



تمرین سری ششم

یادگیری ژرف

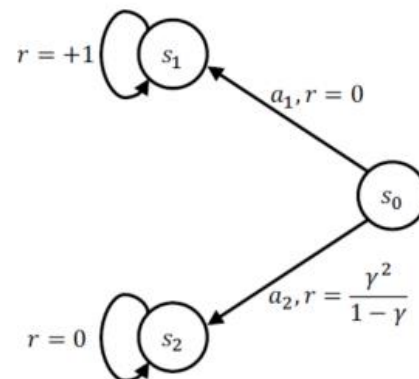
امیر حسین محمدی

۹۹۲۰۱۰۸۱



مسئله ۱. (۱۰ نمره) حد همگرایی در Value Iteration

زنجیره مارکوف زیر را در نظر بگیرید. ارزش اولیه تمام حالت‌ها را صفر فرض کنید. برای $0 < \gamma < 1$ به سوالات زیر پاسخ دهید.



(آ) (۱ نمره) عمل بهینه در زمان $t = 0$ در s کدام است؟ توضیح دهید.

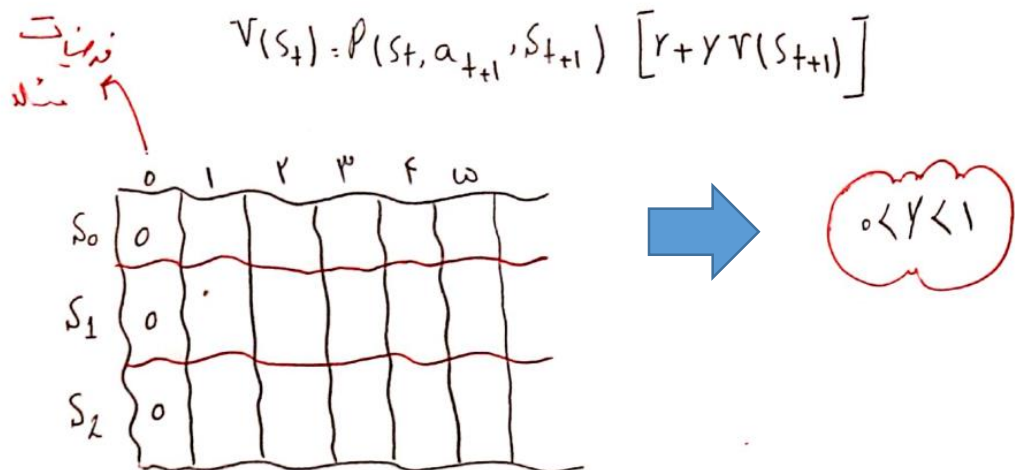
(ب) (۶ نمره) نشان دهید که الگوریتم value iteration پس از مرحله n^* برای ارزش s همگرا می‌شود؛ به طوری که n^* در رابطه زیر صدق می‌کند:

$$n^* \geq \frac{\log(1 - \gamma)}{\log \gamma}$$

(پ) (۳ نمره) با این فرض که اگر تغییرات در ارزش‌ها کمتر از θ باشد الگوریتم همگرا می‌شود، حد بالایی برای n^* برحسب θ پیدا کنید. با این حساب، برای یک γ خاص، کمترین مقدار θ چقدر باشد تا در سریعترین زمان ممکن همگرایی صورت بگیرد؟



در زمان $t=0$ اگر به صورت حریصانه بخواهیم تصمیم بگیریم به دلیل اینکه پاداشی که به ازای رفتن به حالت ۲ بدست می آید بیشتر از حالت ۱ است بنابراین عمل a_2 را در نظر می گیریم و به سمت حالت s_2 می رویم. اما اگر به صورت کلی نگاه کنیم و پاداش کلی را در نظر بگیریم در این صورت انتخاب عمل a_1 و رفتن به حالت s_1 بهینه ترین حالت است.



$t=1$

$$\begin{aligned} \Rightarrow V(s_0) &= P(s_0, a_1, s_1) [0 + \gamma V(s_1)] + P(s_0, a_2, s_2) \left[\frac{\gamma^2}{1-\gamma} + \gamma V(s_2) \right] = \\ & \quad \frac{1}{\gamma} \left[0 + \frac{\gamma^2}{1-\gamma} \right] = \\ & \quad \frac{1}{\gamma} \left[\frac{\gamma^2}{1-\gamma} \right] \\ \Rightarrow V(s_1) &= P(s_1, a_1, s_1) [1 + \gamma \overset{t=0}{V(s_1)}] = 1 \\ \Rightarrow V(s_2) &= P(s_2, a_1, s_2) [0 + \gamma \overset{t=0}{V(s_2)}] = 0 \end{aligned}$$



→ $t=1$

$$\rightarrow V(s_0) = \frac{1}{r} [0 + y x_1] + \frac{1}{r} \left[\frac{y^r}{1-y} + y x_0 \right] = \frac{1}{r} x y + \frac{1}{r} r \left[\frac{y^r}{1-y} \right] = \frac{1}{r} \left[y + \frac{y^r}{1-y} \right]$$

$$\rightarrow V(s_1) = 1 x [1 + y x_1] = 1 + y$$

$$\rightarrow V(s_r) = 1 x [0 + y x_0] = 0$$

→ $t=2$

$$\rightarrow V(s_0) = \frac{1}{r} [0 + y [1 + y]] + \frac{1}{r} \left[\frac{y^r}{1-y} + y x_0 \right] = \frac{1}{r} \left[y + y^r + \frac{y^r}{1-y} \right]$$

