Tema 5.1: Aproximación de la función de valor

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS

Nota previa

- Vamos a ponernos con el RL de verdad
- Predicción y control con aproximación de las funciones de valor es un tema clásico en teoría de RL
- Por lo tanto, nos apoyamos en dos recursos “clásicos”: el libro de Sutton & Barto, y las diapositivas sobre aproximación de la función de valor de David Silver para UCL.

RL a gran escala

- Resolver problemas con espacios de estado muy grandes
- Ejemplos: Ajedrez, Go, navegación de vehículos autónomos, finanzas...

Aproximación de Funciones de Valor

- Problema: Los métodos tabulares son ineficientes (no escalan bien)
  - El problema no cabe en memoria
  - Hace falta demasiada experiencia para entrenar a un agente
- Problema: MDPs grandes con muchos estados y acciones.
- Solución: Aproximación de funciones $\hat{v}(s, w)$ o $\hat{q}(s, a, w)$

Tipos de Aproximación de Funciones

- Combinaciones lineales de características.
- Redes neuronales.
- Árboles de decisión, vecinos más cercanos.
- Bases de Fourier o Wavelet.
- Enfoque en aproximadores diferenciables.

Descenso del gradiente

- Método bien conocido por ser el estándar en entrenamiento de modelos de DL.
- Sea J(w) una función vectorial derivable, cuyo gradiente es:

$$\nabla_w J(w) = \left[\frac{\partial J(w)}{\partial w_1}, \ldots, \frac{\partial J(w)}{\partial w_n}\right]$$

- Podemos llegar a un mínimo local moviéndonos iterativamente en la dirección de máxima reducción de J desde un determinado punto:

$$\Delta w = -\frac{1}{2}\alpha \nabla_w J(w)$$

Aproximación con descenso del gradiente

- En nuestro problema J es:

$$J(w) = \mathbb{E}_{\pi}\left[\left(v_{\pi}(S) - \hat{v}(S,w)\right)^2\right]$$

- Para la que la regla de actualización de los pesos va a ser:

$$\Delta w = -\frac{1}{2}\alpha\nabla_w J(w) = \alpha\mathbb{E}_{\pi}\left[\left(v_{\pi}(S) - \hat{v}(S,w)\right)\nabla_w\hat{v}(S,w)\right]$$

- Pero vamos a moverlos muestra a muestra

$$\Delta w = \alpha\left(v_{\pi}(S) - \hat{v}(S,w)\right)\nabla_w\hat{v}(S,w)$$

Representación del estado

- ¿Qué es el estado?

- ¿Qué tiene que cumplir una señal que nos trasmita información sobre el estado?

- ¿Qué alternativas tenemos para “construir” esa señal?

- ¿Cómo vamos a llamarla? (Ejemplos)

Aproximación Lineal de Funciones de Valor

- Representación del estado con un vector de características x(s) ➔ Aproximación lineal:

$$\hat{v}(s, w) = x(s)^T w = \sum_j x_j(s) w_j$$

- Función de coste de DG cuadrática:

$$J(w) = \mathbb{E}_\pi \left[\left(v_\pi(s) - x(s)^T w \right)^2 \right]$$

- Regla de actualización de pesos:

Relación con caso tabular

- Caso especial de aproximación lineal.
- El vector de características es una codificación *one-hot* de los estados.
- w almacenaría los valores particulares de cada estado.
- ¿Consideras que hay algún tipo de aproximación?

Métodos Incrementales para la Predicción

- Hasta ahora, las fórmulas han contenido el valor real al que hay que aproximarse.
- Pero RL **NO** es aprendizaje supervisado. Sólo disponemos de la recompensa para guiar el aprendizaje.
- ¿Cuáles serán las fórmulas de actualización de pesos en la práctica para MC y TD?

