Aprendizaje Reforzado

Maestría en Ciencia de Datos, DC - UBA

Julián Martínez
Javier Kreiner

El espacio de estados...

- El espacio de estados puede ser *gigante*: Backgammon - 10^{20} estados.
- Espacio de estados *continuo*.

⚠ Difícil o imposible guardar $v_\pi(s)$ para todo s!

Idea:

$$v_\pi(s) \approx \hat{v}(s; w)$$

Diferentes aproximantes

- Combinación lineal de features.
- Redes neuronales
- Fourier

En general, varias de las herramientas vistas en supervisado.

A tener en cuenta:
diferenciabilidad y datos no iid.

Gradiente Descendente Estocástico

Busco w tal que

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

sea mínimo (μ distribución sobre $\mathcal{S}$).

$$\nabla_w J(w) = -2E_\mu[(v_\pi(S) - \hat{v}(S;w))\nabla_w \hat{v}(S;w)]$$

Stochastic Gradient Descent

$$\begin{aligned} w^{k+1} &= w^k + \Delta w^{k+1} \\ &= w^k + \alpha(v_\pi(S) - \hat{v}(S;w^k))\nabla_w \hat{v}(S;w^k), \end{aligned}$$

$S \sim \mu.$

Aproximación lineal (en los features)

$$\mathbf{x}(S) = \begin{pmatrix} \mathbf{x}_1(S) \\ \vdots \\ \mathbf{x}_n(S) \end{pmatrix}$$

A veces trabajamos con sólo algunos atributos del estado del ambiente.

Regresión Lineal

$$\hat{v}(S, \mathbf{w}) = \mathbf{x}(S)^\top \mathbf{w} = \sum_{j=1}^{n} \mathbf{x}_j(S)\mathbf{w}_j$$

$$\Delta \mathbf{w} = \alpha(v_\pi(S) - \hat{v}(S, \mathbf{w}))\mathbf{x}(S)$$

Actualizo solo los features en los cuales estoy interesado

Volviendo a la realidad

En general, **no tengo la función de valor**

Actualización Monte Carlo

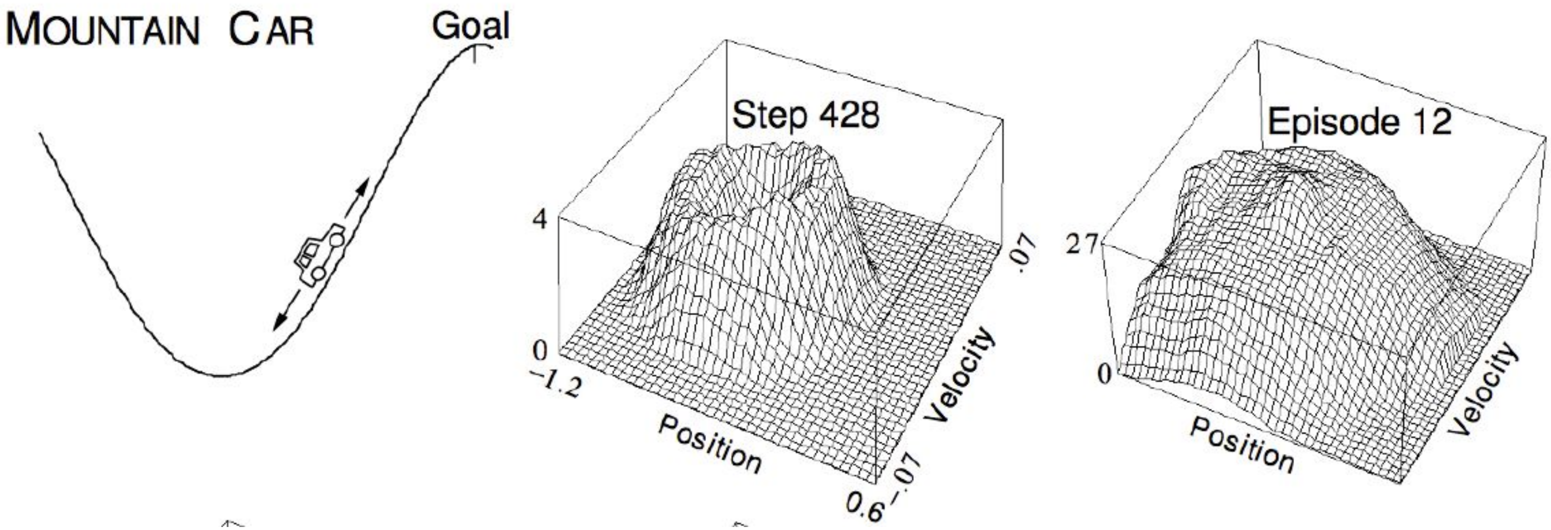
$$\Delta \mathbf{w} = \alpha(G_t - \hat{v}(S_t, \mathbf{w}))\nabla_{\mathbf{w}}\hat{v}(S_t, \mathbf{w})$$