$$\rightarrow V(s_1) = 1 [1 + y [1 + y]] = 1 + y + y^2$$

$$\rightarrow V(s_r) = 1 [0 + y x_0] = 0$$



→ $t = n$

$$* V_1^n = y^1 + y^2 + \dots + y^{n-1}$$

$$\textcircled{\text{I}} * V_1^n = \frac{y(1-y^n)}{1-y}$$

$$\textcircled{\text{II}} * V_1^n = \frac{y}{1-y}$$

لگاریتم گیری از
طرفین

$$\rightarrow \frac{y[1-y^n]}{1-y} \geq \frac{y^r}{1-y} \rightarrow 1-y^n \geq y \rightarrow 1-y \geq y^n$$

$$\rightarrow \log(1-y) \leq \log y^n \rightarrow \log(1-y) \leq n \log y \rightarrow n^* \geq \frac{\log(1-y)}{\log y}$$



$$\left\{ \begin{array}{l} V_1^n = y + \dots + y^{n-1} \\ V_r^n = y + \dots + y^{n-r} \end{array} \right. \Rightarrow V_1^n - V_r^n \leq \theta \Rightarrow y^{n-1} \leq \theta$$

$$\Rightarrow \log y^{(n-1)} \leq \log \theta \Rightarrow (n-1) \log y \leq \log \theta \Rightarrow n-1 \leq \frac{\log \theta}{\log y}$$

$$\Rightarrow n^* \leq \frac{\log \theta}{\log y} + 1 \Rightarrow \frac{\log 1-y}{\log y} \leq n^* \leq \frac{\log \theta}{\log y} + 1$$

$$\Rightarrow \log 1-y \leq \log \theta + \log y \Rightarrow \log 1-y - \log y \leq \log \theta$$

$$\Rightarrow \log \frac{1-y}{y} \leq \log \theta \Rightarrow \theta \geq \frac{1-y}{y}$$



مسئله‌ی ۲. (۱۰ نمره امتیازی) گرادیان در Multi-armed Bandit

برای حل مسئله Bandit با k بازو می‌توان به طور مستقیم و بدون واسطه‌گری تابع ارزش نیز احتمال انتخاب کنش‌ها را مدل‌سازی کرد. اگر $H_t(a)$ میزان تمایل به انتخاب کنش a در زمان t را نشان دهد، می‌توان سیاست را به صورت زیر محاسبه کرد:



$$\pi_t(a) := \frac{e^{H_t(a)}}{\sum_{b=1}^k e^{H_t(b)}}$$

می‌توان توزیع مذکور را با بیشینه‌سازی $\sum_x \pi_t(x) q^*(x)$ $\mathbb{E}[R_t]$ آموزش داد. R_t پاداش لحظه‌ای حاصل از انجام A_t است و داریم: $q^*(a) := \mathbb{E}[R_t | A_t = a]$.

آ (۳ نمره) نشان دهید که

$$\frac{\partial \pi_t(x)}{\partial H_t(a)} = \pi_t(x) (\mathbb{I}[a = x] - \pi_t(a))$$



هدف

$$\rightarrow \frac{\partial \pi_t(b)}{\partial H_t(a)} = \pi_t(b) (\mathbb{I}_{a=b} - \pi_t(a))$$

طبق فرضیات مسئله داریم

$$\rightarrow \frac{\partial \pi_t(b)}{\partial H_t(a)} = \frac{\partial}{\partial H_t(a)} \pi_t(b) \rightarrow \pi_t(a) := \frac{e^{H_t(a)}}{\sum_{b=1}^k e^{H_t(b)}} \rightarrow = \frac{\partial}{\partial H_t(a)} \left[\frac{e^{H_t(b)}}{\sum_{c=1}^n e^{H_t(c)}} \right]$$

داریم:

$$\rightarrow \frac{\partial}{\partial x} \left[\frac{f(x)}{g(x)} \right] = \frac{\frac{\partial f(x)}{\partial x} g(x) - f(x) \frac{\partial g(x)}{\partial x}}{g(x)^2} \rightarrow = \frac{\frac{\partial e^{H_t(b)}}{\partial H_t(a)} \sum_{c=1}^n e^{H_t(c)} - e^{H_t(b)} \frac{\partial \sum_{c=1}^n e^{H_t(c)}}{\partial H_t(a)}}{(\sum_{c=1}^n e^{H_t(c)})^2}$$

$$\rightarrow = \frac{\mathbb{I}_{a=b} e^{H_t(a)} \sum_{c=1}^n e^{H_t(c)} - e^{H_t(b)} e^{H_t(a)}}{(\sum_{c=1}^n e^{H_t(c)})^2} \xrightarrow{\frac{\partial e^x}{\partial x} = e^x} = \frac{\mathbb{I}_{a=b} e^{H_t(b)}}{\sum_{c=1}^n e^{H_t(c)}} - \frac{e^{H_t(b)} e^{H_t(a)}}{(\sum_{c=1}^n e^{H_t(c)})^2} \rightarrow = \mathbb{I}_{a=b} \pi_t(b) - \pi_t(b) \pi_t(a)$$

$$\rightarrow = \mathbb{I}_{a=b} \pi_t(b) - \pi_t(b) \pi_t(a) \rightarrow \frac{\partial \pi_t(b)}{\partial H_t(a)} = \pi_t(b) (\mathbb{I}_{a=b} - \pi_t(a))$$



ب) (۲ نمره) $\frac{\partial E[R_t]}{\partial H_t(a)}$ را بر حسب $\frac{\partial \pi_t(x)}{\partial H_t(a)}$ بنویسید.



