

Continuous Space MDPs

Last Time

- Neural Network Function Approximation

Guiding Questions

- What tools do we have to solve MDPs with continuous S and A ?

Current Tool-Belt

Today: Four Tools

Notation: Continuous Random Variables

Term	Definition	Bernoulli(0.5) Coinflip Example	$\mathcal{U}([0, 1])$ Uniform Example
$\text{support}(X)$ $x \in X$	All the values that X can take	$\{h, t\}$ or $\{0, 1\}$	$[0, 1]$
Distribution • Discrete: PMF • Continuous: PDF	Maps each value in the support to a real number indicating its probability	$P(X = 1) = 0.5$ $P(X = 0) = 0.5$ $P(X)$ is a table	$p(x) = \mathbf{1}_{[0,1]}(x) = \begin{cases} 1 & \text{if } x \in [0, 1] \\ 0 & \text{o.w.} \end{cases}$  $P(X \in [a, b]) = \int_a^b p(x) dx$ $P(X = 0.5) = 0$
Expectation $E[X]$	First moment of the random variable, "mean"	$E[X] = \sum_{x \in X} x P(x)$ $= 0.5$	$E[X] = \int_{x \in X} x p(x) dx$ $= 0.5$

Rules for Continuous RVs

Discrete

- 1) a) $0 \leq P(X | Y) \leq 1$
- b) $\sum_{x \in X} P(x | Y) = 1$
- 2) $P(X) = \sum_{y \in Y} P(X, y)$
- 3) $P(X | Y) = \frac{P(X, Y)}{P(Y)}$
$$P(X, Y) = P(X | Y) P(Y)$$

Continuous

- 1) $0 \leq p(X | Y)$
$$\int_X p(x | Y) dx = 1$$
- 2) $p(X) = \int_Y p(X, y) dy$
- 3) $p(X | Y) = \frac{p(X, Y)}{p(Y)}$
$$p(X, Y) = p(X | Y) p(Y)$$

Multivariate Gaussian Distribution

Joint Distribution

Conditional Distribution

Marginal Distribution

Continuous S and A

e.g. $S \subseteq \mathbb{R}^n$, $A \subseteq \mathbb{R}^m$

The old rules still work!

1. Linear Dynamics, Quadratic Reward

$$S = \mathbb{R}^n, A = \mathbb{R}^m \quad T(s' | s, a) = \mathcal{N}(T_s s + T_a a, \Sigma) \quad R(s, a) = s^\top R_s s + a^\top R_a a$$
$$s' = T_s s + T_a a + w \quad w \sim \mathcal{N}(0, \Sigma) \quad (\text{Also works with other zero-mean } w.)$$

Finite Horizon: $U_h^*(s) = \max_{\pi} \left[\sum_{t=0}^h R(s_t, a_t) \right]$ π_h^* is "optimal h-step policy"

We will show that $U_h^*(s) = s^\top V_h s + q_h$ and $\pi_h^*(s) = -K_h s$

1. Linear Dynamics, Quadratic Reward

We will show that $U_h^*(s) = s^\top V_h s + q_h$ and $\pi_h^*(s) = -K_h s$ by induction.

Base: $U_1^*(s) = \max_a (s^\top R_s s + a^\top R_a a) = s^\top R_s s$

Inductive step: show that if $U_t^* = s^\top V_t s + q_t$, then $U_{t+1}^* = s^\top V_{t+1} s + q_{t+1}$.

$$\begin{aligned} U_{t+1}^*(s) &= \max_a (R(s, a) + \gamma E [U_t^*(s)]) \\ &= \max_a (s^\top R_s s + a^\top R_a a + \int p(w) U_t(T_s s + T_a a + w) dw) \\ &= s^\top R_s s + \max_a (a^\top R_a a + \int p(w) (T_s s + T_a a + w)^\top V_t (T_s s + T_a a + w) + q_t) dw \\ &= s^\top R_s s + s^\top T_s^\top V_t T_s s + \max_a (a^\top R_a a + 2s^\top T_s^\top V_t T_a a + a^\top T_a^\top V_t T_a a) + \int p(w) w^\top V_t w dw + q_t \end{aligned}$$

