb{R}^{n_s}$ and action space $\mathcal{A} = \mathbb{R}^{n_a}$ so states and actions are realvalued vectors.

Classically in control, states are represented by x & actions by u, and actions are called "input!" - But we stick with s, a.

<u>Dynamics</u>

Instead of representing transitions between states as a probability distribution, we represent with a <u>dynamics function</u> $f: \mathcal{S} \times \mathcal{A} \times \mathcal{W} \to \mathcal{S}$

$$s_{t+1} = f(s_t, a_t, w_t) \iff s_{t+1} \sim P(\cdot \mid s_t, a_t)$$

$$w_t \sim \mathcal{D} \in \Delta(\mathcal{W})$$

The "disturbance" w_t encodes all of the randomness in the transitions.

It is possible to consider time-varying dynamics f_t in the finite horizon case.

<u>Cost</u>

Control engineers historically think in terms of costs (to be minimized) rather than rewards (to be maximized). We can always define cost as negative reward and vice versa.

In finite horizon problems we allow cost to be time-varying

$$c = (c_0, c_1, \ldots, c_{H-1}, c_H)$$

final state cost

$$c_t : \mathcal{S} \times \mathcal{A} \to \mathbb{R} \qquad c_H : \mathcal{S} \to \mathbb{R}$$

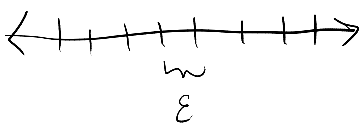
Optimal Control Problem (finite horizon)

$$\min_{\pi} \mathbb{E}_{w_t}\left[\sum_{t=0}^{H-1} c_t(s_t, a_t) + c_H(s_H) \right]$$

$$\begin{aligned} s_{t+1} &= f(s_t, a_t, w_t) \\ a_t &= \pi_t(s_t) \\ s_0 &\sim \mu_0 \end{aligned}$$

equivalent to finite horizon MDP but we use slightly different notation and conventions (described above)

Discretization? Notice that we could always divide up the real line into ε-sized peices.



as long as states/actions are bounded* this would be approximately equivalent to a discrete space.

But how many states/actions would we need?

$$O(1/\varepsilon)^{n_s} \ \& \ O(1/\varepsilon)^{n_a}$$

Exponential dependence is no good!

*we return to question of boundedness when we discuss stability

Therefore, we will work directly with continuous variables and functions.

In finite state/action spaces, arbitrary functions can be described with a finite input/output table. Not so with infinite spaces. Often, we turn to a finite dimensional parametric description, $f_\theta(x)$ with $\theta \in \mathbb{R}^d$. Eg $f_\theta(x) = \theta^T x$ linear

3) Linear Dynamics

We will first focus on a special case:

$$s_{t+1} = A s_t + B a_t + w_t$$

$A \in \mathbb{R}^{n_s \times n_s}$ and $B \in \mathbb{R}^{n_s \times n_a}$ are the dynamics matrices. A describes how the state evolves without any action and B describes the effects of the action.

The disturbance $w_t \in \mathbb{R}^{n_s}$ and we often have $w_t \sim \mathcal{N}(0, \sigma^2 I)$ Gaussian

<u>Example</u>: Robot moving in 1D by choosing to apply force to right (positive) or left (negative).

Newton's 2nd law says "force = mass x accel"

So in other words acceleration $= \frac{a_t}{m}$

Using a discretization,

$$\text{accel} = \frac{v_{t+1} - v_t}{dt} = \frac{a_t}{m}$$

where v_t is velocity.

Similarly,

$$\frac{p_{t+1} - p_t}{d_t} = v_t$$

p_t is position

$$s_t = \begin{bmatrix} p_t \\ v_t \end{bmatrix} \qquad s_{t+1} = \begin{bmatrix} dt & 1 \\ & dt \end{bmatrix} s_t + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} a_t$$

Trajectory

The trajectory of a linear system given actions $(a_0, \dots, a_{t-1})$ and disturbances $(w_0, \dots, w_{t-1})$ is a linear function

$$s_t = A^t s_0 + \sum_{k=0}^{t-1} A^k (B a_{t-k-1} + w_{t-k-1})$$

Proof by induction (HW1)

This means that if $\mathbb{E}[w_{t-k-1}] = 0$,

$$\mathbb{E}[s_t | s_0, a_0, \dots, a_{t-1}] = A^t s_0 + \sum_{k=0}^{t-1} A^k B a_{t-k-1}$$

under a linear policy

$$a_t = K s_t$$

the closed loop trajectory is

$$s_t = (A+BK)^t s_0 + \sum_{k=0}^{t-1} (A+BK)^k w_{t-k-1}$$

or $\mathbb{E}(s_t | s_0, a_t = K s_t) = (A+BK)^t s_0$

4) Stability

How do we ensure that states and actions remain bounded? (for both practical — if my robot arm moves infinitely fast it might break — and analytical reasons)

Eg. — with $\pi(s) = Ks$ and $w_t = 0$, $s_t = (A+BK)^t s_0$

$a_t = K(A+BK)^t s_0$

What about for arbitrarily large horizons? $t \to \infty$?

In general this question is studied by "dynamical systems theory."

For linear systems, we care about three regimes, which we define by considering behavior when $a_t = 0$ and $w_t = 0 \ \forall t$

1) Stable: $s_t \to 0 \quad \forall s_0$

2) unstable: $\|s_t\|_2 \to \infty \quad \forall s_0$

3) marginally stable: when neither stable or unstable.

Spectral Radius

Define $\rho(A) = \max_i |\lambda_i(A)|$, the largest-magnitude eigenvalue.

