Aprendizaje Reforzado

Maestría en Ciencia de Datos, DC - UBA

Julián Martínez
Javier Kreiner

Repaso

Un paso de evalución, uno de mejora

Sarsa (on-policy)

$$Q^{k+1}(S,A) = Q^k(S,A) + \alpha(R^+ + \gamma Q^k(S^+, A^+) - Q^k(S,A)),$$

Q-learning (off-policy)

$$Q^{k+1}(S,A) = Q^k(S,A) + \alpha(R^+ + \gamma \max_{a'} Q^k(S^+, a') - Q^k(S,A))$$

con S^+ proveniente de tomar la acción A^+ con la política $\pi_{k+1} = \varepsilon - greedy(Q^k)$.

Aproximación de función de valor

$$J(w) := E_\mu[(v_\pi(S) - \hat{v}(S;w))^2] = \sum_{s \in \mathcal{S}} \mu(s)(v_\pi(s) - \hat{v}(s;w))^2,$$

En lugar de calcular $v_\pi(s), \quad \forall s$, *aproximamos globalmente* controlando los parámetros w.

Recuerdo: Regresión Lineal

$$J(\beta) = E[(Y - f_\beta(X))^2] \approx \frac{1}{n} \sum_{i=1}^{n} (y_i - f_\beta(x_i))^2$$

El cual se puede minimizar realizando Descenso por Gradiente Estocástico (*Batch*)

Descenso por Gradiente Estocástico (SGD)

- Reemplazar una esperanza por una realización
- Reemplazar la función por el target

TARGET

$$w^{k+1} = w^k + \Delta w^{k+1}$$

$$\Delta w^{k+1} = \alpha(q_\pi(S_t, A_t) - \hat{q}(S_t, A_t; w))\nabla_w \hat{q}(S_t, A_t; w)$$

Sarsa- on-policy

$$\approx \alpha(R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) - \hat{q}(S_t, A_t; w))\nabla_w \hat{q}(S_t, A_t; w)$$

Q-learning- off-policy

$$\approx \alpha(R_{t+1} + \gamma \max_{a'} q_\pi(S_{t+1}, a') - \hat{q}(S_t, A_t; w))\nabla_w \hat{q}(S_t, A_t; w)$$

Podemos usar la experiencia que tengamos en más de una pasada de SGD!

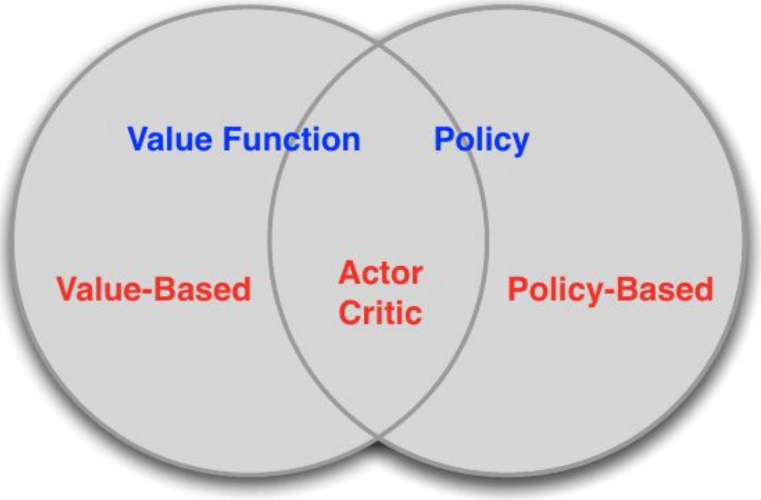
Episodic Semi-gradient Sarsa for Estimating $\hat{q} \approx q_*$

Input: a differentiable action-value function parameterization $\hat{q} : \mathcal{S} \times \mathcal{A} \times \mathbb{R}^d \to \mathbb{R}$
Algorithm parameters: step size $\alpha > 0$, small $\varepsilon > 0$
Initialize value-function weights $\mathbf{w} \in \mathbb{R}^d$ arbitrarily (e.g., $\mathbf{w} = \mathbf{0}$)