$$\Rightarrow E[R_t] = \sum_x \pi(x) q_*(x)$$

$$\Rightarrow \frac{\partial E[R_t]}{\partial H_t(a)} = \frac{\partial}{\partial H_t(a)} \left[\sum_x \pi(x) q_*(x) \right] \Rightarrow \sum_x q_*(x) \frac{\partial \pi_t(x)}{\partial H_t(a)} = \sum_x (q_*(x) - B_t) \frac{\partial \pi_t(x)}{\partial H_t(a)}$$

$$\Rightarrow \frac{\partial E[R_t]}{\partial H_t(a)} = \sum_x (q_*(x) - B_t) \frac{\partial \pi_t(x)}{\partial H_t(a)}$$



پ (۵ نمره) نشان دهید که $H_t(a)$ با رابطه‌ی زیر بروزرسانی می‌شود (منظور از \bar{R}_t میانگین پاداش‌ها از لحظه‌ی اول تا t و α نرخ یادگیری صعود در امتداد گرادیان است).

$$H_{t+1}(a) \leftarrow H_t(a) + \alpha(R_t - \bar{R}_t)(\mathbb{I}[a = A_t] - \pi_t(a))$$



$$\Rightarrow \frac{\partial E[R_t]}{\partial H_t(a)} = \sum_x (q_*(x) - B_t) \frac{\partial \pi_t(x)}{\partial H_t(a)} \Rightarrow = \sum_b \pi_t(b) (q(b) - X_t) \frac{\partial \pi_t(b)}{\partial H_t(a)} / \pi_t(b)$$

Baseline

$$\Rightarrow = \mathbb{E} \left[(q(A_t) - X_t) \frac{\partial \pi_t(A_t)}{\partial H_t(a)} / \pi_t(A_t) \right] \Rightarrow = \mathbb{E} \left[(R_t - \bar{R}_t) \frac{\partial \pi_t(A_t)}{\partial H_t(a)} / \pi_t(A_t) \right]$$

$$\Rightarrow = \mathbb{E} \left[(R_t - \bar{R}_t) \pi_t(A_t) (\mathbb{I}_{a=A_t} - \pi_t(a)) / \pi_t(A_t) \right] \Rightarrow = \mathbb{E} \left[(R_t - \bar{R}_t) (\mathbb{I}_{a=A_t} - \pi_t(a)) \right]$$

$$\Rightarrow H_{t+1}(a) = H_t(a) + \alpha (R_t - \bar{R}_t) (\mathbb{I}_{a=A_t} - \pi_t(a))$$



مسئله‌ی ۳. (۱۲ نمره) الگوریتم‌های یادگیری ارزش حالات

در این سوال به دنبال بررسی عملکرد روش‌های تخمین ارزش حالات با دو الگوریتم temporal difference و monte carlo هستیم. همچنین در نهایت همگرایی دو الگوریتم Q-learning و temporal difference را بررسی می‌کنیم.

آ (۲ نمره) روش MC برای تخمین ارزش حالات را به صورت مختصر توضیح دهید و نشان دهید تخمین MC از ارزش حالات تخمینی unbiased است.



می دانیم که هر مسئله را می توان به یک فرآیند تصمیم گیری مارکوف یا MDP تبدیل کرد، که با پنج با مولفه s, P, a, R و γ عملیات تصمیم گیری انجام می شود. وقتی هر پنج مورد را بدانیم، محاسبه یک استراتژی بهینه برای دریافت حداکثر پاداش آسان است. با این حال، در دنیای واقعی، ما تقریباً هرگز همه این اطلاعات را همزمان در اختیار نداریم. برای مثال، تشخیص احتمال transition حالت P دشوار است و بدون آن، نمی توانیم از معادله بلمن زیر برای حل مقادیر V و Q استفاده کنیم.



$$V^*(s) = \max_a \sum_{s'} P(s, a, s') [R(s, a, s') + \gamma V(s')]$$

$$Q_{k+1}(s, a) \leftarrow \sum P(s, a, s') [R(s, a, s') + \gamma \max_{a'} Q_k(s', a')] \text{ for all } (s, a)$$

❖ اما اگر بخواهیم بدون داشتن دانش قبلی از P بخواهیم مسئله ی مورد نظر را با یک فرآیند مارکوف حل کنیم داریم:
اگرچه ما نمی دانیم احتمال انتقال حالت P چقدر است، اما به طور عینی می دانیم که وجود دارد. بنابراین، ما به سادگی باید آن را پیدا کنیم. برای انجام این کار، می توانیم برای عامل خود آزمایش هایی انجام دهیم، دائماً نمونه ها را جمع آوری کنیم، پاداش دریافت کنیم، و در نتیجه عملکرد ارزش را ارزیابی کنیم. روش مونت کارلو دقیقاً به این صورت است: چندین بار امتحان می کند و مقدار V تخمینی نهایی بسیار نزدیک به مقدار V واقعی خواهد بود.



همانطور که گفته شد، روش مونت کارلو شامل اجازه دادن به یک عامل از طریق تعامل با محیط و جمع‌آوری نمونه است. این معادل نمونه برداری از توزیع احتمال $P(s, a, s')$ و $R(s, a)$ است. با این حال، تخمین مونت کارلو یا MC فقط برای یادگیری مبتنی بر آزمایش است. به عبارت دیگر، یک MDP بدون حتی تاپل P می‌تواند با آزمون و خطا، از طریق تکرارهای زیاد، یاد بگیرد.

در این فرآیند یادگیری، هر تلاش (try) یک episode نامیده می‌شود و تمام قسمت‌ها باید پایان یابد. یعنی باید به وضعیت نهایی MDP رسید. مقادیر برای هر state فقط بر اساس پاداش نهایی G_t به‌روزرسانی می‌شوند، نه بر اساس برآورد state های همسایه، همانطور که در معادله بهینه‌سازی بلمن رخ می‌دهد.

