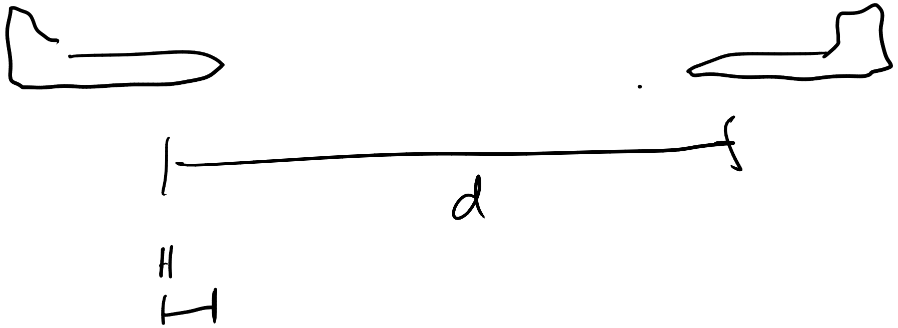
Last Time



$d = 0$, $d = v\Delta t$

Last Time

- What are the differences between online and offline solutions?
- Are there solution techniques that are *independent* of the state space size?

Guiding Questions

Guiding Questions

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

Continuous S and A

Continuous S and A

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

Continuous S and A

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

The old rules still work!

$$V^*(s) = \max_a \left(R(s,a) + \gamma \int_S T(s'|s,a)\, V^*(s') \right)$$

$$V^\pi(s) = R(s, \pi(s)) + \gamma \underset{s' \sim T(s,\pi(s))}{E} \left[V^\pi(s') \right]$$

Linear Dynamics, Quadratic Reward

Linear Dynamics, Quadratic Reward

$$s' = T_s s + T_a a + w$$

Linear Dynamics, Quadratic Reward

$$s' = T_s s + T_a a + w$$

w is zero-mean, finite variance R.V.

Linear Dynamics, Quadratic Reward

$$s' = T_s s + T_a a + w$$

w is zero-mean, finite variance

LQR

$$R(s, a) = s^\top R_s s + a^\top R_a a$$

Linear Dynamics, Quadratic Reward

$$s' = T_s s + T_a a + w$$

w is zero-mean, finite variance

$$R(s,a) = s^\top R_s s + a^\top R_a a$$

$$V_h(s) = s^\top P_h s + q_h$$

Linear Dynamics, Quadratic Reward

$$s' = T_s s + T_a a + w$$

w is zero-mean, finite variance

$$R(s,a) = s^\top R_s s + a^\top R_a a$$

$$V_h(s) = s^\top P_h s + q_h$$

$$P_{h+1} = R_s + T_s^\top P_h^\top T_s - \left(T_a^\top P_h T_s\right)^\top \left(R_a + T_a^\top P_h T_a\right)^{-1} \left(T_a^\top P_h T_s\right)$$

Linear Dynamics, Quadratic Reward

$s' = T_s s + T_a a + w$ $\qquad$ w is zero-mean, finite variance

$R(s,a) = s^\top R_s s + a^\top R_a a$

$$V_h(s) = s^\top P_h s + q_h$$

$$P_{h+1} = R_s + T_s^\top P_h^\top T_s - \left(T_a^\top P_h T_s\right)^\top \left(R_a + T_a^\top P_h T_a\right)^{-1} \left(T_a^\top P_h T_s\right)$$

$$\pi_h^*(s) = \underline{-\left(R_a + T_a^\top P_h T_a\right)^{-1} T_a^\top P_h T_s}\, s = -Ks$$

Linear Dynamics, Quadratic Reward

$$s' = T_s s + T_a a + w$$

w is zero-mean, finite variance

$$R(s,a) = s^\top R_s s + a^\top R_a a$$

$$V_h(s) = s^\top P_h s + q_h$$

$$P_{h+1} = R_s + T_s^\top P_h^\top T_s - \left(T_a^\top P_h T_s\right)^\top \left(R_a + T_a^\top P_h T_a\right)^{-1} \left(T_a^\top P_h T_s\right)$$

$$\pi_h^*(s) = -\left(R_a + T_a^\top P_h T_a\right)^{-1} T_a^\top P_h T_s s = -Ks$$

if $w \sim \mathcal{N}(0, \Sigma)$, then $q_{h+1} = \Sigma_{i=1}^h \mathrm{tr}(\Sigma P_i)$

Linear Dynamics, Quadratic Reward

$$s' = T_s s + T_a a + w$$

w is zero-mean, finite variance

$$R(s,a) = s^\top R_s s + a^\top R_a a$$

$$V_h(s) = s^\top P_h s + q_h$$

$$P_{h+1} = R_s + T_s^\top P_h^\top T_s - \left(T_a^\top P_h T_s\right)^\top \left(R_a + T_a^\top P_h T_a\right)^{-1} \left(T_a^\top P_h T_s\right)$$

$$\pi_h^*(s) = -\left(R_a + T_a^\top P_h T_a\right)^{-1} T_a^\top P_h T_s s = -Ks$$

if $w \sim \mathcal{N}(0, \Sigma)$, then $q_{h+1} = \Sigma_{i=1}^h \mathrm{tr}(\Sigma P_i)$

"Linear Quadratic Regulator" / "LQR"

If not Linear Quadratic…

Offline:

Online:

If not Linear Quadratic...

Offline:

- Approximate Dynamic Programming (ADP)

Online:

If not Linear Quadratic...

Offline:

- Approximate Dynamic Programming (ADP) V_θ

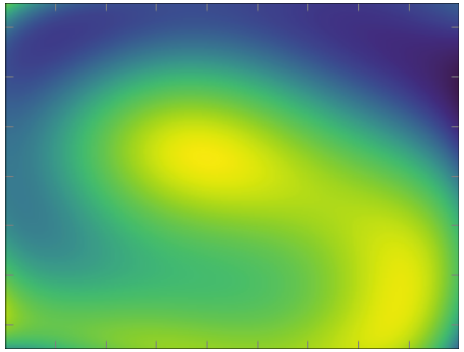
Online:

If not Linear Quadratic...