Loop for each episode:
  $S, A \leftarrow$ initial state and action of episode (e.g., ε-greedy)
  Loop for each step of episode:
    Take action A, observe R, S'
    If S' is terminal:
      $\mathbf{w} \leftarrow \mathbf{w} + \alpha\big[R - \hat{q}(S, A, \mathbf{w})\big]\nabla\hat{q}(S, A, \mathbf{w})$
      Go to next episode
    Choose A' as a function of $\hat{q}(S', \cdot, \mathbf{w})$ (e.g., ε-greedy)
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha\big[R + \gamma\hat{q}(S', A', \mathbf{w}) - \hat{q}(S, A, \mathbf{w})\big]\nabla\hat{q}(S, A, \mathbf{w})$
    $S \leftarrow S'$
    $A \leftarrow A'$

Policy Gradient

Optimizamos *directamente* sobre la política.



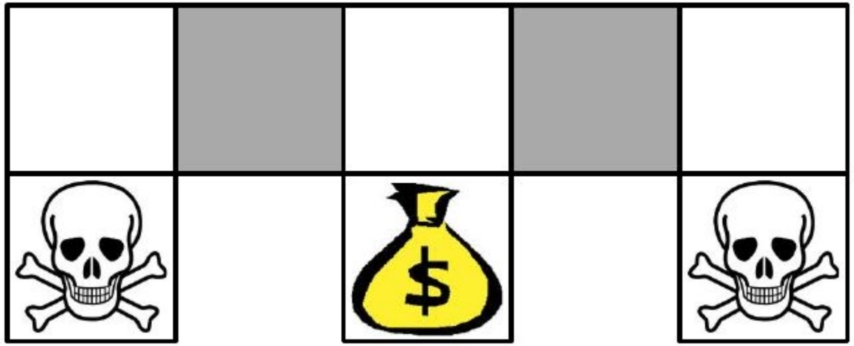
Modelizamos $\pi(a|s,\theta)$ *como una distribución* parametrizada por θ.

Recordar: Regresión Logística

- Puede aproximar más rápidamente una política determinística.
- En algunos contextos, la política óptima *puede no ser determinística*.

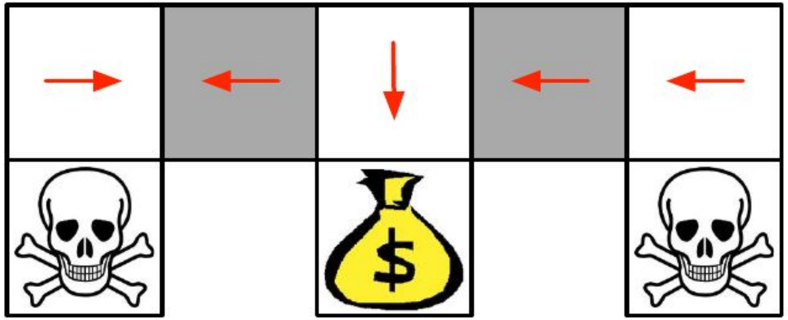
Aliased Gridworld

- Los sitios grises tienen iguales condiciones.
- Supongamos que sólo tengo *información parcial* del ambiente.

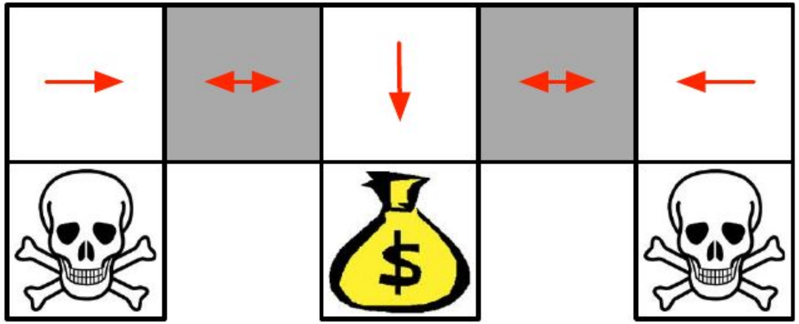


$$\phi(s, a) = \mathbf{1}(\text{wall to N}, a = \text{move E})$$

Aproximando la función de estado-acción (determinística)



Aproximando la política (estocástica)



