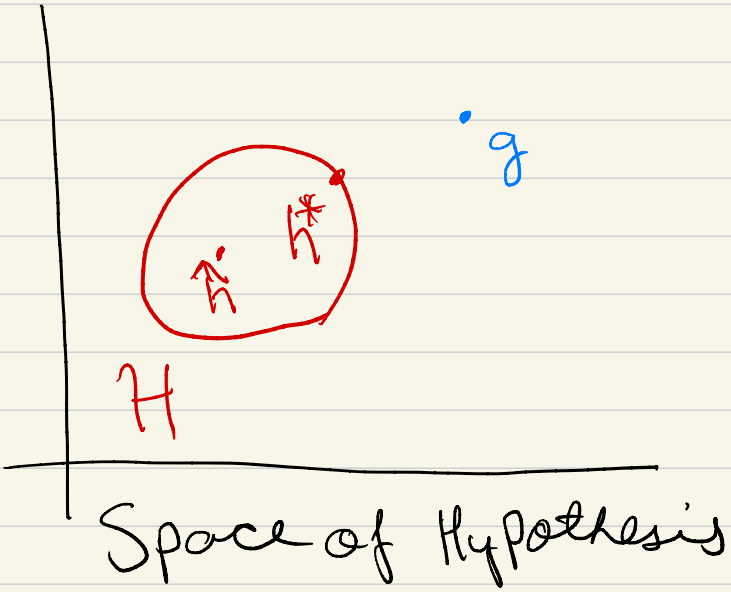
4.7 Approximation & Estimation Error



g : Best Possible Hypothesis

H : Set of all hypothesis of a particular type
Eg: Set of all SVMs

h^* : Best possible hypothesis of that class: Eg: Most optimal SVM

$\hat{h}$: Learned from finite Data

$\mathcal{E}(h)$ = Risk / Generalization Error

$= E_{(x,y)\sim D}[1\{h(x) \neq y\}]$

$\hat{\mathcal{E}}_S(h)$: Empirical Risk

$= \frac{1}{m}\sum_{i=1}^{m} 1\{h(x^{(i)}) \neq y^{(i)}\}$

$\mathcal{E}(g)$ = Bayes Error / Irreducible (unavoidable)

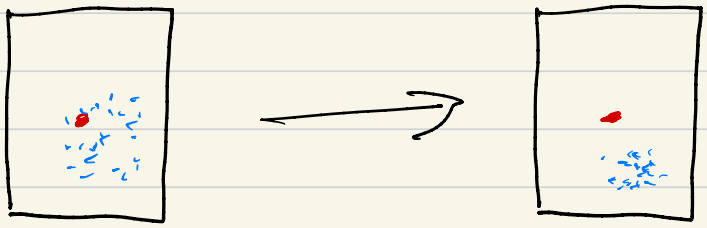
$\mathcal{E}(h^*) - \mathcal{E}(g)$ = Approximation Error (class)

$\mathcal{E}(\hat{h}) - \mathcal{E}(h^*)$ = Estimation Error (Data)

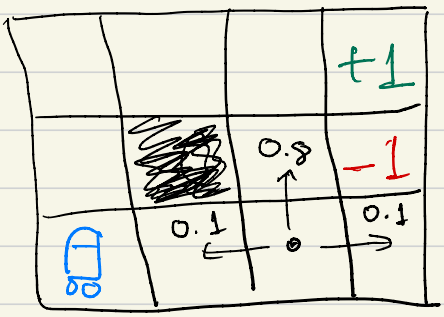
Total Error = Estimation + Approximation + Irreducible

Est Var + Est Bias + Approx + Irreducible

Variance Bias



RL Intro



Total Payoff:

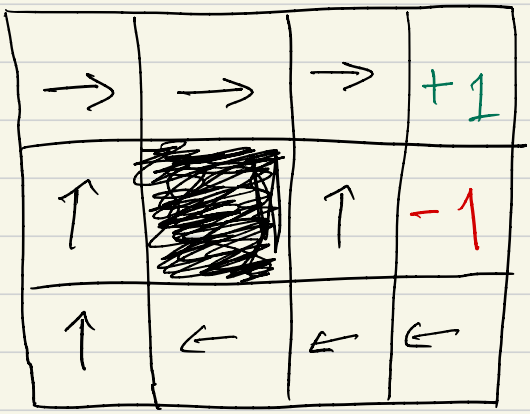
$$R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \dots$$

$\gamma < 1$ (usually 0.99)

Goal of RL is to maximise

$$E\left[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \dots\right]$$

Optimal Policy



Define: V^{π}, V^{*}, π^{*}

$$V^{\pi}(s) = E\left[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \dots \mid \pi, s_0 = s\right]$$

$$V^{\pi}(s) = R(s) + \gamma \sum_{s' \in S} P_{s\pi(s)}(s')\, V^{\pi}(s')$$

- $R(s)$: Immediate Reward
- γ: Discount
- $\sum_{s' \in S}$: Computes Expectation
- $P_{s\pi(s)}(s')$: Transition Probability
- $V^{\pi}(s')$: Recursive Value Function

$$V^*(s) = \max_\pi V^\pi(s)$$

$$V^*(s) = R(s) + \max_a \gamma \sum_{s'} P_{sa}(s') V^*(s')$$

Expected future reward

P_{sa}: if we take action a

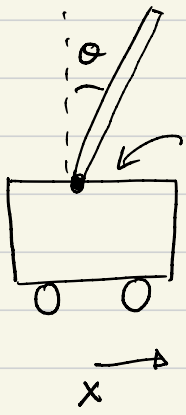
$$\pi^*(s) = \arg\max_{a \in A} \sum_{s'} P_{sa}(s') V^*(s')$$

P_{sa}: if we take action a

What if I don't know P_{sa}?

$$P_{sa}(s') = \frac{\#\text{ times I took action } a \text{ in } s \text{ and got } s'}{\#\text{ times I took action } a \text{ in } s}$$

(or $\frac{1}{|S|}$ if above is $\frac{0}{0}$)



State space = $[x, \theta, \dot{x}, \dot{\theta}]$

$s \in \mathbb{R}^N$

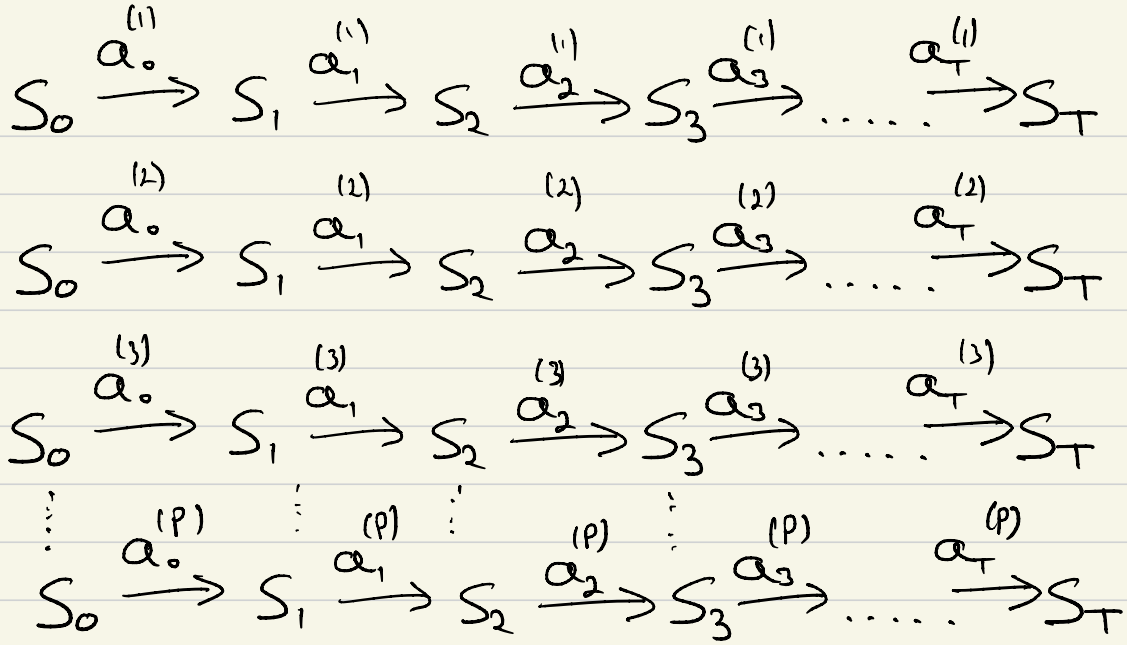
where N is dimensionality of state space