<u>Offline:</u>

- Approximate Dynamic Programming (ADP)

V_θ



<u>Online:</u>

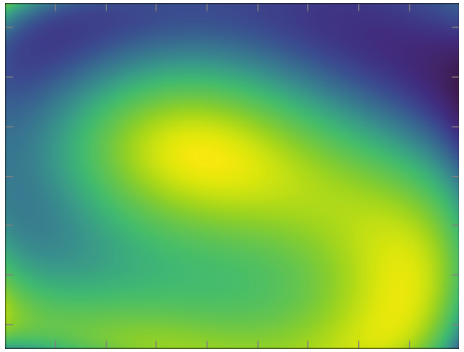
If not Linear Quadratic...

<u>Offline:</u>

- Approximate Dynamic Programming (ADP)
- Policy Search

<u>Online:</u>

V_θ



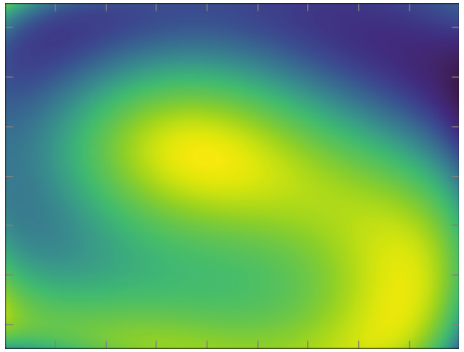
If not Linear Quadratic...

<u>Offline:</u>

- Approximate Dynamic Programming (ADP)
- Policy Search

<u>Online:</u>

V_θ



π_θ

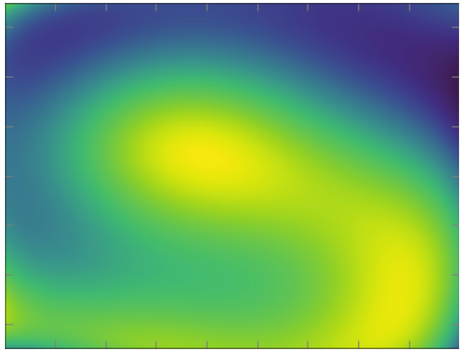
If not Linear Quadratic...

<u>Offline:</u>

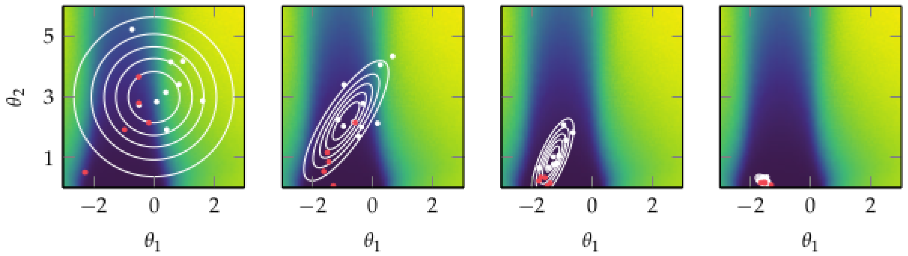
- Approximate Dynamic Programming (ADP)
- Policy Search

<u>Online:</u>

V_θ



π_θ



If not Linear Quadratic…

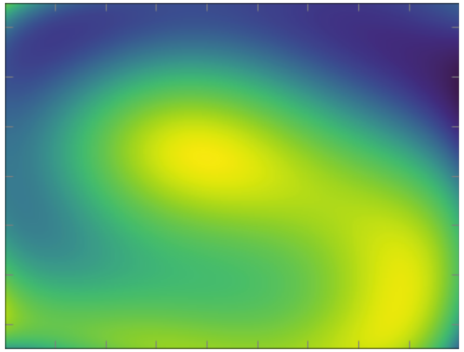
Offline:

- Approximate Dynamic Programming (ADP)
- Policy Search

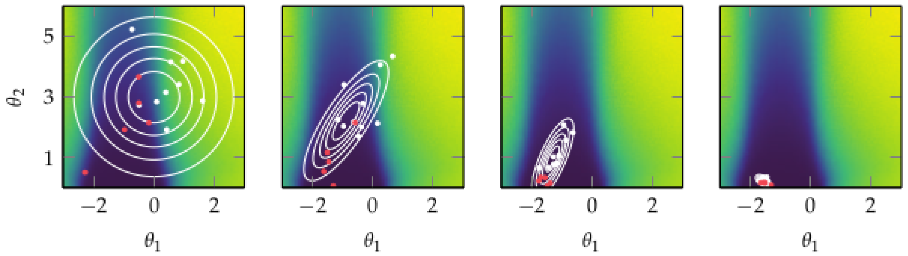
Online:

- Model Predictive Control (MPC)

V_θ



π_θ



If not Linear Quadratic…

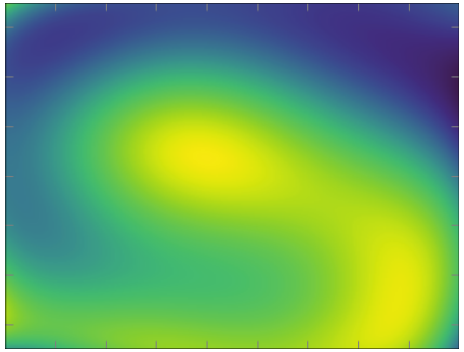
<u>Offline:</u>

- Approximate Dynamic Programming (ADP)
- Policy Search

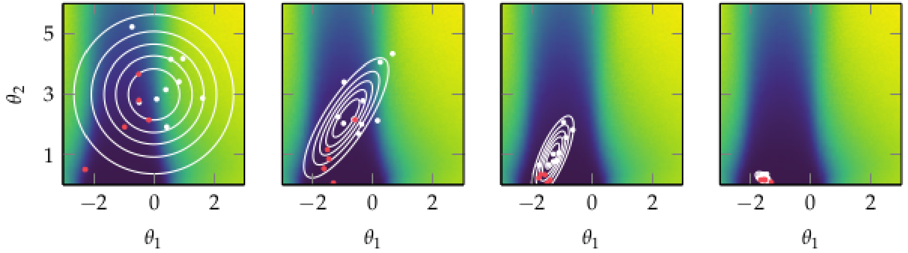
<u>Online:</u>

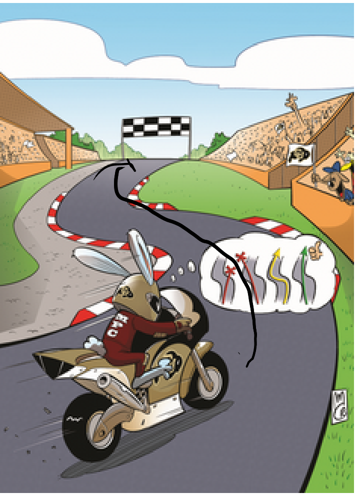
- Model Predictive Control (MPC)

V_θ



π_θ





If not Linear Quadratic...