Theorem (linear system stability):

The dynamics $s_{t+1} = A s_t$ are

1) stable if $\rho(A) < 1$
2) unstable if $\rho(A) > 1$
3) marginally stable if $\rho(A) = 1$

Proof: We consider diagonalizable matrices with real-valued eigenvalues (vs. complex) though the theorem holds more generally.

Thus $A = VDV^{-1}$ where $D = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_{n_s} \end{bmatrix}$

Change of variables: $\tilde{s} = V^{-1} s$.

$$s_{t+1} = A s_t \iff \tilde{s}_{t+1} = D \tilde{s}_t$$

Then $\tilde{s}_t = D^t \tilde{s}_0$ and $D^t = \begin{bmatrix} \lambda_1^t & & \\ & \ddots & \\ & & \lambda_{n_s}^t \end{bmatrix}$

each element $\tilde{s}_t^i = \lambda_i^t \tilde{s}_0^i$ for $i = 1, \dots, n_s$.

<u>case</u> $\rho(A) < 1$

This means that $|\lambda_i| < 1 \quad \forall i$.

Thus, $|\tilde{s}_t^i| = |\lambda_i|^t |\tilde{s}_0^i| \to 0$ as $t \to \infty$.

$s_t = V\tilde{s}_t$ with V invertible, $s_t \to 0$ as well.

<u>case</u> $\rho(A) > 1$.

Thus $\exists\ i$ such that $|\lambda_i| > 1$.

$$\|\tilde{s}_t\|_\infty \geq |\tilde{s}_t^i| = |\lambda_i|^t |\tilde{s}_0^i| \to \infty \text{ as } t \to \infty$$

$s_t = V\tilde{s}_t$ so $\|s_t\|_\infty \to \infty$ as well.

<u>case</u> $\rho(A) = 1$.

first, consider all i with $|\lambda_i| < 1$.
By previous argument, $\tilde{s}_t^i \to 0$ as $t \to \infty$.

Then, for all i with $|\lambda_i| = 1$,

$\tilde{s}_t^i = \underbrace{\text{sign}(\lambda_i)}_{\pm 1}{}^t \tilde{s}_0^i$. Thus $|\tilde{s}_t^i| = |\tilde{s}_0^i|$

This is finite but nonzero.

$s_t = V\tilde{s}_t$ will also be finite but nonzero.

In the proof, we use an interesting fact that can let us visualize trajectories of

$$s_{t+1} = A s_t \,, \qquad A = VDV^{-1}$$

change of basis by V

$$\tilde{s}_t = \begin{bmatrix} \lambda_1^t & & \\ & \ddots & \\ & & \lambda_{n_s}^t \end{bmatrix} \tilde{s}_0$$

λ_i^t controls decay/growth along each axis

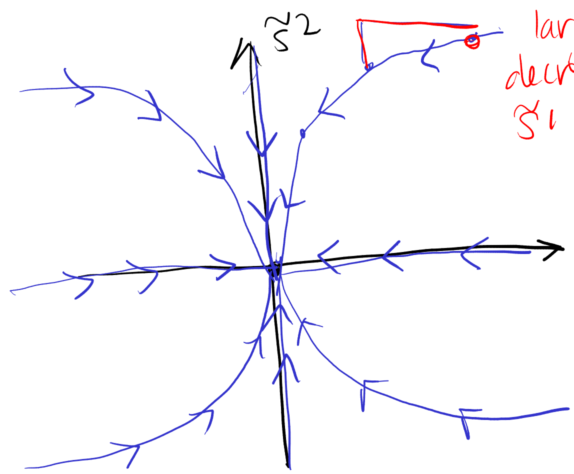
definition of eigenvector

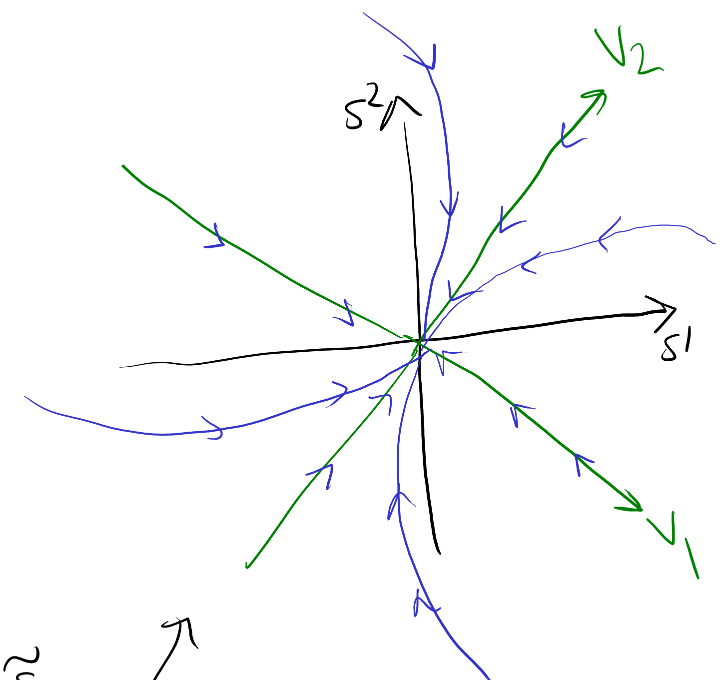
$$A v_i = \lambda_i v_i$$

if $s_0 \propto v_i$, then $s_t \propto v_i$

λ_i^t controls decay/growth along each eigenvector

2D example $\lambda_1 > \lambda_2 > 0$:





$$s = V\tilde{s}$$

$$\tilde{s} = V^{-1} s$$