Métodos de Montecarlo

Alvaro J. Riascos Villegas
Universidad de los Andes y Quantil

Noviembre de 2024

Contenido

Introducción

- Usualmente no conocemos el MDP (ambiente) con exactitud o puede ser muy dificil hacerlo (e.g., juego de 21) Sin embargo, es posible muestrear de tuplas de estados, acciones y recompensas **observados** o **simulados** de la interacción con el medio ambiente.
- En ambos casos la idea es aprender de la experiencia.
- En el caso de una interacción directa con el medio ambiente, se aprende de explotar y explorar (como en el caso de un bandido multiarmado).
- En el caso de una interacción simulada, se utiliza un modelo para generar estados futuros pero no es necesario conocer la distribucion completa de estados y recompensa (i.e., p(s',r')).

Introducción

- Vamos a seguir una estrategia de estimación de funciones valor y política inspiradas en PD.
- Para esto lo primero es aprender a evaluar la función valor del estado y acción.
- MC usa episodios completos para aprender. Se actualiza una vez termina el episodio.
- No hace bootstrapping. En PD se actualiza la estimación de la función valor en un estado utilizando estimaciones de la misma función en todos los estados.
- En MC las estimaciones en cada estado son independientes.
- Dos versiones: *on policy* y *off policy*. La primera se basa en aprender de su propia interacción con el ambiente (i.e., mejor estimación de la política). La segunda, de observar otra política.

Contenido

Primera Visita Predicción de MC

First-visit MC prediction, for estimating $V \approx v_\pi$

Input: a policy π to be evaluated

Initialize:

  $V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$

  $Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):

  Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$

  $G \leftarrow 0$

  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:

    $G \leftarrow \gamma G + R_{t+1}$

    Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:

      Append G to $Returns(S_t)$

      $V(S_t) \leftarrow \text{average}(Returns(S_t))$

- Otra alternativa es usar todas las visitas. En este caso se usa el mismo pseudo-algoritmo excepto cuando se verifica si S_t ya ocurrio entre los estados anteriores.

Primera Visita Predicción de MC

- Calcular el promedio de los retornos y mantenerlos en memoria es ineficiente.
- Se puede modificar el algoritmo para hacer la actualización de forma incremental y más adecuada para entornos no estacionarios.
- Lo denominamos α - Montecarlo predicción de la función política:

$$V(S_t) \leftarrow V(S_t) + \alpha(G_t - V(S_t)) \tag{1}$$

Ejemplo: El Juego de 21

- El centro (dealer) utiliza una política fija: para cuando la cartas suman 17 o mayor. Caso contrario, continua.
- Si el centro se pasa de 21 pierde. De lo contrario el resultado depende de quien este más cerca de 21 (puede haber un empate).
- El jugador toma su decisión con base a tres variables: la suma de sus cartas, la carta que muestra el centro (un As es 10) y si el tiene o no un As usable (en total 200 estados).
- La política del jugador es parar si sus cartas suman 20 o 21. De lo contrario, continuar.

Ejemplo: El Juego de 21

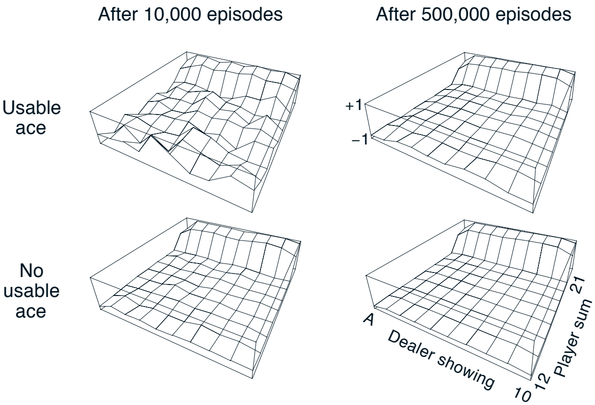


Figure 5.1: Approximate state-value functions for the blackjack policy that sticks only on 20 or 21, computed by Monte Carlo policy evaluation. ■

Contenido

Primera Visita Función Valor del Estado y la Acción

- Cuando se tiene un modelo se puede estimar el mejoramiento de una función de política simplemente estimando una acción de forma codiciosa. **Esto no es posible cuando no tenemos un modelo**.
- La alternativa es estimar la función valor de la acción.
- La estrategia es la misma del algoritmo de Primera Visita Predicción de MC. En este caso lo que interesa es las visitas a parejas (s, a).

Primera Visita Función Valor del Estado y la Acción

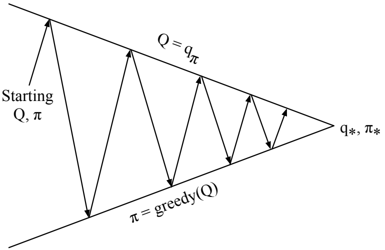
- Un problema relevante es que no se visiten todas las acciones. Por ejemplo, si la política es determinística. Dos formas de atacar el problema:
  1. Se eligen de forma aleatoria parejas estado-acción para iniciar el algoritmo.
  2. Se usan solamente funciones de política que le dan probabilidad positiva a todas las acciones (se introducen más adelante).