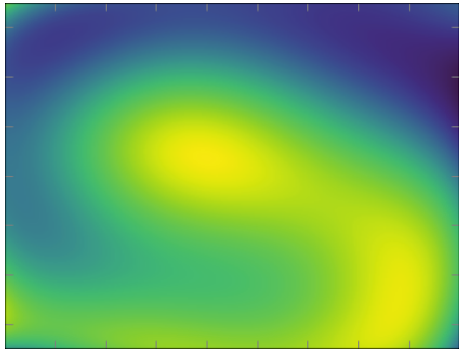
<u>Offline:</u>

- Approximate Dynamic Programming (ADP)
- Policy Search

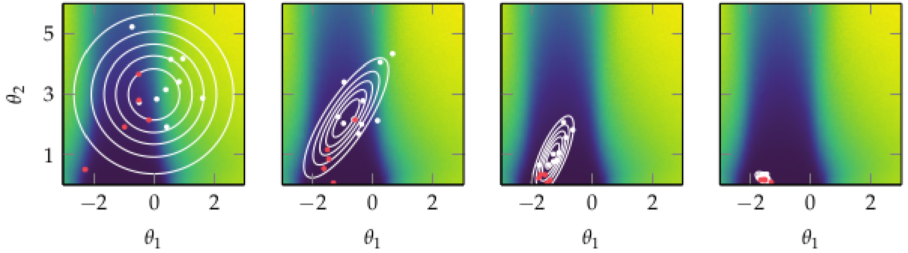
<u>Online:</u>

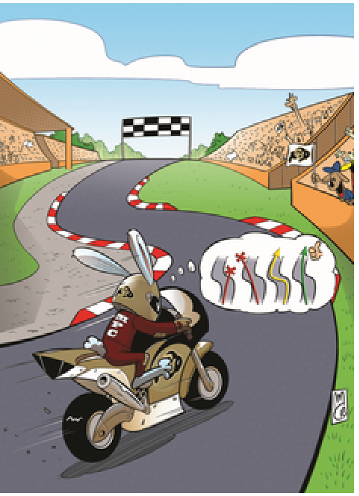
- Model Predictive Control (MPC)
- Sparse Tree Search/Progressive Widening

V_θ



π_θ





If not Linear Quadratic...

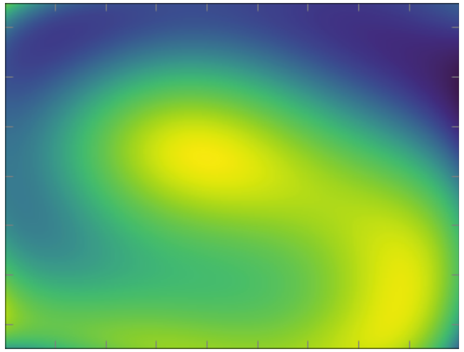
<u>Offline:</u>

- Approximate Dynamic Programming (ADP)
- Policy Search

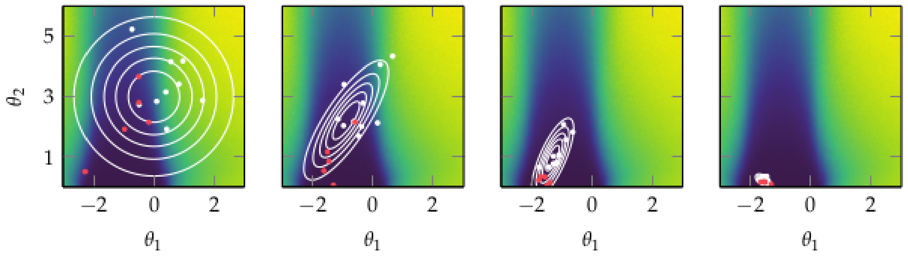
<u>Online:</u>

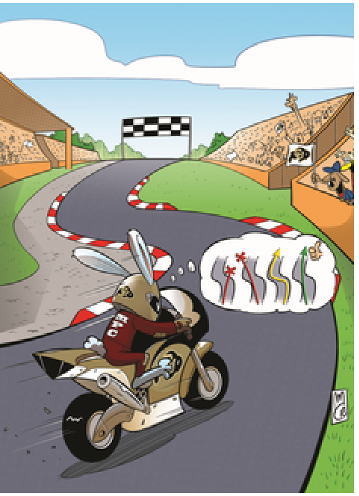
- Model Predictive Control (MPC)
- Sparse Tree Search/Progressive Widening

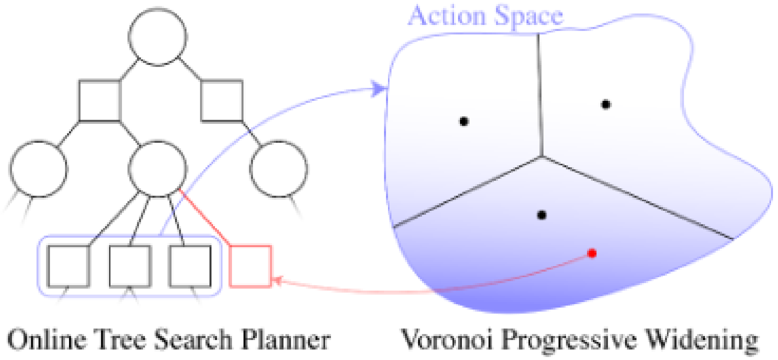
V_θ



π_θ







Approximate Dynamic Programming

Approximate Dynamic Programming

$V_\theta(s) = f_\theta(s)$ (e.g. neural network)

Approximate Dynamic Programming

$V_\theta(s) = f_\theta(s)$ (e.g. neural network)

$V_\theta(s) = \theta^\top \beta(s)$ (linear feature)

Approximate Dynamic Programming

$V_\theta(s) = f_\theta(s)$ (e.g. neural network)

$V_\theta(s) = \theta^\top \beta(s)$ (linear feature)

while not converged

$\theta \leftarrow \theta'$

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

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

Approximate Dynamic Programming

$V_\theta(s) = f_\theta(s)$ (e.g. neural network)

