- Stochastic gradient descent :-

$\min_{x} f(x)$. $\nabla f(x)$.

$\circledast$ $\min_{\boxed{\theta}} \underset{x}{E}[f(x,\theta)]$.

$\downarrow$

$\min_{\theta} \boxed{\sum_{x} \mu(x) f(x,\theta)}$.

RGD: $x_{n+1} = x_n - \alpha_n \nabla f(x_n)$.

$\boxed{*}$ $\theta_{n+1} = \theta_n - \alpha_n \left[\sum_{x} \mu(x) \nabla_{\theta} f(x,\theta) \right]$

$\boxed{*}$ Assuming $\mu(x)$ is known.

Stochastic GD :-

$\boxed{*}$ At time instant 'n': you are given a sample $x_n \sim \mu(\cdot)$. ← condition

$$\boxed{\theta_{n+1} \leftarrow \theta_n - \alpha_n \left(\nabla_{\theta} f(x_n, \theta_n) \right).}$$

Approximation :- $\boxed{Q(s,a)}$:-

→ states are very high, infinite !!

↳ Mountain car :- $\begin{bmatrix} 0.6\ldots \\ \end{bmatrix}$

MDP

↓

If model info is known < V.I, P.I

↓

Model info is not known
C.P.T.M Not known < Q-learn, SARSA, Exp sarsa

↓

+ No. of states are too high / infinite } → Approximation.

Prediction :- given a policy π, want $V^{\pi}(\cdot)$

Parametrization : $V^{\pi}(s) \approx \hat{V}(s, w)$ ↳ $w^{T}\phi(s)$

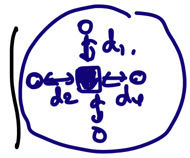
Parameters — w

features corresponding to state s — $\phi(s)$

coordinates

$$\text{feature}(s) = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ \text{speed} \\ \text{acceleration} \end{bmatrix}$$

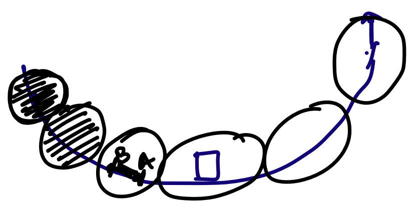
feature set



• Generalization :- "If two states should have similar value fns, then their features should be closer".

$V(s_1) = 10$ $\qquad$ $V(s_2) = 10.2$

Approx: $w^{T}\phi(s_1)$ $\qquad$ $w^{T}\phi(s_2)$



• "Approximated value function should be closer to the actual value function"

$$J(w) = E\left[\left(V(s) - \hat{V}(s;w)\right)^2\right]$$

$\boxed{s \sim \mu(\cdot)}$ ↳ stationary distn.

π : MDP + π → $\boxed{MC}$ ↳ stationary dn. ↳ μ

• $w_{n+1} = w_n + \frac{1}{2}\alpha\left[V(s) - \hat{V}(s,w)\right]\left(\nabla\hat{V}(s,w)\right)$. ← ?

approximate $V(s)$ and use an estimate of $V(s)$ in above eq.

Monte-Carlo estimate :-

$$V^{\pi}(s) = \mathbb{E}_{\tau}[G_\tau].$$

gradient M.C algorithm :-

$$w_{n+1} = w_n + \frac{\alpha}{\alpha}\left[\underbrace{G_\tau - \hat{v}(s,w)}\right] \nabla \hat{v}(s,w).$$

At time n :- generate a trajectory τ :

compute G_τ :-

$$\underbrace{s_1, a, 1, s_2, a_2, 2, s_3, a_3, 4, s_4.}$$

$$G_\tau : \underbrace{1 + \gamma(2) + \gamma^2(4)}$$

total discounted sum.

$\boxed{\text{TD(0) estimate}}$:-

$$V^{\pi}(s) = \mathbb{E}[r + \gamma \underbrace{V^{\pi}(s')}]$$

unbiased → estimate of $V^{\pi}(\cdot)$

$$\boxed{s' \sim P(\cdot \mid s, a \sim \pi)}.$$

semi gradient TD(0) update rule :-

$$w_{n+1} \leftarrow w_n + \alpha\left[\boxed{r + \gamma V^{\pi}(s')} - \hat{v}(s,w)\right] \nabla \hat{v}(s,w).$$

→ $\hat{v}(s', w)$.