Contenido

Función de Política Óptima de MC

- La idea es iterar sobre la estimación de la función valor de la acción y el mejoramiento de la política.

$$\pi_0 \xrightarrow{\text{E}} q_{\pi_0} \xrightarrow{\text{I}} \pi_1 \xrightarrow{\text{E}} q_{\pi_1} \xrightarrow{\text{I}} \pi_2 \xrightarrow{\text{E}} \cdots \xrightarrow{\text{I}} \pi_* \xrightarrow{\text{E}} q_*,$$



Policy evaluation Monte-Carlo policy evaluation, $Q = q_\pi$

Función de Política Óptima de MC

- Para mitigar el problema de que algunas acciones no se visitan infinitas veces, usamos la exploración aleatoria de parejas estados acción de inicio del algoritmo.
- Esta estrategia puede no ser viable en particular cuando la estimación está basada en la interacción con el ambiente.
- Alternativamente podemos hacer una mejora ϵ-codiciosa.

Exploración Inicialización Algortimo Primera Visita

Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

Initialize:
$\quad \pi(s) \in \mathcal{A}(s)$ (arbitrarily), for all $s \in \mathcal{S}$
$\quad Q(s, a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$
$\quad Returns(s, a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Loop forever (for each episode):
$\quad$ Choose $S_0 \in \mathcal{S}$, $A_0 \in \mathcal{A}(S_0)$ randomly such that all pairs have probability > 0
$\quad$ Generate an episode from S_0, A_0, following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
$\quad G \leftarrow 0$
$\quad$ Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
$\quad\quad G \leftarrow \gamma G + R_{t+1}$
$\quad\quad$ Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:
$\quad\quad\quad$ Append G to $Returns(S_t, A_t)$
$\quad\quad\quad Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$
$\quad\quad\quad \pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

Exploración Inicialización Algortimo Primera Visita

- Obsérvese que este algoritmo es ineficiente al tener que guardar todas las recompensas durante un episodio para después sacar el promedio.
- Se puede hacer una versión con actualizacione incrementales.
- Dada un par estado accion, obsérvese que se promedian los retornos de funciones de política distintas.

Ejemplo: Juego de 21

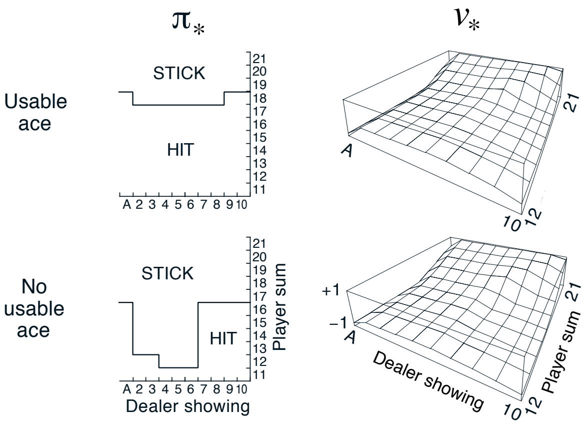
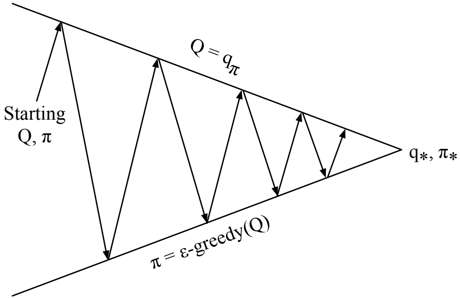


Figure 5.2: The optimal policy and state-value function for blackjack, found by Monte Carlo ES. The state-value function shown was computed from the action-value function found by Monte Carlo ES. ■

ϵ-Suave

- Si la exploracion de parejas de inicio no es viable la alternativa es usar funciones de política que con probablidad positiva elijan todas las acciones (véase el siguiente algoritmo).



Policy evaluation Monte-Carlo policy evaluation, $Q = q_\pi$

Policy improvement ϵ-greedy policy improvement

Primera Visita ϵ-Suave

On-policy first-visit MC control (for ε-soft policies), estimates $\pi \approx \pi_*$

Algorithm parameter: small $\varepsilon > 0$

Initialize:

$\quad \pi \leftarrow$ an arbitrary ε-soft policy

$\quad Q(s,a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

$\quad Returns(s,a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Repeat forever (for each episode):

$\quad$ Generate an episode following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$

$\quad G \leftarrow 0$

$\quad$ Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:

$\quad\quad G \leftarrow \gamma G + R_{t+1}$

$\quad\quad$ Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:

$\quad\quad\quad$ Append G to $Returns(S_t, A_t)$

$\quad\quad\quad Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$

$\quad\quad\quad A^* \leftarrow \arg\max_a Q(S_t, a)$ (with ties broken arbitrarily)

$\quad\quad\quad$ For all $a \in \mathcal{A}(S_t)$:

$$\pi(a|S_t) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(S_t)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(S_t)| & \text{if } a \neq A^* \end{cases}$$

Primera Visita ϵ-Suave

- Se llama ϵ-Suave porque $\pi(a \mid s) \geq \frac{\epsilon}{A(s)}$.
- Son un ejemplo de políticas ϵ-codiciosas.

Contenido

Predicción y Control Off-Policy

- Hasta ahora todos los algoritmos utilizados utilizan la función de política corriente para simular datos y mejorarla. Se llaman on-policy.
- Los algoritmos off-policy usan una función politica externa para ayudarle al algoritmo a aprender.

Contenido

Resumen

- Programación dinámica:
    - Se conoce el proceso Markoviano.
    - Bootstrapping: se actualiza la función valor con base en estimaciones de la función (futuras).
- Montecarlo
    - No se conoce el proceso Markoviano.
    - Se aprende de la interacción o simulación.
    - Se puede enfocar en una región de estados, sin actualizar la función valor en todo el espacio.
    - No hace boostrapping. Promedia retornos observados de la interaccion con el ambiente comenzando de un estado (acción). Esto le permite adaptarse mejor a procesos no Markovianos.