Predicción MC con aproximación lineal

- El retorno G_t es un estimador no sesgado pero ruidoso del valor real del estado
- La interacción con el entorno nos va a dar muestras para entrenar usando DG estocástico:

$$\langle S_1, G_1 \rangle, \langle S_2, G_2 \rangle, \ldots, \langle S_T, G_T \rangle$$

- La predicción MC converge al óptimo global en el caso lineal, y es robusta frente al tipo de función con la que aproximemos el valor de estado.

Predicción TD con aproximación lineal

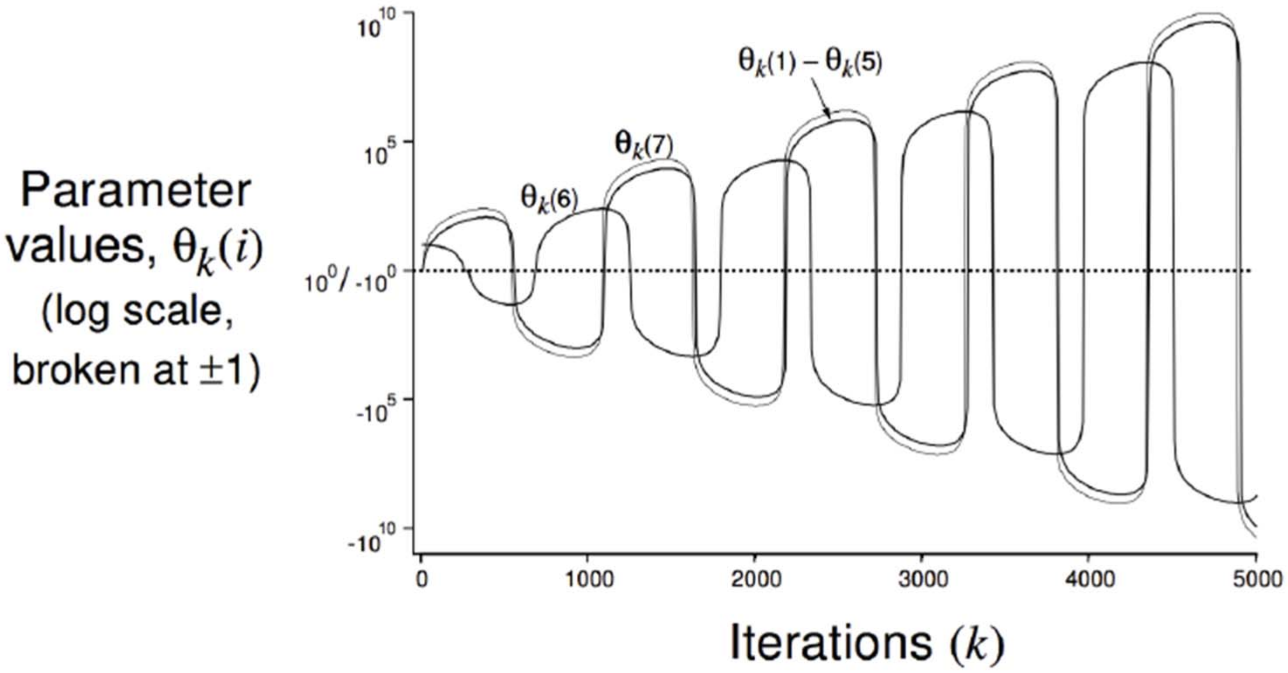
- El *TD-target* es un estimador sesgado (menos ruidoso que Gt) del valor real del estado

- La interacción con el entorno nos va a dar muestras para entrenar usando DG estocástico. ¿Qué pinta tienen?

- La predicción TD(0) converge a un punto cercano al óptimo global. Además, los algoritmos de predicción TD se pueden "romper" (sus pesos pueden divergir).