MC از complete episode ها یاد می‌گیرد و بنابراین فقط برای ما چیزی که MDP اپیزودیک می‌نامیم مناسب است. در اینجا فرمول مقدار وضعیت به روز شده به شکل زیر است:

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

که در آن $V(S_t)$: مقدار حالتی است که می‌خواهیم تخمین بزنیم که می‌تواند به صورت تصادفی یا با یک استراتژی خاص مقداردهی اولیه شود.

$$G_t = R_{t+1} + \gamma R_{t+2} + \dots + \gamma^{T-1} R_T$$

G_t در بالا محاسبه می‌شود، T زمان پایان است. گاما پارامتری مانند نرخ یادگیری است که می‌تواند بر همگرایی تأثیر بگذارد.



روش های مختلفی برای محاسبه $V(S_t)$ وجود دارد:

• First-Visit Monte-Carlo Policy Evaluation

برای هر episode، تنها اولین باری که agent به S می رسد، حساب می شود:

The first time state s appears: $N(s) \leftarrow N(s) + 1$

Total rewards update: $S(s) \leftarrow S(s) + G_t$

State s value: $V(s) = S(s) / N(s)$

When $N(s) \rightarrow \infty$, $V(s) \rightarrow v_{\pi}(s)$ so try as many episodes as you can to get as close as possible to the real state s value.

• EverEvery-Visit Monte-Carlo Policy Evaluation

برای هر قسمت، هر بار که agent به S می رسد، حساب می شود:

The first time state s appears: $N(s) \leftarrow N(s) + 1$

Total rewards update: $S(s) \leftarrow S(s) + G_t$

State s value: $V(s) = S(s) / N(s)$

When $N(s) \rightarrow \infty$, $V(s) \rightarrow v_{\pi}(s)$ so try as many episodes as you can to get as close as possible to the real state s value.



• Incremental Monte-Carlo Policy Evaluation

برای هر حالت S_t در قسمت، یک جایزه G_t وجود دارد، و برای هر بار که S_t ظاهر می شود، مقدار متوسط حالت $V(S_t)$ با فرمول زیر محاسبه می شود:

$$N(s_t) \leftarrow N(s_t) + 1$$

$$V(s_t) \leftarrow V(s_t) + \frac{1}{N(s_t)} [G_t - V(s_t)]$$



هر دو MC، first-visit و every-visit به v_π ها همگرا می‌شوند زیرا تعداد بازدیدها یا اولین بازدیدها به S ، به بی نهایت می‌رسد. مشاهده این مسئله برای first-visit MC آسان است. در این مورد، هر بازده یک تخمین مستقل و یکسان توزیع از v_π ها با واریانس محدود است. طبق قانون اعداد بزرگ، توالی میانگین‌های این تخمین‌ها به مقدار مورد انتظارشان همگرا می‌شود.

بنابراین هر میانگین به خودی خود یک تخمین unbiased است و انحراف استاندارد خطای آن به صورت 1 تقسیم بر رادیکال n کاهش می‌یابد، که در آن n تعداد میانگین‌های بازگشتی است.

به عبارت دیگر MC فقط از مشاهدات واقعی مسیر برای به روز رسانی جدول جستجو استفاده می‌کند. مزیت کلیدی این است که روش را unbiased می‌کند. به عنوان مثال تصور کنید که مشاهدات به صورت تصادفی از توزیع احتمال مربوطه نمونه برداری می‌شوند. به خصوص زمانی که مقادیر Q با توابع تقریبی نمایش داده می‌شوند که اغلب در صورت افزایش اندازه مشکلات اجتناب ناپذیر است. Bias می‌تواند به طور جدی عملکرد را مختل کند. با این حال، مسیرهای پاداش MC واریانس بسیار بیشتری را نشان می‌دهند و بنابراین معمولاً بسیار کندتر یاد می‌گیرند. همچنین در MC به روز رسانی‌های ارزش‌ها (values) بدون تاثیر از تخمین‌های اشتباه پیشین تابع ارزش انجام می‌شود.



ب) (۲ نمره) یکی از مشکلات روش MC الزام به پایان رساندن هر episode برای بهروزرسانی ارزش حالات است. موضوعی که به خصوص در مسائل long horizon چالش برانگیز است. روش TD چگونه این مشکل را برطرف می‌کند؟ روابط بهروزرسانی ارزش حالات در روش TD را ذکر کنید.



الگوریتم یادگیری تقویتی مونت کارلو بر دشواری تخمین استراتژی ناشی از یک مدل ناشناخته غلبه می کند. با این حال، یک نقطه ضعف این است که استراتژی فقط پس از کل episode به روز می شود. به عبارت دیگر، روش مونت کارلو به طور کامل از ساختار وظیفه یادگیری MDP استفاده نمی کند. بنابراین روش temporal difference از ساختار MDP استفاده کامل می کند.

همانطور که می دانیم، روش مونت کارلو مستلزم انتظار تا پایان قسمت برای تعیین $V(S_t)$ است. از طرف دیگر، برای روش تفاوت زمانی یا TD، فقط باید تا مرحله بعدی صبر کرد. یعنی در زمان $t + 1$ ، روش TD از پاداش مشاهده شده R_{t+1} استفاده می کند و بلافاصله یک هدف TD یا $R(t+1)+V(S_{t+1})$ را تشکیل می دهد و $V(S_t)$ را با خطای TD به روزرسانی می کند.

