روش های مونت کارلو (۲ جلسه)

یادگیری تقویتی در [illegible]

بعد از دانستن ... تعیین ارزش حالت ها با Monte Carlo ، 3- تعیین ارزش حالت عمل ها با Monte Carlo ، 4- On-Policy Monte Carlo Control

و در نهایت / Off-Policy Monte Carlo Control

اینجا Model free کار می کنیم و به دینامیک نیاز نداریم و تنها با داشتن چند تا episode و تجربه عملی می توانیم ارزش ها را تخمین بزنیم و سیاست ها را مشخص کنیم.

- ★ فرض اساسی در برنامه ریزی پویا **داشتن مدل کامل MDP** است. $p(s', r \mid s, a)$
- ★ در عمل در بسیاری از مسائل یادگیری تقویتی، دینامیک مشخص نیست.
- ★ در **روش های Monte Carlo** بدون نیاز به دینامیک ارزش ها و سیاست ها تخمین زده خواهند شد.
- ★ روش های MC تنها از طریق تجربه و تعامل با محیط یادگیری را انجام می بخشند.
- ★ تجربه می تواند ناشی از محیط واقعی باشد یا محیط شبیه سازی
- ★ در برخی کاربردها که می توان دینامیک را تعیین کرد ولی به دلیل **محاسبات ریاضی سنگین** در برنامه ریزی پویا کاربردی ندارند. مثل شطرنج 10^{50} حالت.
- ★ در برخی کاربردها هم توصیف ریاضی دینامیک مسئله شدنی نیست.
- ★ روش های MC از نمونه برداری تصادفی جهت محاسبات ریاضی استفاده می کنند.
- ★ شبیه سازی سیستم ها و محاسبات عددی مهمترین کاربردهای MC است

که شامل انتگرال $\int_a^b h(x)\,dx$

و فرض کنید $h(x) = e^{\sin x}$

$$w(x) = h(x)\cdot(b-a)$$

$$f(x) = \frac{1}{b-a} \longrightarrow U[a \quad b]$$

تابع چگالی احتمال یکنواخت

$$\int_a^b h(x)\,dx = \int_a^b w(x) f(x)\,dx \longrightarrow$$

استفاده از $w(x)$ برای تابع چگالی احتمال $f(x)$

یادآوری: $E(x) = \int x f(x)\,dx \longrightarrow E[g(x)] = \int_x g(x) f(x)\,dx$

$$x_1, \dots, x_{10^6} \in U[a \quad b]$$

برای محاسبه انتگرال از توزیع یکنواخت استفاده می کنیم و از توزیع یکنواخت ۱۰ بار نمونه برداری می کنیم و یک مجموعه تک دیتا به دست می آید که عدد تصادفی را وارد $w(x)$ می کنیم یک مجموعه دیگر به دست می آید و میانگین آن مجموعه دیتا برابر با مقدار انتگرال است.

s.a.m

CS CamScanner

تخمین ارزش حالت ها با Monte Carlo:

* روش های مونت کارلو اغلب برای محیط های دوره ای کاربرد دارند.
* بعد از پایان هر دوره، از روی بازده ها ارزش حالت ها و حالت عمل ها تخمین زده خواهد شد.

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \dots S_T \quad (A_0 \leftarrow \pi)$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1}$$

$$v_\pi(s) \doteq E_\pi[G_t \mid S_t = s] \qquad q_\pi(s,a) \doteq E_\pi[G_t \mid S_t = s, A_t = a]$$

(S_0, A_0)

* محاسبه بازگشت (Return) می تواند به دو روش First-visit, Every-visit انجام شود.
  - First-visit: که اولین دیدن یک state
  - Every-visit: هر دیدن از state در نظر گرفته می شود.

تعداد اپیزودها اگر زیاد شود ارزش ها به همان ارزش واقعی که می توانند از روی معادلات بلمن محاسبه شوند برسند.

نکته: هر محاسبه بازگشت ارزش ها است:

$$v_n(s) \longrightarrow v_{n+1}(s)$$

که n تعداد دوره هاست.

$$V_{n+1}(s) = V_n(s) + \frac{1}{n}\left(G_t(s) - V_n(s)\right)$$

در بندیت داشتیم ← $Q_{n+1} = Q_n + \frac{1}{n}(R_n - Q_n)$

$$Q_{n+1}(s,a) = Q_n(s,a) + \frac{1}{n}\left(G_t(s,a) - Q_n(s,a)\right)$$

الگوریتم First-Visit:

Input: a policy π to be evaluated

Initialize:

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

  Returns(s) ← an empty list, for all $s \in S$ [* * * ...]

Loop forever (for each episode):

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

  $G \leftarrow 0$ (بازگشت: $G_t = R_{t+1} + \gamma G_{t+1}$)

  Loop for each step of episode, $t = T-1, T-2, \dots, 0$: (برای محاسبه اش از آخر به اول می آییم.)

  s.a.m

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

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

Append G to Returns(S_t)

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

روش MC برای محیط‌های غیرایستا ← بجای $\frac{1}{n}$ از α استفاده کنیم

Constant α Monte Carlo

$$V(S_t) \leftarrow V(S_t) + \alpha [G_t - V(S_t)]$$

می‌خواهیم ترم دوم در صورت تغییر قواعد بازی بعد از 1000 قدم همچنان موثر باشد؛ $\frac{1}{n}$ تأثیرش کم می‌شود.

New Estimate ← Old Estimate + Step size [Target − Old Estimate]

← بعد از پایان هر دوره، ارزش حالت‌ها به‌روزرسانی خواهند شد.

تعیین ارزش حالت-عمل‌ها با Monte Carlo:

با داشتن ارزش حالت‌ها ($v_\pi(s)$) باید مدل MDP را هم داشته باشیم تا سیاست بهبود پیدا کند.

ولی با داشتن ارزش حالت-عمل‌ها به تنهایی می‌توان سیاست را بهبود داد.

$$\pi'(s) = \arg\max_a q_\pi(s,a)$$
$$= \arg\max_a E[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a]$$
$$= \arg\max_a \sum_{s',r} p(s',r \mid s,a)[r + \gamma v_\pi(s')]$$

$$q_\pi(s,a) = E[G_t \mid S_t = s, A_t = a]$$ ← اگر a هم در s انجام شد

$$v_\pi = E_\pi[G_t \mid S_t = s]$$ ولی بعد از آن از π تبعیت نماید.

مشابه Bandit!

ممکن است برخی زوج حالت-عمل‌ها در trajectory ظاهر نشوند.

اگر سیاست قطعی باشد. مثلاً ارزش یک عمل در هر حالت محاسبه می‌شود.

در هر state فقط یک action را می‌بینیم و action‌های دیگر چه می‌شود؟

راه‌حل: Exploring Starts

همه زوج حالت-عمل‌ها باید احتمال انتخاب داشته باشند.

راه‌های بهتر:

On-policy Monte Carlo

off policy Monte Carlo

s.a.m

CamScanner

Initialize:

$\pi_{(s)} \in A(s)$

الزيارة الأولى First Visit

s.a.m

CamScanner