Aprendizaje Reforzado

Maestría en Data Mining, Universidad Austral

Javier Kreiner

Programación dinámica

- Conjunto de algoritmos para calcular políticas óptimas cuando tenemos un modelo perfecto del ambiente como un proceso de decisión de Markov
- Tienen limitada utilidad práctica, pero son muy útiles teóricamente
- Programación dinámica provee una base esencial para entender los otros métodos
- La idea de PD y otros métodos es usar la función de valor para organizar la búsqueda de políticas óptimas

Ecuación de Bellman

$$v(s) = \mathcal{R}_s + \gamma \sum_{s'} p_{s,s'} v(s')$$

$$v = (v(s_1), \ldots, v(s_K))$$

Un sistema de K ecuaciones!

- Con matrices: $\begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix} = \begin{bmatrix} \mathcal{R}_1 \\ \vdots \\ \mathcal{R}_n \end{bmatrix} + \gamma \begin{bmatrix} \mathcal{P}_{11} & \ldots & \mathcal{P}_{1n} \\ \vdots & & \\ \mathcal{P}_{11} & \ldots & \mathcal{P}_{nn} \end{bmatrix} \begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix}$

- Escrito en una línea: $v = \mathcal{R} + \gamma p v$

Recordar

Ecuaciones que relacionan las probabilidades de transición y las recompensas esperadas con una política fija:

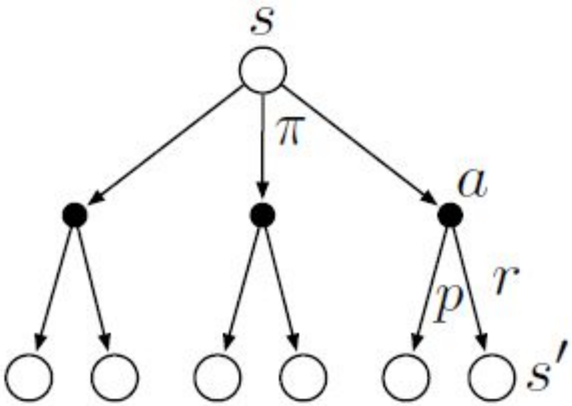
$$\mathcal{P}^{\pi}_{s,s'} = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{P}^{a}_{ss'}$$

$$\mathcal{R}^{\pi}_{s} = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{R}^{a}_{s}$$

Evaluación de Política

$$v_\pi(s) = \sum_a [\mathcal{R}_s^a + \gamma \sum_{s'} v_\pi(s') p_{s,s'}^a] \pi(a|s)$$

$$= \mathcal{R}_s^\pi + \gamma \sum_{s'} v_\pi(s') p_{s,s'}^\pi$$



Método Iterativo

$$v_\pi^{k+1}(s) = \mathcal{R}_s^\pi + \gamma \sum_{s'} v_\pi^k(s') p_{s,s'}^\pi$$

Algoritmo de evaluación de política

Iterative Policy Evaluation, for estimating $V \approx v_\pi$

Input π, the policy to be evaluated
Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
- $\Delta \leftarrow 0$
- Loop for each $s \in \mathcal{S}$:
  - $v \leftarrow V(s)$
  - $V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\big[r + \gamma V(s')\big]$
  - $\Delta \leftarrow \max(\Delta, |v - V(s)|)$