❖ در مونت کارلو، G_t خروجی واقعی از complete episode است. حال، اگر G_t را با خروجی تخمینی $R(t+1)+V(S_{t+1})$ جایگزین کنیم، TD به این شکل خواهد بود:



$$V(S_t) \leftarrow V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

❖ به طوریکه:

$R(t+1)+V(S_{t+1}) - V(S_t)$ ➡ error

$R(t+1)+V(S_{t+1})$ ➡ Target value

بنابراین مونت کارلو از خروجی دقیق G_t برای به روزرسانی مقدار استفاده می کند، در حالی که TD از معادله بهینه سازی بلمن برای تخمین ارزش استفاده می کند و سپس مقدار تخمین زده شده را با مقدار هدف به روزرسانی می کند.



بدیهی است که روش‌های TD نسبت به روش‌های DP مزیتی دارند زیرا نیازی به مدلی از محیط، پاداش و توزیع احتمال بعدی آن ندارند.

مزیت آشکار بعدی روش‌های TD نسبت به روش‌های مونت کارلو این است که روش‌های مونت کارلو باید تا پایان یک episode صبر کرد، زیرا تنها در این صورت return یا خروجی مشخص می‌شود، در حالی که با روش‌های TD فقط باید یک مرحله زمانی منتظر بمانیم. برخی از برنامه‌ها دارای episode‌های بسیار طولانی هستند، به طوری که به تاخیر انداختن همه یادگیری‌ها تا پایان یک episode بسیار فرایند را کند می‌کند. روش‌های TD بسیار کمتر در معرض این مشکلات هستند زیرا آنها از هر transition بدون توجه به اقدامات بعدی یاد می‌گیرند.



ت) (۵ نمره) روش TD برای یادگیری ارزش حالات از حدس ارزش حالات بعدی استفاده می‌کند. آیا این موضوع همگرایی این الگوریتم را با مشکل مواجه می‌کند؟ اگر جواب مثبت است تحت چه شرایطی همگرایی قابل تضمین نیست؟ توضیح دهید. در مورد الگوریتم Q-learning که در آن کنش‌ها به صورت تصادفی انتخاب می‌شوند چطور؟ آیا همگرایی برای آن الگوریتم تضمین می‌شود؟



الگوریتم‌های TD مختلفی وجود دارد، به عنوان مثال Q-learning و SARSA که خواص همگرایی آنها به طور جداگانه در بسیاری از موارد مورد مطالعه قرار گرفته است. بنابراین همگرایی TD به صورت کلی اثبات و تضمین نمی شود اما میتوان تحت شرایطی آن را اثبات کرد. برای اثبات همگرایی TD ثابت شده است که به ازای هر fixed policy مثل π ، به V^π همگرا می شود به این شرط که پارامتر ثابت اندازه گام به اندازه کافی کوچک باشد، و با احتمال ۱ اگر پارامتر اندازه گام مطابق با شرایط approximation conditions معمول کاهش یابد که به شکل زیر تعریف می شود:

$$\sum_{k=1}^{\infty} \alpha_k(a) = \infty \quad \text{and} \quad \sum_{k=1}^{\infty} \alpha_k^2(a) < \infty. \quad \Rightarrow \quad \text{رابطه ی ۱}$$

بنابراین در غیر از این حالت همگرایی آن تضمین نمی شود. اکثر اثبات‌های همگرایی فقط برای رابطه‌ی زیر تعریف می شود، اما برخی نیز در مورد linear function approximation اعمال می شوند:

$$V(s_t) \leftarrow V(s_t) + \alpha [r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]. \quad \Rightarrow \quad \text{رابطه ی ۲}$$

همچنین همگرایی برای الگوریتم Q-learning تضمین نمی شود که خود یک نوعی از الگوریتم های TD است. اما همگرایی آن تحت شرایط خاصی اثبات می شود. در برخی از اثبات‌های همگرایی، به عنوان مثال. در مقاله Convergence of Q-learning: A Simple Proof از رابطه ی ۲ به عنوان شرط همگرایی یاد می شود که به رابطه‌ی رابینز-مونرو معروف است. که مجدداً به صورت زیر بازنویسی می کنیم:

$$\begin{aligned} 1. \sum_t \alpha_t(s, a) &= \infty \\ 2. \sum_t \alpha_t^2(s, a) &< \infty \end{aligned} \quad \Rightarrow \quad \text{Robbins-Monro}$$

که در آن $\alpha_t(s, a)$ نرخ یادگیری در مرحله زمانی t است (که می تواند به حالت s و عمل a بستگی داشته باشد. البته همانطور که گفته شد شرایط خاص مورد نیاز برای همگرایی روش‌های TD ممکن است بسته به اثبات و الگوریتم TD خاص متفاوت باشد.



ت (۳ نمره) یکی از دسته روش‌های بهینه‌سازی روش‌های trust region هستند که در آن‌ها همانند سایر روش‌های بهینه‌سازی تکرار شونده، در هرگام از حدس گام قبل با مکانیزمی به حدس گام بعد می‌رسیم. نکته مهم در این روش‌ها کنترل نزدیکی حدس بعد به حدس قبلی و به اصطلاح باقی ماندن در فضای اطمینان است. **TRPO** با الهام‌گیری از همین موضوع سعی در کنترل میزان تغییرات سیاست در هرگام با استفاده از تابع هزینه ذیل دارد. این کنترل میزان تغییرات و جلوگیری از ناگهانی در سیاست سبب ایجاد پایداری در یادگیری سیاست می‌گردد.