$\boxed{\text{Not known !}}$

$$\boxed{w_{n+1} \leftarrow w_n + \alpha\left[r + \gamma \hat{v}(s', w) - \hat{v}(s,w)\right] \nabla \hat{v}(s,w)}$$

$$\boxed{n : (s, a, r, s').}$$

$$\boxed{\hat{v}(s,w) = w^T \phi(s)}$$

Policy gradient :-

(control)

"construct an objective function which can be optimized to obtain optimal policy" ①

Bellman Eqn
↓ Model free
SARSA

Q-Bellman
↓ Model free
Q-learning.

1. Parameterization of policy:

$$\pi(s) \approx \pi(s, \theta)$$

θ → Parameter to be optimized.

$$\pi_\theta(s,a) = \frac{e^{\theta^T \phi(s,a)}}{\sum_b e^{\theta^T \phi(s,b)}}$$

one example

$\phi(s,a)$ → features corr. to state (s) & action (a)

$\pi(s, a_1) \rightarrow$
$\pi(s, a_2) \rightarrow$
⋮
a_n

$$a^* = \arg\max_a Q(s,a)$$

2. constructing the objective function :-

$$J(\theta) = V_{\pi_\theta}(s_0)$$

(θ)

→ Find 'θ' s.t value fn. starting from state s_0 is maximum.

$s_0 \leftarrow$ fixed initial state

$V(s) \leftarrow$ value fn. correspondy to state s.

3. can be optimized:

$\nabla J(\theta)$

MC. TD(0).

$$\nabla J(\theta) \propto \sum_s \mu(s) \sum_a q_\pi(s,a) \nabla \pi(s,a)$$

$$\underset{s \sim \mu(s)}{E} \left[\sum_a q_\pi(s,a) \cdot \nabla \pi(s,a) \right].$$

$$\theta^* \leftarrow \arg\max_\theta J(\theta)$$

$$\theta_{n+1} \leftarrow \theta_n + \alpha_n \nabla J(\theta_n)$$

↳ needs to be estimated

$\boxed{s \sim \mu(\cdot)}$

1. computing : $\nabla \pi(s, \theta)$.
2. Approximating : $q_\pi(s,a)$ — MC $\rightarrow$ Reinforce ; TD(0) $\rightarrow$ Actor-critic
3. sampling problem : $s \sim \underbrace{\mu(\cdot)}_{\text{not known.}}$

general rule : $\theta_{t+1} = \theta_t + \alpha \left(\sum_a \hat{q}(s,a) \cdot \nabla \pi(s,a) \right)$.

$$\boxed{(*)\ \nabla \log \pi(s,a) = \frac{\nabla \pi(s,a)}{\pi(s,a)}}$$

$$\sum_s \mu(s) \sum_a q_\pi(s,a) \cdot \nabla \pi(s,a).$$

$$= \sum_s \mu(s) \sum_a q_\pi(s,a) \cdot \frac{\pi(s,a)}{\pi(s,a)} \cdot \nabla \pi(s,a).$$

$$= \sum_s \mu(s) \sum_a \pi(s,a) \left[q_\pi(s,a) \cdot \nabla \log \pi(s,a) \right].$$

$$= \mathbb{E}_{(s,a)} \left[q_\pi(s,a) \cdot \nabla \log \pi(s,a) \right]$$

$$\boxed{\theta_{t+1} \leftarrow \theta_t + \alpha \left[\hat{q}(s,a) \cdot \nabla \log \pi(s,a) \right].}$$

$\hookrightarrow$ final update rule.

Reinforce :-

$$\theta_{t+1} \leftarrow \theta_t + \alpha \left[\boxed{G_\tau} \nabla \log \pi(s,a) \right].$$

TD(0) :- approximation

↳ Actor-critic :-

$$Q(s,a) = E[(r + \gamma V^{*}(s')]$$

$$\theta_{t+1} \leftarrow \theta_t + \alpha \left[(r + \gamma V^{*}(s')) \nabla \log \pi(s,a) \right].$$

$$\boxed{\hat{V}(s') \approx \omega^T \psi(s).}$$ → on policy TD(0) approximation as above

$$\theta_{t+1} \leftarrow \theta_t + \alpha \left[(r + \gamma \omega^T \psi(s)) \nabla \log \pi(s,a) \right].$$

↳ final AC algorithm.

2 updates : ω : estimating v.f → critic
θ : improving policy → Actor.