$V_\theta(s) = \theta^\top \beta(s)$ (linear feature)

while not converged

$\theta \leftarrow \theta'$

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

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

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

Monte Carlo

Approximate Dynamic Programming

$V_\theta(s) = f_\theta(s)$ (e.g. neural network)

$V_\theta(s) = \theta^\top \beta(s)$ (linear feature)

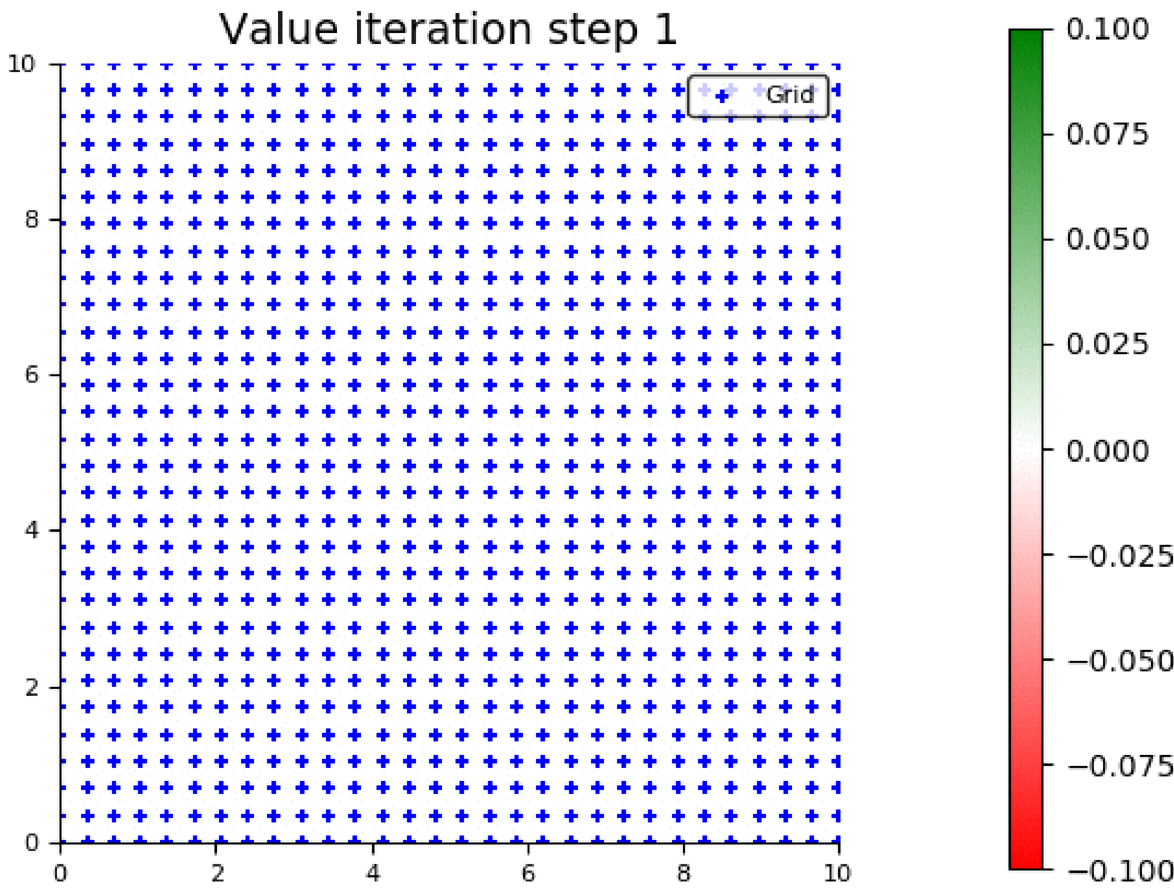
while not converged

$\theta \leftarrow \theta'$

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

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

LSVI



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

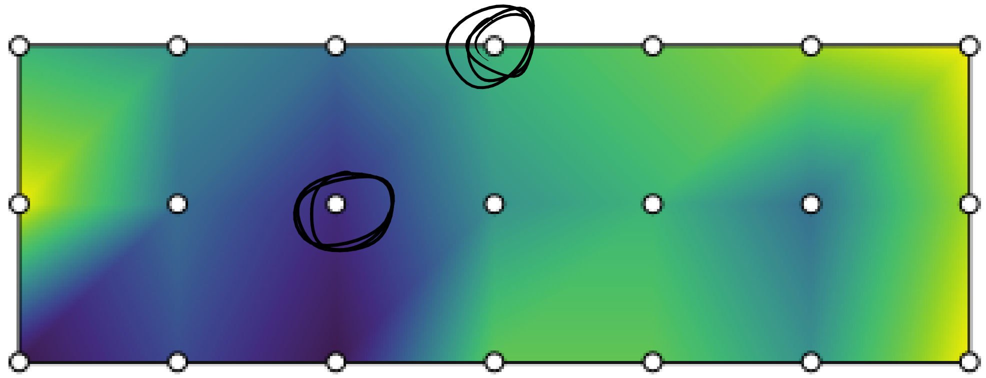
Function Approximation

Function Approximation

- Local: (e.g. simplex interpolation)

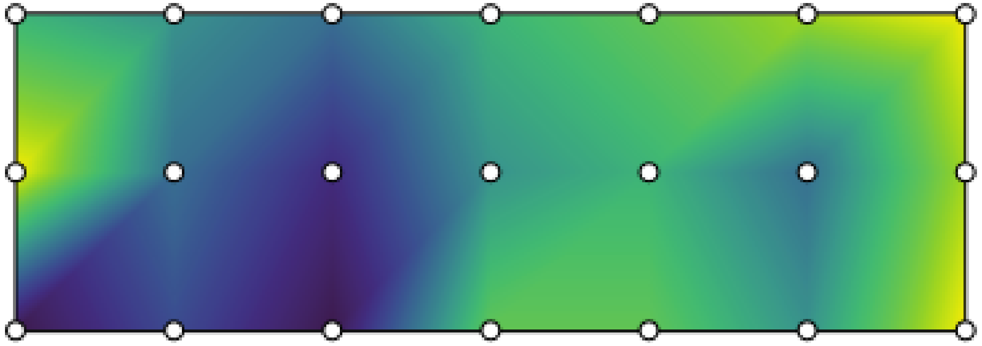
Function Approximation

- Local: (e.g. simplex interpolation)