until $\Delta < \theta$

Programación

Función de Valor Óptima

$$v_*(s) = \max_{\pi} v_\pi(s)$$

$$q_*(s, a) = \max_{\pi} q_\pi(s, a)$$

Optimalidad de MDP

$$\exists\ \pi_* \ / \pi_* \geq \pi \ \forall\ \pi \qquad \text{such that}$$

$$v_*(s) = v_{\pi_*}(s), \qquad q_*(s,a) = q_{\pi_*}(s,a) \qquad \forall\ s, a$$

$$\pi_*(s) = arg\,\mathbf{max}_a\ q_*(s,a)$$

$$v_*(s) = \mathbf{max}_a\ q_*(s,a)$$

Ecuación de optimalidad de Bellman

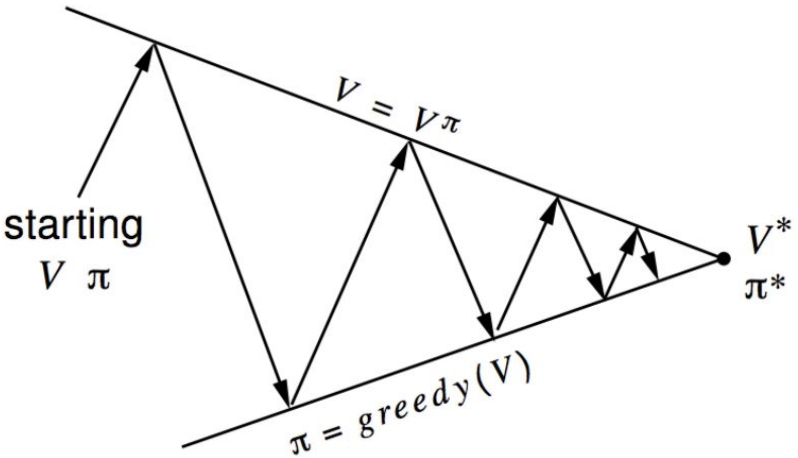
$$v_*(s) = \max_a \left[\mathcal{R}_s^a + \gamma \sum_{s'} p_{s,s'}^a v_*(s')\right]$$

Son ecuaciones NO lineales!

¿Cómo obtener v_* y π_*?

Evaluación y Mejora

$$v_\pi(s) = \sum_a [\mathcal{R}_s^a + \sum_{s'} v_\pi(s')p_{s,s'}^a]\pi(a|s)$$



Evaluation / Improvement

$$\pi_0 \xrightarrow{\mathrm{E}} v_{\pi_0} \xrightarrow{\mathrm{I}} \pi_1 \xrightarrow{\mathrm{E}} v_{\pi_1} \xrightarrow{\mathrm{I}} \pi_2 \xrightarrow{\mathrm{E}} \cdots \xrightarrow{\mathrm{I}} \pi_* \xrightarrow{\mathrm{E}} v_*,$$

$$q_{\pi_k}(s,a) = \mathcal{R}_s^a + \gamma \sum_{s'} v_{\pi_k(s')} p_{s,s'}^a$$

$$\pi_{k+1}(s) = arg\max_a q_{\pi_k}(s,a)$$

Evaluación y mejora

- Primero evaluamos la función de valor de la política actual
- Luego obtenemos la política greedy respecto de la función de valor actual
- Esta nueva política es estrictamente mejor que la anterior (si fueran iguales serían óptimas)
- Como el número de políticas es finito (conjunto de estados y acciones finitos), entonces este procedimiento debe terminar eventualmente

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Loop:
   - $\Delta \leftarrow 0$
   - Loop for each $s \in \mathcal{S}$:
     - $v \leftarrow V(s)$
     - $V(s) \leftarrow \sum_{s',r} p(s', r \,|\, s, \pi(s))\big[r + \gamma V(s')\big]$
     - $\Delta \leftarrow \max(\Delta, |v - V(s)|)$

   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   - *old-action* $\leftarrow \pi(s)$
   - $\pi(s) \leftarrow \operatorname{argmax}_a \sum_{s',r} p(s', r \,|\, s, a)\big[r + \gamma V(s')\big]$
   - If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*

   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

Iteración de Valor:

- Un solo paso de mejora y evaluación
- Se puede pensar como usar la ecuación de optimalidad de Bellman para el update:

$$v^{k+1}(s) = \max_a \left[\mathcal{R}_s^a + \gamma \sum_{s'} p_{s,s'}^a v^k(s')\right]$$

Otras variantes:

- Actualizar un sólo estado en cada iteración evaluation / improvement.
- Actualizar algunos estados en evaluation y otros en improvement.
- No actualizar los estados que sean poco probables.

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
| $\Delta \leftarrow 0$
| Loop for each $s \in \mathcal{S}$:
| $v \leftarrow V(s)$
| $V(s) \leftarrow \max_a \sum_{s',r} p(s', r \,|\, s, a)\big[r + \gamma V(s')\big]$
| $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that
  $\pi(s) = \arg\max_a \sum_{s',r} p(s', r \,|\, s, a)\big[r + \gamma V(s')\big]$

Eficiencia de Programación Dinámica

- Complejidad polinómica en n y k, donde n es el número de estados y k el número de acciones
- Como el número de políticas determinísticas es k^n, es exponencialmente mejor que búsqueda directa
- En la práctica se pueden resolver problemas con millones de estados

Programación

Ejercicio

- Problema del apostador

Lectura recomendada

- AlphaStar de deepmind le gana a profesionales del Starcraft 2: https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/
- hilo de twitter con aplicaciones de RL: https://twitter.com/jackclarkSF/status/919584404472602624