In Linear Regression, we try to approximate

$$y \approx \theta^T \phi(x)$$

(θ: Parameters (Model); $\phi(x)$: Features; x: State)

$$V^*(s) \approx \theta^T \phi(s)$$



S_{t+1} as function of S_t, a_t

(S') (S, a)

Eg: Linear Regression version

$$S_{t+1} = \underbrace{A}_{\mathbb{R}^{N \times N}} S_t + \underbrace{B}_{\mathbb{R}^{N \times K}} a_t$$

(tall skinny matrix)

Objective:

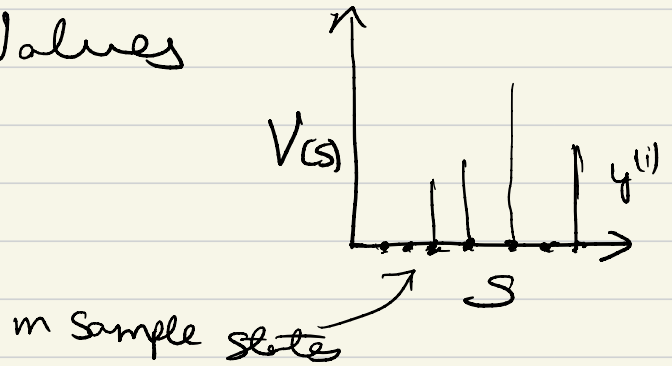
$$\min_{A,B} \sum_{i=0}^{P} \sum_{t=0}^{T} \left\| S_{t+1}^{(i)} - \left(A S_t^{(i)} + B a_t^{(i)}\right) \right\|$$

$$S_{t+1} = AS_t + BA_t + \varepsilon_t$$ (Stochastic Model)

ε_t ↑ Noise

Learn a mapping from $S \rightarrow V(S)$

States $\xrightarrow{to}$ Values



1. Randomly sample n states $s^{(1)}, s^{(2)}, \ldots s^{(n)} \in S$.
2. Initialize $\theta := 0$.
3. Repeat {

actions are discrete

For $i = 1, \ldots, n$ {

For each action $a \in A$ {

Stochastic => sample then average

Sample $s'_1, \ldots, s'_k \sim P_{s^{(i)}a}$ (using a model of the MDP).

Set $q(a) = \frac{1}{k} \sum_{j=1}^{k} R(s^{(i)}) + \gamma V(s'_j)$

old estimate of $V = \theta^T \phi(s)$

// Hence, $q(a)$ is an estimate of $R(s^{(i)}) + \gamma \mathrm{E}_{s' \sim P_{s^{(i)}a}}[V(s')]$.

}

Set $y^{(i)} = \max_a q(a)$.

// Hence, $y^{(i)}$ is an estimate of $R(s^{(i)}) + \gamma \max_a \mathrm{E}_{s' \sim P_{s^{(i)}a}}[V(s')]$.

}

// In the original value iteration algorithm (over discrete states)

// we updated the value function according to $V(s^{(i)}) := y^{(i)}$.

// In this algorithm, we want $V(s^{(i)}) \approx y^{(i)}$, which we'll achieve

// using supervised learning (linear regression).