a^* is where $\nabla_a (\text{max term}) = 0$

$$0 = 2R_a a^* + 2T_a^\top V_t T_s s + 2T_a^\top V_t T_a a^*$$

$$a^* = -\underbrace{(R_a + T_a^\top V_t T_a)^{-1} T_a^\top V_t T_s s}_{K_t}$$

$$U_{t+1}^*(s) = s^\top \underbrace{\left(R_s + T_s^\top V_t T_s - (T_a^\top V_t T_s)^\top (R_a + T_a^\top V_t T_a)^{-1} (T_a^\top V_t T_s) \right)}_{V_{t+1}} s + \underbrace{\int p(w) w^\top V_t w dw + q_t}_{q_{t+1}}$$

$$U_{t+1}^*(s) = s^\top V_{t+1} s + q_{t+1} \quad \square$$

1. Linear Dynamics, Quadratic Reward

As $h \rightarrow \infty$

$$V_\infty = T_s^\top \left(V_\infty - V_\infty T_a \left(T_a^\top V_\infty T_a + R_a \right)^{-1} T_a^\top V_\infty \right) T_s + R_s$$

$$K_\infty = \left(T_a^\top V_\infty T_a + R_a \right)^{-1} T_a^\top V_\infty T_s \quad \pi_\infty^*(s) = -K_\infty s$$

(K_∞ has no dependence on Σ)

Certainty-Equivalence Principle: For Linear-Quadratic problems, the optimal policy with noise is the same as the optimal policy without noise!

Practical Implication: If a continuous problem has roughly linear dynamics, a convex cost function, and roughly zero-mean additive noise, you can use *certainty-equivalent control*, i.e. control as if there is no noise.