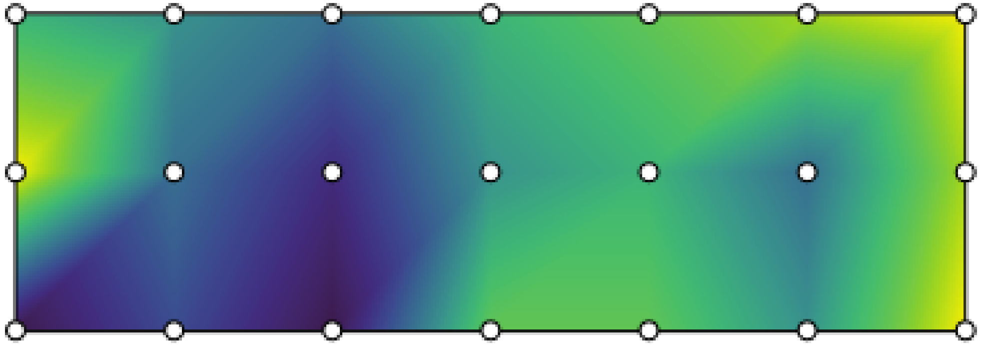
Function Approximation

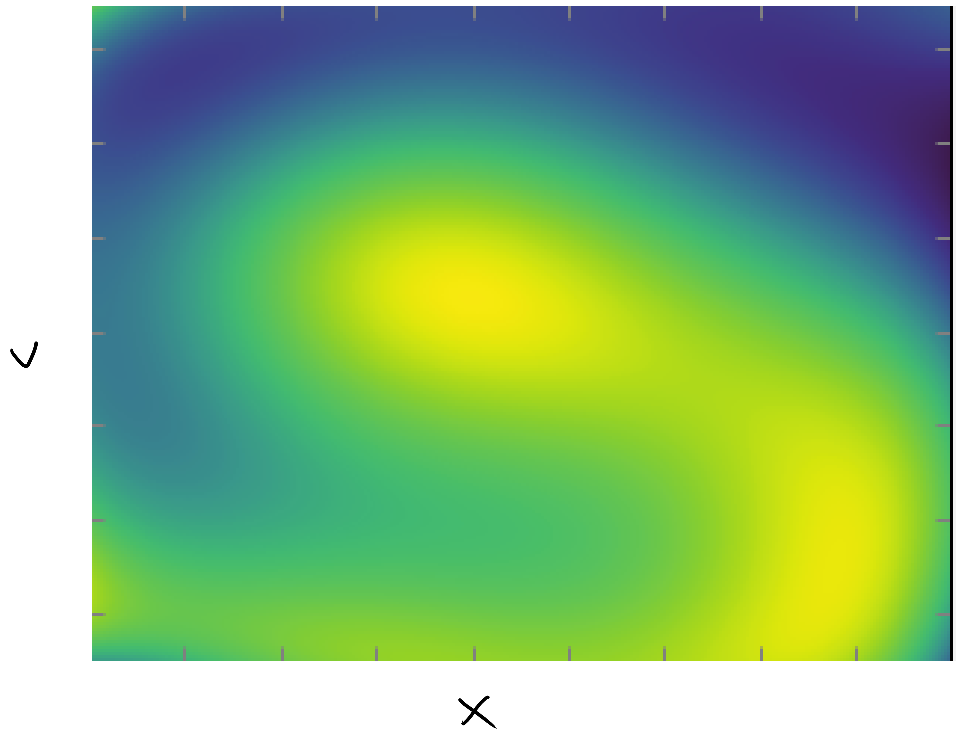
- Local: (e.g. simplex interpolation)
- Global: (e.g. Fourier, neural network)