Set $\theta := \arg\min_\theta \frac{1}{2} \sum_{i=1}^{n} \left(\theta^T \phi(s^{(i)}) - y^{(i)}\right)^2$

}

update V by learning new mapping $S \to V(S)$

(can also be a neural network or any supervised learning algo

$$V_t^*(s) = \max_a R(s,a) + \sum_{s'} P_{sa}(s') V_{t+1}^*(s')$$

$$V_T^*(s) = \max_a R(s,a)$$

LQR:

Applies to an MDP: $(S, A, \{P_{sa}\}, T, R)$

$S: \mathbb{R}^n \quad A: \mathbb{R}^d$

$$s_{t+1} = \underbrace{A}_{\mathbb{R}^{n\times n}} s_t + \underbrace{B}_{\mathbb{R}^{n\times d}} a_t + \underbrace{w_t}_{\hookrightarrow \sim \mathcal{N}(0, \Sigma_w)}$$

Matrices

Linear system

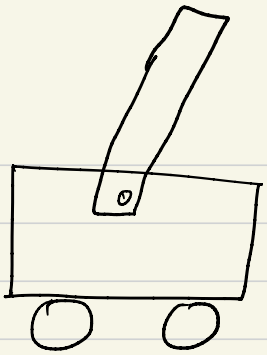
$$R(s,a) = -(s^T U s + a^T V a)$$

Reward

where $U \in \mathbb{R}^{n\times n}$, $V \in \mathbb{R}^{d\times d}$

$U, V \geq 0$ (positive semi-definite)

Quadratic cost function



$$a \approx \pi_\theta(s) = \frac{1}{1+e^{-\theta^T s}}$$

Let us assume that we train a logistic

$$\pi_\theta(s, \text{"R"}) = \frac{1}{1+e^{-\theta^T s}}$$

$$\pi_\theta(s, \text{"L"}) = 1 - \frac{1}{1+e^{\theta^T s}}$$

$$s = \begin{pmatrix} 1 \\ x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{pmatrix}$$

Suppose policy $\theta = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$

$$\pi_\theta(s, \text{"R"}) = \frac{1}{1+e^{-\phi}}$$

Intuitively, policy aggresively pushes the cart to the right as the pole angle ϕ increases

Goal: Find θ so that when we execute $\pi_\theta(s, a)$ we maximize

$$\max_\theta E[R(s_0, a_0) + \dots + R(s_T, a_T) \mid \pi_\theta]$$

Observation $y = \begin{pmatrix} x \\ \phi \end{pmatrix}$ + noise

↖ only a partial state is measured.

Cannot use $V^*(s)$ because s is not fully observable.

Policy search attempts to learn $\pi^*(y)$

Assume $T = 1$

$$E[R(s_0, a_0) + R(s_1, a_1) \mid \pi_\theta]$$

$$= \sum_{s_0, a_0, s_1, a_1} P(s_0) \pi_\theta(s_0, a_0) P_{s_0 a_0}(s_1) \pi_\theta(s_1, a_1) \cdot [\text{payoff}]$$

Reinforce Algorithm

Loop: {

← Run your MDP (using current policy)

Sample $s_0, a_0, s_1, a_1 \ldots s_T, a_T$

Compute payoff $= R(s_0) + R(s_1) + \ldots R(s_T)$

Update

$$\theta : \theta + \alpha \left[\frac{\nabla_\theta \pi_\theta(s_0, a_0)}{\pi_\theta(s_0, a_0)} + \frac{\nabla_\theta \pi_\theta(s_1, a_1)}{\pi_\theta(s_1, a_1)} + \ldots \right] \cdot \text{payoff}$$

$$\pi(s,a) = \frac{e^{\theta^T s}[a]}{\sum_{a'} e^{\theta^T s}[a']}$$

indexing at action with a certain index

$$\frac{e^{\theta^T s}}{\sum_{a'} e^{\theta^T s}} \Bigg\} \rightarrow (\text{num\_states}, \text{num\_actions})$$

$$\nabla_\theta \ln\left(\pi_\theta(s_t, a_t)\right)$$

$$\nabla_\theta \ln\left(\frac{e^{\theta^T s_t}[a_t]}{\sum_{a'} e^{\theta^T s_t}[a']}\right)$$