Actualización Diferencias Temporales

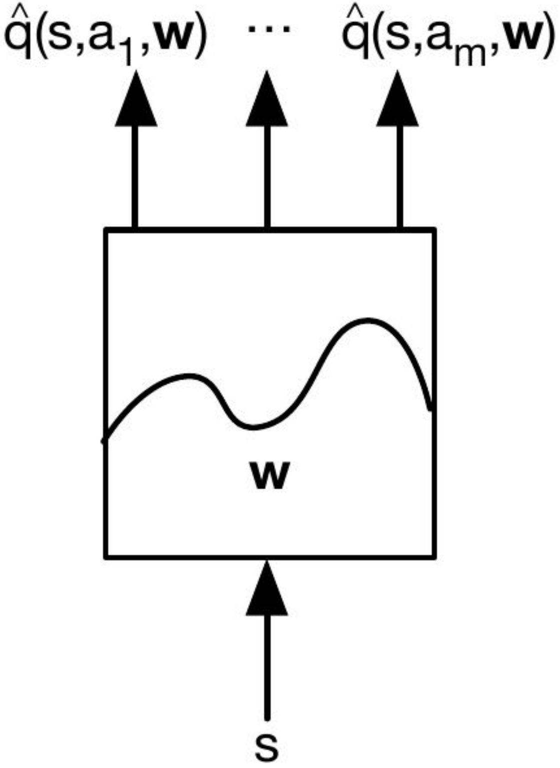
$$\Delta \mathbf{w} = \alpha(R_{t+1} + \gamma\hat{v}(S_{t+1}, \mathbf{w}) - \hat{v}(S_t, \mathbf{w}))\nabla_{\mathbf{w}}\hat{v}(S_t, \mathbf{w})$$

Aproximación de función de estado-acción

Reemplazo $\hat{v}(s; w)$ por $\hat{q}(s, a; w)$.



Otra variante



Aproximar la función de estado-acción para *varias acciones* al mismo tiempo.

Predicción por cuadrados mínimos

En cada paso, utilizamos *sólo una vez* la experiencia observada.

Tenemos experiencia acumulada

$$\mathcal{D} = \{\langle s_1, v_1^\pi \rangle, \langle s_2, v_2^\pi \rangle, ..., \langle s_T, v_T^\pi \rangle\}$$

Minimizamos esta función de pérdida

$$LS(\mathbf{w}) = \sum_{t=1}^{T} (v_t^\pi - \hat{v}(s_t, \mathbf{w}))^2$$

SGD con Replay

Tomo una muestra al azar de la observada con anterioridad

$$\langle s, v^{\pi} \rangle \sim \mathcal{D}$$

Actualizo con SGD

$$\Delta \mathbf{w} = \alpha(v^{\pi} - \hat{v}(s, \mathbf{w}))\nabla_{\mathbf{w}} \hat{v}(s, \mathbf{w})$$

Converge a

$$\mathbf{w}^{\pi} = \underset{\mathbf{w}}{\operatorname{argmin}}\ LS(\mathbf{w})$$

Recordemos Q-learning

Dada $Q^k(s,a)$:

$$\pi_{k+1}(s) = \arg\max_{a'} Q^k(S_t, a'), \qquad \mu_{k+1}(a|s) = \pi^{\varepsilon}_{k+1}.$$

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