Function Approximation

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





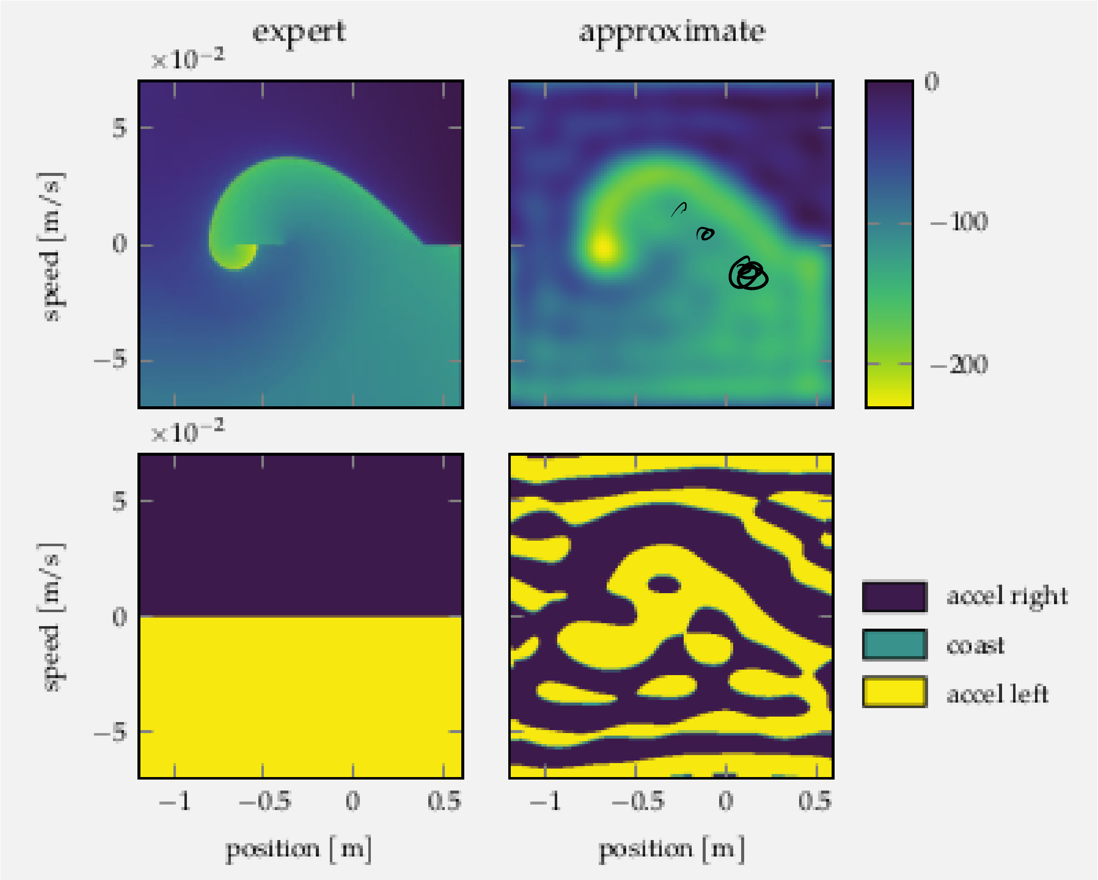
$s = (x, v)$

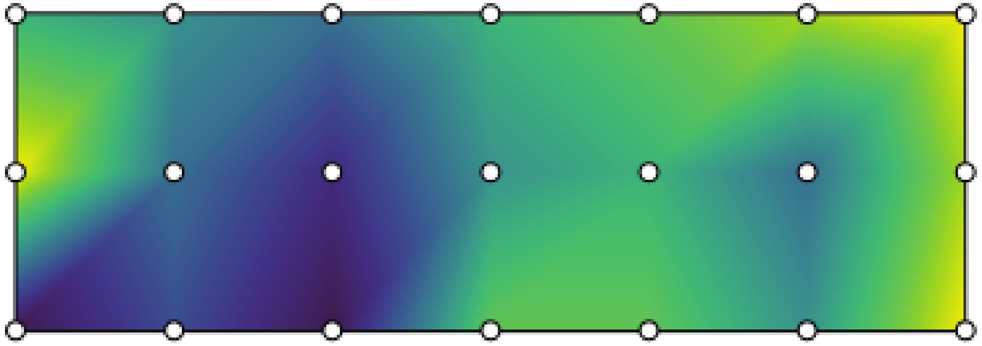
$V_\theta(s) = \theta^T \beta(s)$

$$\beta(s) = \begin{array}{l} [1, \\ x, \quad v, \\ x^2, \quad xv, \quad v^2, \\ x^3, \quad x^2v, \quad xv^2, \quad v^3, \\ x^4, \quad x^3v, \quad x^2v^2, \quad xv^3, \quad v^4, \\ x^5, \quad x^4v, \quad x^3v^2, \quad x^2v^3, \quad xv^4, \quad v^5, \\ x^6, \quad x^5v, \quad x^4v^2, \quad x^3v^3, \quad x^2v^4, \quad xv^5, \quad v^6] \end{array}$$

Function Approximation

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





Policy Search

Policy Search

$$\underset{\theta}{\text{maximize}} \quad U(\pi_\theta)$$

Policy Search

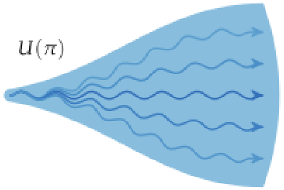
$$\underset{\theta}{\text{maximize}} \quad U(\pi_\theta)$$

$$U(\pi) \approx \frac{1}{m} \sum_{i=1}^{m} R(\tau^{(i)})$$

Policy Search

$$\underset{\theta}{\text{maximize}} \quad U(\pi_\theta)$$

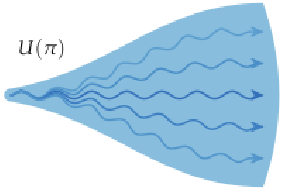
$$U(\pi) \approx \frac{1}{m}\sum_{i=1}^{m} R(\tau^{(i)})$$



Policy Search

$$\underset{\theta}{\text{maximize}} \quad U(\pi_\theta)$$

$$U(\pi) \approx \frac{1}{m}\sum_{i=1}^{m} R(\tau^{(i)})$$

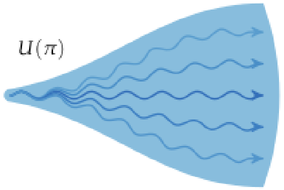


Common Approaches

Policy Search

$$\underset{\theta}{\text{maximize}} \quad U(\pi_\theta)$$

$$U(\pi) \approx \frac{1}{m}\sum_{i=1}^{m} R(\tau^{(i)})$$



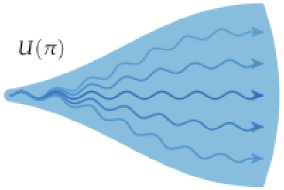
<u>Common Approaches</u>

- Evolutionary Algorithms

Policy Search

$$\underset{\theta}{\text{maximize}} \quad U(\pi_\theta)$$

$$U(\pi) \approx \frac{1}{m}\sum_{i=1}^{m} R(\tau^{(i)})$$



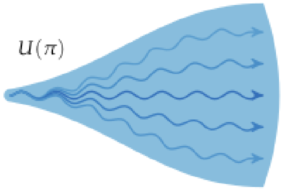
Common Approaches

- Evolutionary Algorithms
- Cross Entropy Method

Policy Search

$$\underset{\theta}{\text{maximize}} \quad U(\pi_\theta)$$

$$U(\pi) \approx \frac{1}{m} \sum_{i=1}^{m} R(\tau^{(i)})$$



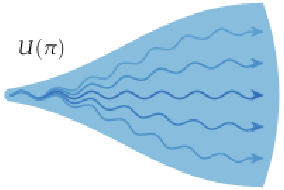
<u>Common Approaches</u>

- Evolutionary Algorithms
- Cross Entropy Method
- Policy Gradient (will cover in RL section)