$$\begin{aligned} & \underset{\theta}{\text{maximize}} && \hat{\mathbb{E}}_t \left[\frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \hat{A}_t \right] \\ & \text{subject to} && \hat{\mathbb{E}}_t [\text{KL} [\pi_{\theta_{\text{old}}}(\cdot | s_t), \pi_{\theta}(\cdot | s_t)]] \leq \delta \end{aligned}$$

با این حال این مسئله بهینه‌سازی دارای hard constraint ای است که حل آن را چالش برانگیز می‌کند. به صورت مختصر و کلی بیان کنید روش پیشنهادی **PPO** چگونه این مشکل را حل می‌کند؟



بر اساس الگوریتم Trust region Policy optimization (TRPO) منطقه ای که تقریب های محلی تابع در آن دقیق است، یک منطقه Trust است. TRPO مطمئن می شود که policy از نقطه شروع خیلی دور نمی شود. برای اندازه گیری این تغییر در Policy، از KL-divergence استفاده می کند (در واقع KL-divergence به جای نگاه کردن به بردار پارامتر، به توزیع بردار پارامتر نگاه می کند) که به صورت زیر تعریف می شود:

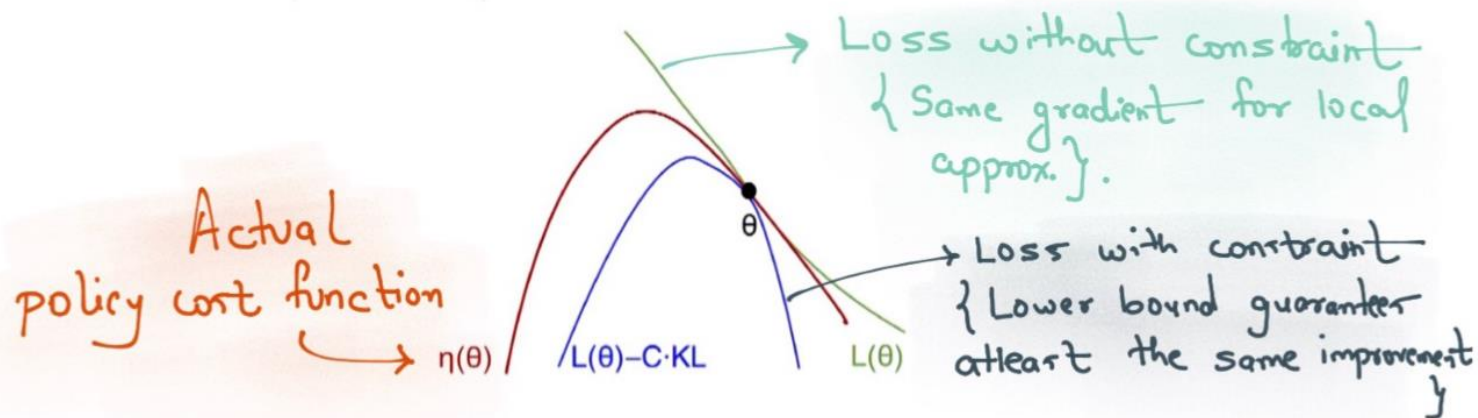


$$\begin{aligned} & \underset{\theta}{\text{maximize}} \quad \hat{\mathbb{E}}_t \left[\frac{\pi_{\theta}(a_t | s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t)} \hat{A}_t \right] \\ & \text{subject to} \quad \hat{\mathbb{E}}_t [\text{KL}[\pi_{\theta_{\text{old}}}(\cdot | s_t), \pi_{\theta}(\cdot | s_t)]] \leq \delta. \end{aligned}$$

بنابراین اساساً ما همان مسئله بهینه سازی policy gradient را داریم، فقط یک محدودیت اضافه کردیم:

$$\underset{\theta}{\text{maximize}} \quad \hat{\mathbb{E}}_t \left[\frac{\pi_{\theta}(a_t | s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t)} \hat{A}_t \right] - \beta \hat{\mathbb{E}}_t [\text{KL}[\pi_{\theta_{\text{old}}}(\cdot | s_t), \pi_{\theta}(\cdot | s_t)]]$$

□ به صورت تئوری مشخص می شود که اگر به جای گرفتن میانگین KL، از حداکثر KL استفاده کنیم، آنگاه یک کران پایین برای policy به دست می آوریم. همانطور که در شکل زیر می بینیم:



منحنی آبی منحنی محدودیت است که همان کران پایینی است. اگر حد پایین را به حداکثر برسانیم، نسبت به کران پایین بهبود بیشتری خواهیم داشت.



محدودیت های روش TRPO

- استفاده از معماری با خروجی های متعدد سخت است. به عنوان مثال تابع policy و value نیازمند وزن دهی عبارات مختلف در متریک فاصله است زیرا KL divergence به روز رسانی تابع ارزش (value) کمکی نمی کند
- به طور تجربی در وظایفی که نیاز به CNN و RNN عمیق دارند، عملکرد ضعیفی دارد.
- گرادیان های conjugate پیاده سازی را پیچیده تر و کمتر از SGD انعطاف پذیر می کند.