Consideraciones sobre convergencia

- Evolución de los pesos en el contraejemplo de Baird (predicción TD)

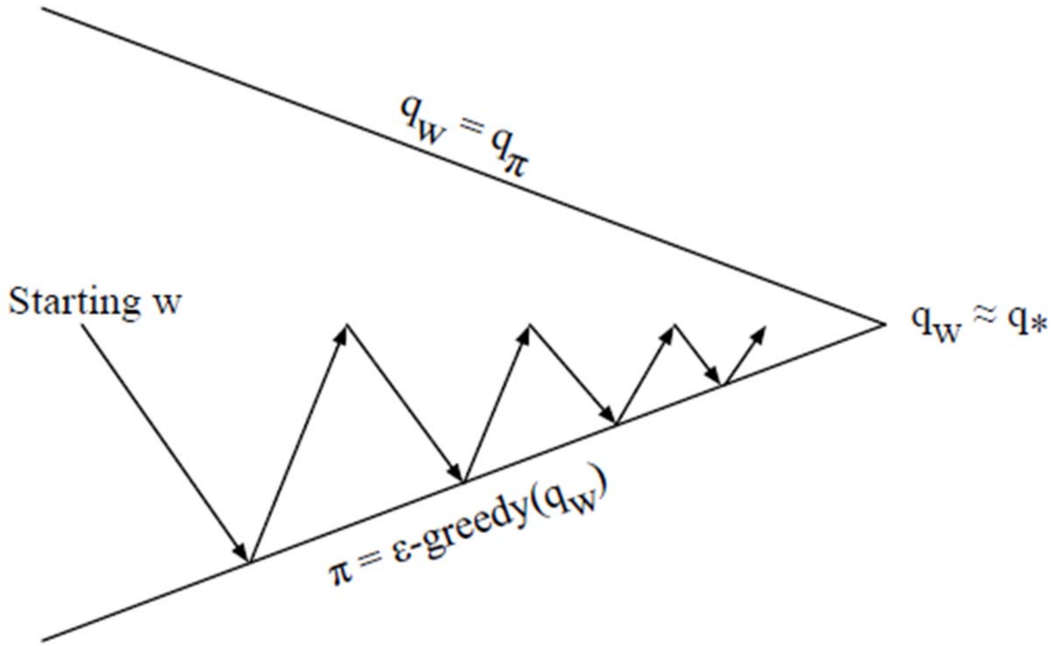


Consideraciones sobre convergencia (predicción)

Tipo de búsqueda	Algoritmo	Tabular	Lineal	No lineal
On-policy	MC	SÍ	SÍ	SÍ
	TD(0)	SÍ	SÍ	**NO**
Off-policy	MC	SÍ	SÍ	**NO**
	TD(0)	SÍ	**NO**	**NO**

- **Inductores de inestabilidad**
  - Aproximación de función de valor
  - *Bootstrapping*
  - *Off-policy*

CONTROL con aproximación



Paso de v a q

• ¿Cuáles serían las expresiones de la función aproximada, la de coste y la regla de actualización de pesos para aproximar q en lugar de v?

SARSA con Aproximación

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)
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha\big[R + \gamma\hat{q}(S', A', \mathbf{w}) - \hat{q}(S, A, \mathbf{w})\big]\nabla\hat{q}(S, A, \mathbf{w})$
    $S \leftarrow S'$
    $A \leftarrow A'$

Q-learning con aproximación

- Identifica las líneas que difieren de SARSA y define cómo serían para q-learning

Consideraciones sobre convergencia (control)

Algoritmo	Tabular	Lineal	No lineal
MC Control	SÍ	(SÍ)	**NO**
Sarsa	SÍ	(SÍ)	**NO**
Q-learning	SÍ	**NO**	**NO**

(SÍ) ➔ cerca, pero no quieto

- **Inductores de inestabilidad**
  - Aproximación de función de valor
  - *Bootstrapping*
  - *Off-policy*

Trabajo para práctica 3

- Leer sección 9.5.4 (tile coding) del Sutton & Barto
- Realizar los ejercicios con el código que os vamos a entregar la semana que viene