Diferentes posibilidades

$$\pi(a|s,\boldsymbol{\theta}) \doteq \frac{e^{h(s,a,\boldsymbol{\theta})}}{\sum_b e^{h(s,b,\boldsymbol{\theta})}},$$

$$h(s,a,\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathbf{x}(s,a),$$

Softmax

$$\pi(a|s,\boldsymbol{\theta}) \doteq \frac{1}{\sigma(s,\boldsymbol{\theta})\sqrt{2\pi}} \exp\left(-\frac{(a-\mu(s,\boldsymbol{\theta}))^2}{2\sigma(s,\boldsymbol{\theta})^2}\right),$$

Gaussianas (acciones continuas)

Funciones objetivo

- Tareas episódicas:

$$J_0(\theta) = v_{\pi_\theta}(s_0),$$

- Tareas continuas:

$$J_{avV}(\theta) = \sum_s d^{\pi_\theta}(s) v_{\pi_\theta}(s),$$

- Recompensa promedio en un paso:

$$J_{avR}(\theta) = \sum_s d^{\pi_\theta}(s) \sum_a \pi_\theta(s, a) \mathcal{R}_s^a,$$

con d^{π_θ} la distribución invariante de $S_t | A_0, \ldots, A_{t-1}$.

Policy Gradient Theorem - Caso simple

$$J(\theta) = \sum_s d(s) E_{\pi_\theta}[R_1 | S_0 = s] = \sum_s d(s) \sum_a \pi_\theta(s,a) \mathcal{R}_s^a,$$

$$\begin{aligned}\nabla_\theta \pi_\theta(s,a) &= \pi_\theta(s,a) \frac{\nabla_\theta \pi_\theta(s,a)}{\pi_\theta(s,a)} \\ &= \pi_\theta(s,a) \nabla_\theta \log \pi_\theta(s,a)\end{aligned}$$

$$\begin{aligned}\nabla J(\theta) &= \sum_s d(s) \sum_a \nabla \pi_\theta(s,a) \mathcal{R}_s^a, \\ &= \sum_s d(s) \sum_a \pi_\theta(s,a) \nabla \log \pi_\theta(s,a) \mathcal{R}_s^a, \\ &= \sum_s d(s) E_{\pi_\theta}[\nabla \log \pi_\theta(S_0, A_0) \; R_1 | S_0 = s].\end{aligned}$$

Policy Gradient Theorem

$$J(\theta) = E_d[E_{\pi_\theta}[q_{\pi_\theta}(S_0, A_0)|S_0]],$$

entonces

$$\nabla_\theta J(\theta) = E_d[E_{\pi_\theta}[\nabla_\theta \log \pi_\theta(S_0, A_0) q_{\pi_\theta}(S_0, A_0)|S_0]],$$

Me da una expresión explícita para calcular el gradiente *en términos de los ingredientes que sí puedo aproximar!*

Ejemplos de función score

$$\pi_\theta(s,a) \propto e^{\phi(s,a)^\top \theta}$$

$$\nabla_\theta \log \pi_\theta(s,a) = \phi(s,a) - \mathbb{E}_{\pi_\theta}\left[\phi(s,\cdot)\right]$$

$$\pi(a|s,\boldsymbol{\theta}) \doteq \frac{1}{\sigma(s,\boldsymbol{\theta})\sqrt{2\pi}} \exp\left(-\frac{(a-\mu(s,\boldsymbol{\theta}))^2}{2\sigma(s,\boldsymbol{\theta})^2}\right),$$

$$\nabla_\theta \log \pi_\theta(s,a) = \frac{(a-\mu(s))\phi(s)}{\sigma^2}$$

REINFORCE - MC Policy Gradient

REINFORCE: Monte-Carlo Policy-Gradient Control (episodic) for π_*

Input: a differentiable policy parameterization $\pi(a|s, \boldsymbol{\theta})$
Algorithm parameter: step size $\alpha > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):
  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \boldsymbol{\theta})$
  Loop for each step of the episode $t = 0, 1, \ldots, T-1$:
    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ $\quad (G_t)$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \gamma^t G \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta})$

Algoritmo Actor-Crítico

MC Policy Gradient tiene mucha varianza. ¿Cómo reducirla?