Policy Search

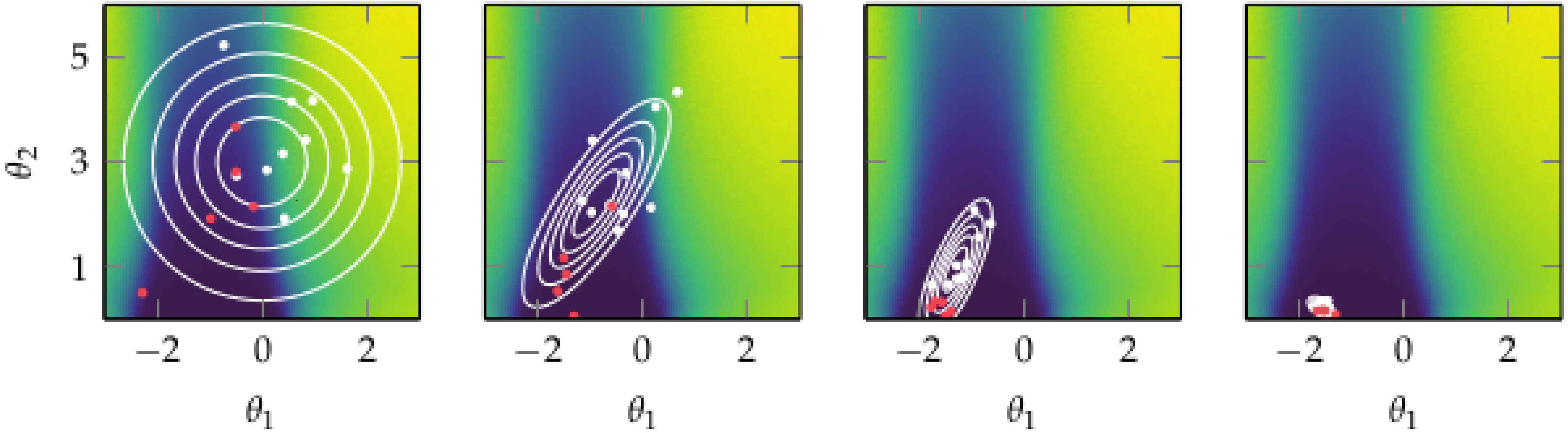
$$\underset{\theta}{\text{maximize}} \quad U(\pi_\theta)$$

$$U(\pi) \approx \frac{1}{m}\sum_{i=1}^{m} R(\tau^{(i)})$$



<u>Common Approaches</u>

- Evolutionary Algorithms
- Cross Entropy Method
- Policy Gradient (will cover in RL section)



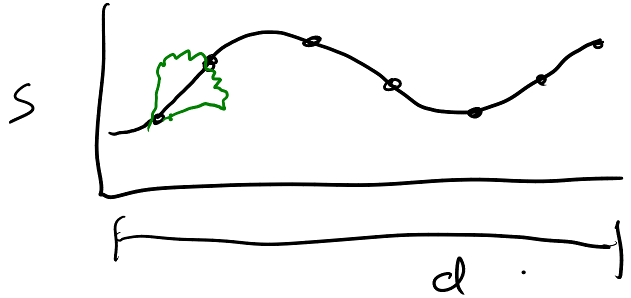
Model Predictive Control

Model Predictive Control

$$\underset{a_{1:d}, s_{1:d}}{\text{maximize}} \quad \sum_{t=1}^{d} \gamma^t R(s_t, a_t)$$

$$\text{subject to} \quad s_{t+1} = \mathrm{E}[T(s_t, a_t)] \quad \forall t$$

mean



Model Predictive Control

$$\underset{a_{1:d}, s_{1:d}}{\text{maximize}} \quad \sum_{t=1}^{d} \gamma^t R(s_t, a_t)$$

$$\text{subject to} \quad s_{t+1} = \mathrm{E}[T(s_t, a_t)] \quad \forall t$$

$$\underset{a_{1:d}, s_{1:d}^{(1:m)}}{\text{maximize}} \quad \frac{1}{m} \sum_{i=1}^{m} \sum_{t=1}^{d} \gamma^t R(s_t^{(i)}, a_t)$$

$$\text{subject to} \quad s_{t+1} = G(s_t^{(i)}, a_t, w_t^{(i)}) \quad \forall t, i$$

Model Predictive Control

$$\underset{a_{1:d}, s_{1:d}}{\text{maximize}} \quad \sum_{t=1}^{d} \gamma^t R(s_t, a_t)$$

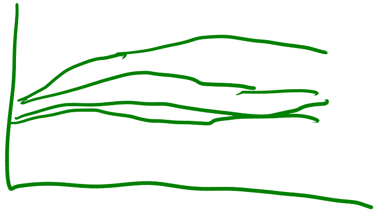
$$\text{subject to} \quad s_{t+1} = \mathrm{E}[T(s_t, a_t)] \quad \forall t$$

Optimistic

$$\underset{a_{1:d}, s_{1:d}^{(1:m)}}{\text{maximize}} \quad \frac{1}{m} \sum_{i=1}^{m} \sum_{t=1}^{d} \gamma^t R(s_t^{(i)}, a_t)$$

$$\text{subject to} \quad s_{t+1} = G(s_t^{(i)}, a_t, w_t^{(i)}) \quad \forall t, i$$

Open Loop



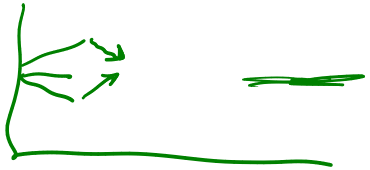
Pessimistic /conservative

$$\underset{a_{1:d}^{(1:m)}, s_{1:d}^{(1:m)}}{\text{maximize}} \quad \frac{1}{m} \sum_{i=1}^{m} \sum_{t=1}^{d} \gamma^t R(s_t^{(i)}, a_t^{(i)})$$

$$\text{subject to} \quad s_{t+1} = G(s_t^{(i)}, a_t^{(i)}, w_t^{(i)}) \quad \forall t, i$$