2. Value Function Approximation

$$V_\theta(s) = f_\theta(s) \quad (\text{e.g. neural network})$$

$$V_\theta(s) = \theta^\top \beta(s) \quad (\text{linear feature})$$

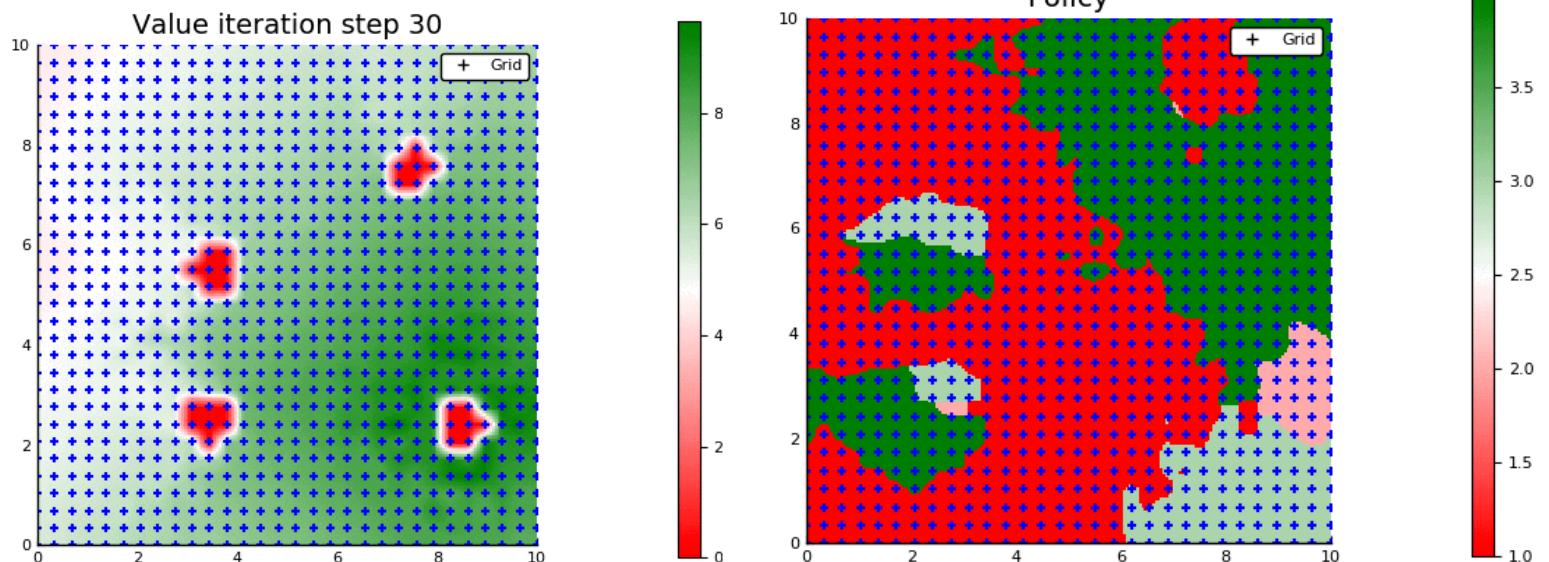
Fitted Value Iteration

while not converged

$$\theta \leftarrow \theta'$$

$$\hat{V}' \leftarrow B_{\text{approx}}[V_\theta]$$

$$\theta' \leftarrow \text{fit}(\hat{V}')$$



$$B_{\text{MC}(N)}[V_\theta](s) = \max_a \left(R(s, a) + \gamma \sum_{i=1}^N V_\theta(G(s, a, w_i)) \right)$$

Function Approximation

- Global: (e.g. Fourier, neural network)
- Local: (e.g. simplex interpolation)

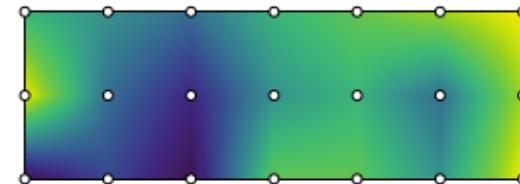
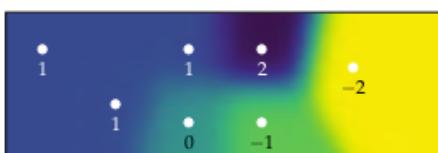
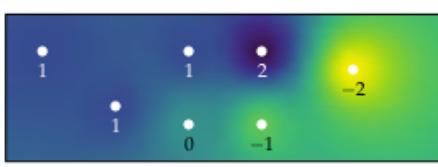
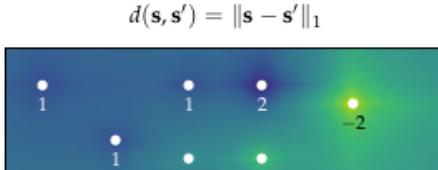
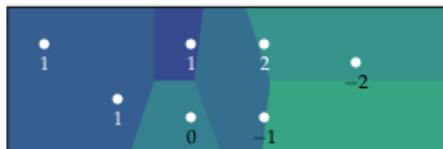
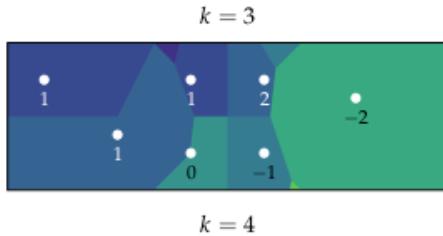
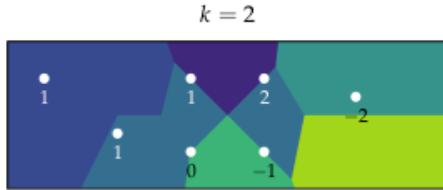
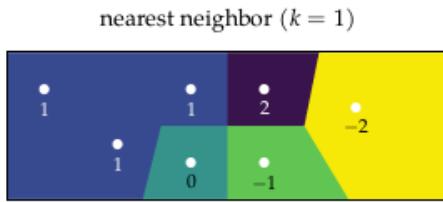


Figure 8.9. Two-dimensional linear interpolation over a 3×7 grid.