- crítico: Evalua la política.

$$\hat{q}(s, a; w) \approx q_{\pi_0}(s, a)$$

- actor: Actualiza los parámetros de la política siguiendo las sugerencias del crítico.

$$\Delta\theta^{k+1} = \alpha \nabla_{\theta^k} \log \pi_{\theta^k}(s, a) \hat{q}(s, a; w)$$

- Simple actor-critic algorithm based on action-value critic
- Using linear value fn approx. $Q_w(s,a) = \phi(s,a)^\top w$

  Critic Updates w by linear TD(0)

  Actor Updates θ by policy gradient

function QAC

    Initialise s, θ
    Sample $a \sim \pi_\theta$
    for each step **do**
        Sample reward $r = \mathcal{R}_s^a$; sample transition $s' \sim \mathcal{P}_{s,\cdot}^a$
        Sample action $a' \sim \pi_\theta(s', a')$
        $\delta = r + \gamma Q_w(s', a') - Q_w(s,a)$
        $\theta = \theta + \alpha \nabla_\theta \log \pi_\theta(s,a) Q_w(s,a)$
        $w \leftarrow w + \beta \delta \phi(s,a)$
        $a \leftarrow a', s \leftarrow s'$
    end for

end function

Sesgo en Actor-Crítico

$$\hat{q}(s, a; w) \approx q_{\pi_0}(s, a)$$

Introduce un sesgo en el gradiente.
¿Cómo controlar cuánto afecta en la actualización de la política?

$$\nabla_w Q_w(s, a) = \nabla_\theta \log \pi_\theta(s, a)$$

Compatible

Minimiza el EM

$$\mathbb{E}_{\pi_\theta}\left[\left(Q^{\pi_\theta}(s, a) - Q_w(s, a)\right)^2\right]$$

El gradiente es exacto

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s, a)\ Q_w(s, a)\right]$$

Reduciendo la varianza de MC Policy Gradient

$$\mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a) B(s)\right] = \sum_{s\in\mathcal{S}} d^{\pi_\theta}(s) \sum_a \nabla_\theta \pi_\theta(s,a) B(s)$$
$$= \sum_{s\in\mathcal{S}} d^{\pi_\theta} B(s) \nabla_\theta \sum_{a\in\mathcal{A}} \pi_\theta(s,a)$$
$$= 0$$

$$B(s) = V^{\pi_\theta}(s)$$

Restarle una *Baseline* puede reducir la varianza **sin modificar el gradiente.**

$$A^{\pi_\theta}(s,a) = Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s)$$
$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a)\ A^{\pi_\theta}(s,a)\right]$$

Más parámetros...

Actualizar las dos funciones de valor usando TD

$$V_v(s) \approx V^{\pi_\theta}(s)$$

$$Q_w(s,a) \approx Q^{\pi_\theta}(s,a)$$

$$A(s,a) = Q_w(s,a) - V_v(s)$$

Función de Ventaja

Otra posibilidad

$$\begin{aligned} A^{\pi_\theta}(s,a) &\approx r^{+} + \gamma V^{\pi_\theta}(s^{+}) - V^{\pi_\theta}(s) \\ &\approx r^{+} + \gamma V_v(s^{+}) - V_v(s) \end{aligned}$$

- For the true value function $V^{\pi_\theta}(s)$, the TD error δ^{π_θ}

$$\delta^{\pi_\theta} = r + \gamma V^{\pi_\theta}(s') - V^{\pi_\theta}(s)$$

- is an unbiased estimate of the advantage function

$$\begin{aligned}
\mathbb{E}_{\pi_\theta}\left[\delta^{\pi_\theta} | s, a\right] &= \mathbb{E}_{\pi_\theta}\left[r + \gamma V^{\pi_\theta}(s') | s, a\right] - V^{\pi_\theta}(s) \\
&= Q^{\pi_\theta}(s, a) - V^{\pi_\theta}(s) \\
&= A^{\pi_\theta}(s, a)
\end{aligned}$$

- So we can use the TD error to compute the policy gradient

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s, a)\ \delta^{\pi_\theta}\right]$$

- In practice we can use an approximate TD error

$$\delta_v = r + \gamma V_v(s') - V_v(s)$$

- This approach only requires one set of critic parameters v