$$a_1^{(i)} = a_1^{(j)} \quad \forall i, j$$

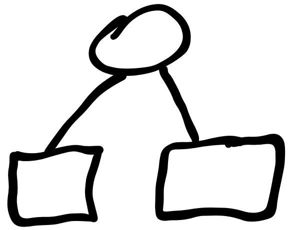
Hindsight Opt



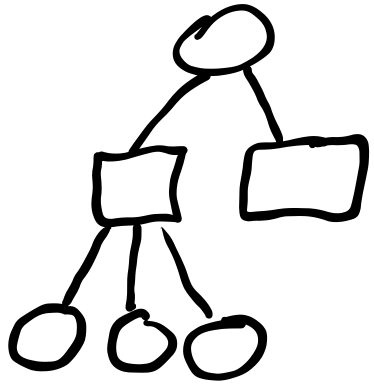
Optimistic

Sparse Tree Search/Progressive Widening

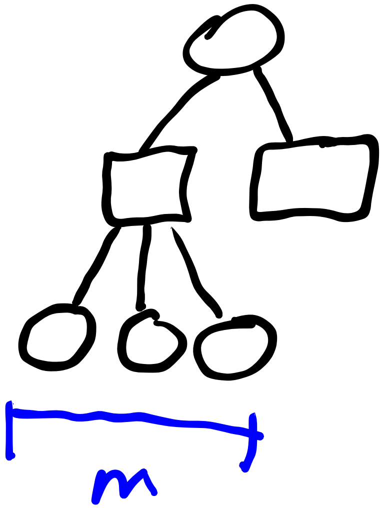
Sparse Tree Search/Progressive Widening



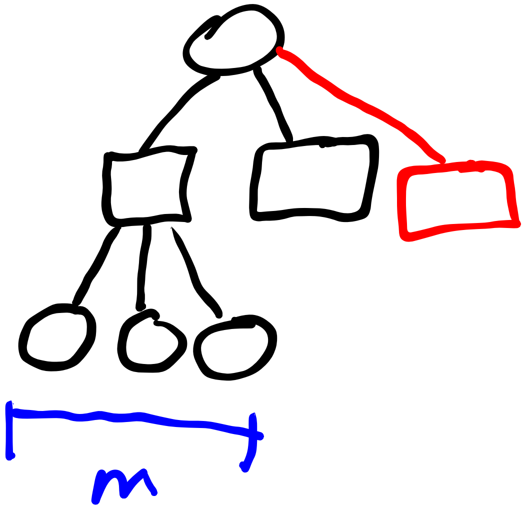
Sparse Tree Search/Progressive Widening



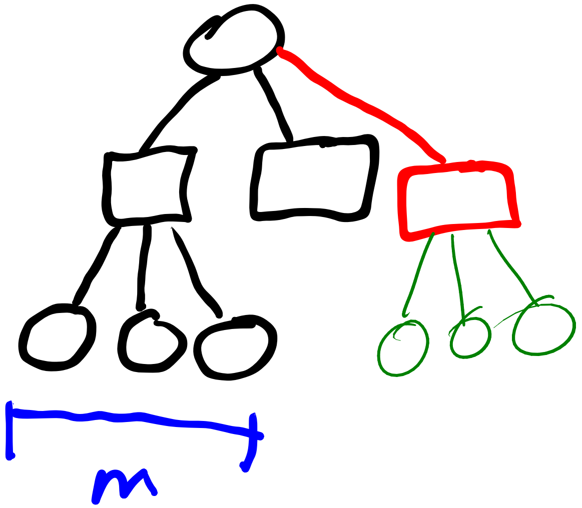
Sparse Tree Search/Progressive Widening



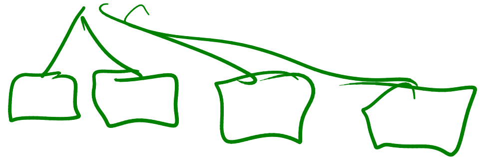
Sparse Tree Search/Progressive Widening

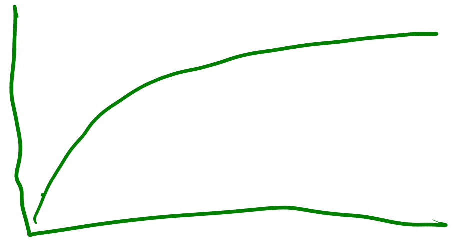


Sparse Tree Search/Progressive Widening

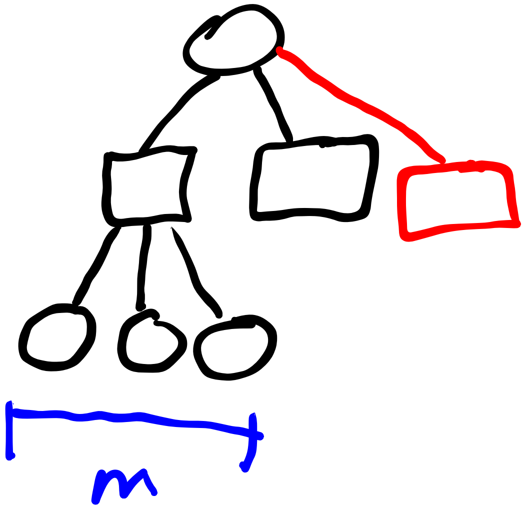


add new branch if $C < kN^{\alpha}$

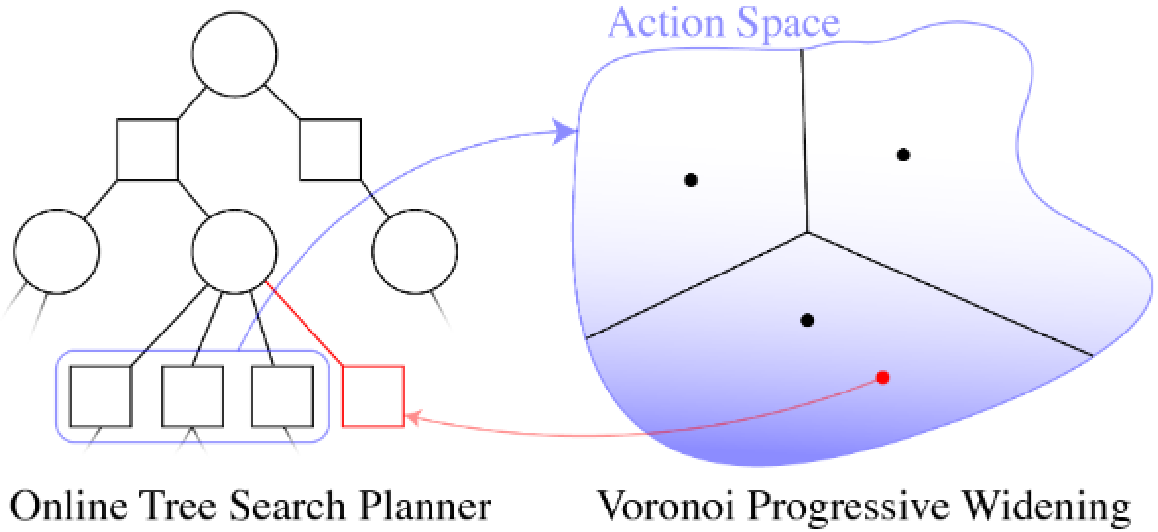




Sparse Tree Search/Progressive Widening



add new branch if $C < kN^{\alpha}$



Guiding Questions

Guiding Questions

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