Weighting of 2^d points

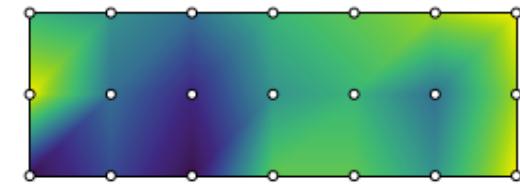
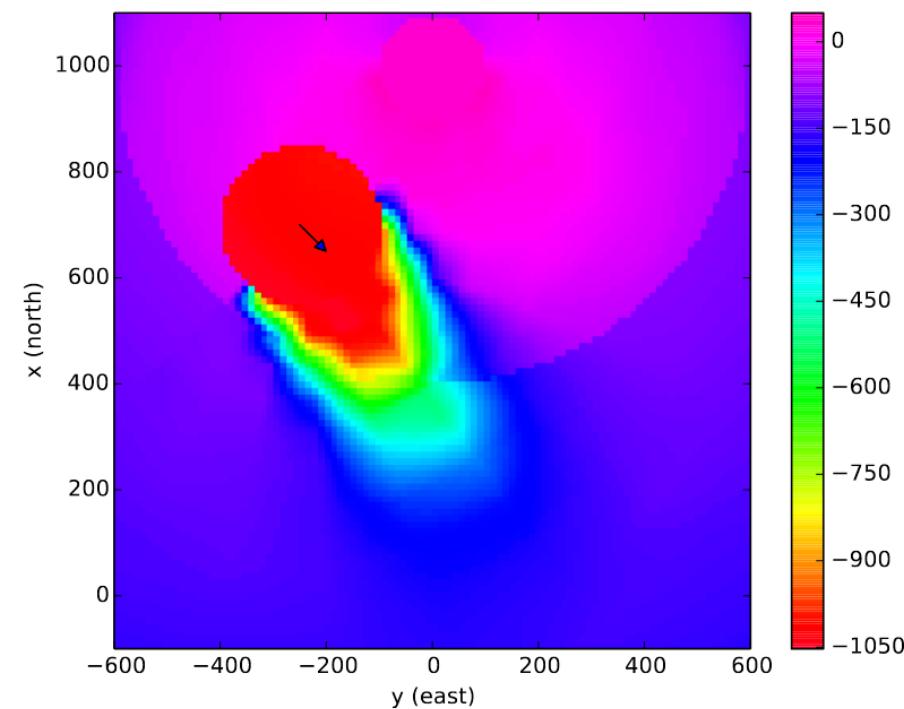
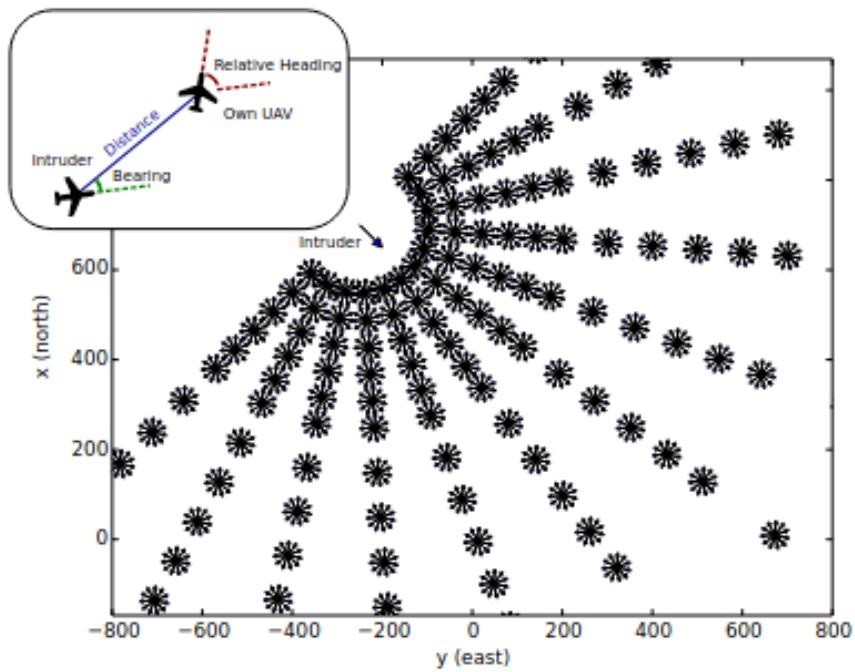


Figure 8.10. Two-dimensional simplex interpolation over a 3×7 grid.

Weighting of only $d + 1$ points!

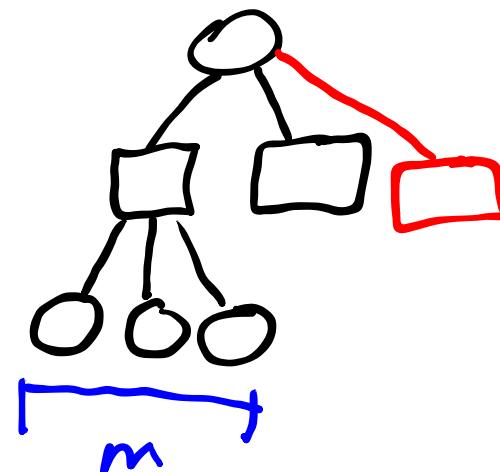
2. Value Function Approximation



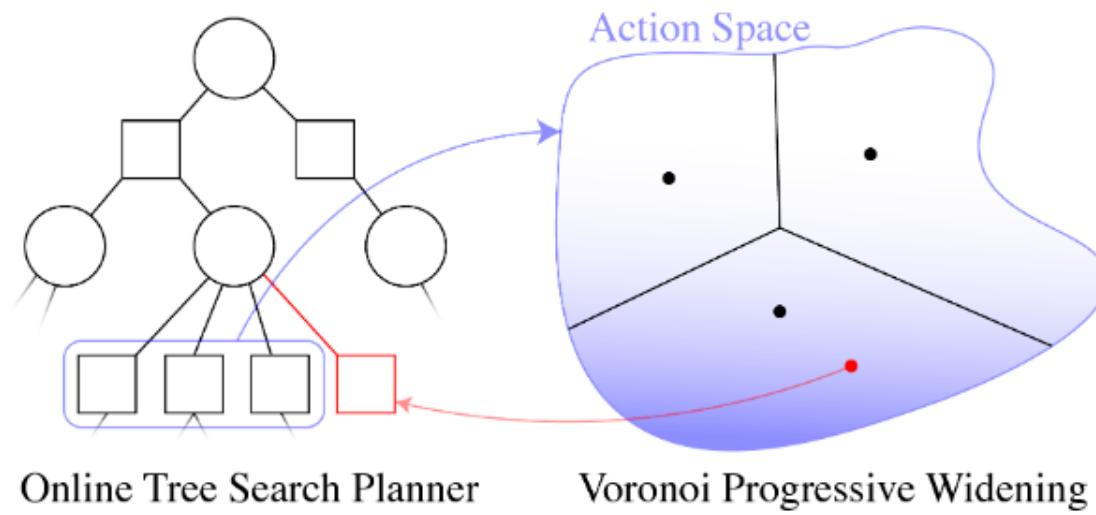
Break

What will a Monte Carlo Tree Search tree look like if run on a problem with continuous spaces?

3. Sparse Tree Search/Progressive Widening



add new branch if $C < kN^\alpha$ ($\alpha < 1$)



4. Model Predictive Control

(Use off-the-shelf optimization software, e.g. Ipopt)

Certainty-
Equivalent

$$\begin{aligned} & \underset{a_{1:d}, s_{1:d}}{\text{maximize}} \quad \sum_{t=1}^d \gamma^t R(s_t, a_t) \\ & \text{subject to} \quad s_{t+1} = \mathbb{E}_{s' \sim T(s'|s,a)}[s'] \quad \forall t \end{aligned}$$

Open-Loop

$$\begin{aligned} & \underset{a_{1:d}, s_{1:d}^{(1:m)}}{\text{maximize}} \quad \frac{1}{m} \sum_{i=1}^m \sum_{t=1}^d \gamma^t R(s_t^{(i)}, a_t) \\ & \text{subject to} \quad s_{t+1} = G(s_t^{(i)}, a_t, w_t^{(i)}) \quad \forall t, i \end{aligned}$$

Hindsight
Optimization

$$\begin{aligned} & \underset{a_{1:d}^{(1:m)}, s_{1:d}^{(1:m)}}{\text{maximize}} \quad \frac{1}{m} \sum_{i=1}^m \sum_{t=1}^d \gamma^t R(s_t^{(i)}, a_t^{(i)}) \\ & \text{subject to} \quad s_{t+1} = G(s_t^{(i)}, a_t^{(i)}, w_t^{(i)}) \quad \forall t, i \\ & \quad a_1^{(i)} = a_1^{(j)} \quad \forall i, j \end{aligned}$$

Guiding Questions

- What tools do we have to solve MDPs with continuous S and A ?