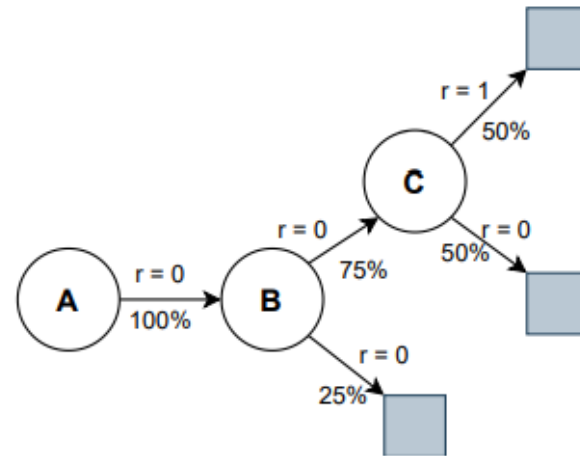
برای حل مشکلات گفته شده در روش TRPO خصوصاً conjugate gradient روش Proximal Policy Optimisation ارائه شد. PPO تعادل خوبی بین سهولت اجرا، پیچیدگی نمونه و سهولت تنظیم برقرار می کند. PPO سعی می کند در هر مرحله زمانی یک به روز رسانی را محاسبه کند که تابع هزینه را به حداقل می رساند و در عین حال اطمینان می دهد که انحراف از policy قبلی نسبتاً کم است. PPO برای پیاده سازی و تنظیم ساده تر است و الگوریتم یادگیری تقویتی پیش فرض در OpenAI است. PPO به جای constraint از penalty استفاده می کند.

$$\underset{\theta}{\text{maximize}} \sum_{n=1}^N \frac{\pi_{\theta}(a_n | s_n)}{\pi_{\theta_{\text{old}}}(a_n | s_n)} \hat{A}_n - C \cdot \overline{\text{KL}}_{\pi_{\theta_{\text{old}}}}(\pi_{\theta})$$

PPO عملکردی مشابه TRPO دارد، اما فقط بهینه سازی مرتبه اول را ارائه می دهد. از آنجایی که انتخاب β که در کل دوره بهینه سازی به خوبی کار کند بسیار مشکل است، بنابراین فقط از KL به عنوان یک ابرپارامتر برای تعیین ضرایب جریمه β استفاده می شود. بنابراین اساساً، اگر برای اندازه گام (step-size) اگر KL زیاد باشد، ضریب جریمه (penalty) نیز افزایش می یابد، یا اگر KL خیلی کم باشد ضریب جریمه کاهش می یابد.



پ (۳ نمره) برای درک بهتر تفاوت این دو روش، ارزش حالات مربوط به markov reward process زیر را با توجه به episode های بیان شده با هر دو روش محاسبه کنید. آیا تفاوتی در مقدار محاسبه شده وجود دارد؟ نتیجه را تفسیر کنید.



A 0 B 0 C 0

A 0 B 0

B 0

B 0 C 0

C 1

C 1

C 0

C 1



با توجه به روش temporal difference و episode های گفته شده در مسئله داریم:

- برای حالت C طبق فرضیات مسئله داریم:

→
A 0 B 0 C 0
B 0 C 0
C 1
C 1
C 0
C 1

- همانطور که از episode هایی که شامل C است مشخص می شود که از ۶ حالت موجود ۳ حالت با $r=1$ به پایان می رسند و ۳ حالت با صفر. بنابراین برای ارزش حال C یا $V(C)$ داریم:

→
$$V(C) = \frac{3}{6} * 0 + \frac{3}{6} * 1 = \frac{1}{2}$$

↓
A 0 B 0 C 0
A 0 B 0
B 0
B 0 C 0
C 1
C 1
C 0
C 1



با توجه به روش temporal difference و episode های گفته شده در مسئله داریم:

- برای حالت B طبق فرضیات مسئله داریم:

→
A 0 B 0 C 0
A 0 B 0
B 0
B 0 C 0

↓
A 0 B 0 C 0
A 0 B 0
B 0
B 0 C 0
C 1
C 1
C 0
C 1

- همانطور که از episode هایی که شامل B است مشخص می شود که از 4 حالت موجود ۲ حالت با $r=0$ به پایان می رسند و ۲ حالت با صفر به C می روند. بنابراین برای ارزش حالت B یا $V(B)$ داریم:

→
$$V(B) = \frac{2}{4} * 0 + \frac{2}{4} * V(C) = \frac{2}{4} * 0 + \frac{1}{2} * \frac{1}{2} = \frac{1}{4}$$

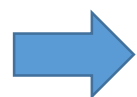


با توجه به روش temporal difference و episode های گفته شده در مسئله داریم:



A 0 B 0 C 0
A 0 B 0
B 0
B 0 C 0
C 1
C 1
C 0
C 1

- برای حالت B طبق فرضیات مسئله داریم:



A 0 B 0 C 0
A 0 B 0

- همانطور که از episode هایی که شامل A است مشخص می شود که از ۲ حالت موجود ۲ حالت با $r=0$ به حالت B می روند. بنابراین برای ارزش حالت A یا $V(A)$ داریم:



$$V(A) = \frac{2}{2} * V(B) = 1 * \frac{1}{4} = \frac{1}{4}$$



با توجه به روش Monte carlo و episode های گفته شده در مسئله داریم:

- برای حالت C طبق فرضیات مسئله داریم:

→

A	0	B	0	C	0
		B	0	C	0
				C	1
				C	1
				C	0
				C	1

↓

A	0	B	0	C	0		
		A	0	B	0		
				B	0		
				B	0	C	0
						C	1
						C	1
						C	0
						C	1

- همانطور که از episode هایی که شامل C است مشخص می شود که از ۶ حالت موجود ۳ حالت با $r=1$ به پایان می رسند و ۳ حالت با صفر. بنابراین برای ارزش حال C یا $V(C)$ داریم:

→

$$V(C) = \frac{3}{6} * 0 + \frac{3}{6} * 1 = \frac{1}{2}$$



با توجه به روش Monte Carlo و episode های گفته شده در مسئله داریم:

- برای حالت B طبق فرضیات مسئله داریم:

→

A	0	B	0	C	0
A	0	B	0		
		B	0		
B	0	C	0		

↓

A	0	B	0	C	0
A	0	B	0		
		B	0		
B	0	C	0		
				C	1
				C	1
				C	0
				C	1

- همانطور که از episode هایی که شامل B است مشخص می شود که از 4 حالت موجود ۲ حالت با $r=0$ به پایان می رسند و ۲ حالت با صفر به C می روند. بنابراین برای ارزش حالت B یا $V(B)$ داریم:

→

$$V(B) = \frac{2}{4} * 0 + \frac{2}{4} * 0 = 0$$

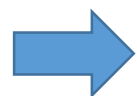


با توجه به روش temporal difference و episode های گفته شده در مسئله داریم:



A 0 B 0 C 0
A 0 B 0
B 0
B 0 C 0
C 1
C 1
C 0
C 1

- برای حالت B طبق فرضیات مسئله داریم:



A 0 B 0 C 0
A 0 B 0

- همانطور که از episode هایی که شامل A است مشخص می شود که از ۲ حالت موجود ۲ حالت با $r=0$ به حالت B می روند. بنابراین برای ارزش حالت A یا $V(A)$ داریم:



$$V(A) = \frac{2}{2} * 0 = 0$$



Monte Carlo



$$\left\{ \begin{array}{l} V(C) = \frac{1}{2} \\ V(B) = 0 \\ V(A) = 0 \end{array} \right.$$

Temporal difference



$$\left\{ \begin{array}{l} V(C) = \frac{1}{2} \\ V(B) = \frac{1}{4} \\ V(A) = \frac{1}{4} \end{array} \right.$$

همانطور که ملاحظه می شود بین روش TD و MC در محاسبه ی ارزش های A,B,C تفاوت وجود دارد. این مسئله به این علت است که الگوریتم MC در هر مرحله صرفاً همان State را در نظر می گیرد و این در حالی است که روش TD در هر مرحله علاوه بر ارزش آن مرحله ارزش مرحله ی بعدی را نیز در نظر می گیرد. بنابراین این دو روش صرفاً در حالت C با یکدیگر برابر هستند و این مسئله به این علت است حالت C حالت پایانی هست و بعد از آن حالتی وجود ندارد.



پ (۴ نمره) یکی از روش‌های موفق یادگیری تقویتی off policy الگوریتم SAC می‌باشد. در این مقاله تابع هدف یادگیری تقویتی

$$J(\pi) = \sum_{t=0}^T \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \rho_\pi} [r(\mathbf{s}_t, \mathbf{a}_t)] \quad (۳)$$

$$J(\pi) = \sum_{t=0}^T \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \rho_\pi} [r(\mathbf{s}_t, \mathbf{a}_t) + \alpha \mathcal{H}(\pi(\cdot | \mathbf{s}_t)))] \quad (۴)$$

تغییر پیدا کرده است.

اولاً ترم $\mathcal{H}(\pi(\cdot | \mathbf{s}_t))$ چه تاثیری دارد؟

دوماً طبق مقاله بیان کنید که سیاست جدید به دست آمده در گام policy improvement در الگوریتم policy iteration برای این تابع هدف جدید به چه شکل خواهد بود؟



در یادگیری تقویتی، اکتشاف (exploration) در مقابل بهره برداری exploitation بخش مهمی از مفهوم است. تصمیم گیری خیلی سریع بدون کاوش کافی می تواند یک شکست بزرگ باشد. کاوش جزء اصلی یادگیری است. همانطور که مشخص است، افزودن نویز به عمل یکی از موارد معقول در کاوش است. آنتروپی یکی دیگر از ابزارهای کاوش قدرتمند است. آنتروپی بالا باید به ما اطمینان دهد که از اقدامات مکرر جلوگیری کنیم.

الگوریتم SAC بر اساس چارچوب maximum entropy در یادگیری تقویتی است. مکانیسم actor learning، policy ها را بهینه می کند تا هم بازده مورد انتظار (expected return) و هم آنتروپی مورد انتظار policy را به حداکثر برساند که به صورت زیر تعریف می شود.

$$J(\pi) = \sum_{t=0}^T \mathbb{E}_{(s_t, a_t) \sim \rho_{\pi}} [r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot | s_t))]$$

فرمول maximum entropy بهبود قابل توجهی در exploration و robustness ایجاد می کند. policy های maximum entropy در مواجهه با خطاهای مدل و تخمین قوی هستند. آنها کاوش (exploration) را با به دست آوردن رفتارهای متنوع بهبود می بخشند.



در Soft Actor-Critic، policy با فرمول زیر به روز می شود:

$$\pi_{\text{new}} = \arg \min_{\pi' \in \Pi} D_{\text{KL}} \left(\pi'(\cdot | s_t) \parallel \frac{\exp \left(\frac{1}{\alpha} Q^{\pi_{\text{old}}}(s_t, \cdot) \right)}{Z^{\pi_{\text{old}}}(s_t)} \right) \quad \text{رابطه ی یک} \rightarrow$$

اولین توزیع در رابطه بالا توزیع $\pi'(\cdot | s_t)$ است. این توزیعی از مجموعه توزیع های گاوسی Π است. اساساً، این توزیع حداقل cross entropy را فراهم می کند. این همان حداقل واگرایی KL است که در رابطه ی زیر وجود دارد:

$$H(p, q) = H(p) + D_{\text{KL}}(p \parallel q)$$

توزیع دوم در رابطه ی یک با استفاده از تابع softmax ساخته شده است و مخرج آن تابع پارتیشن است که به صورت زیر تعریف می شود. مقادیر تابع حالت-مقدار (state-value) به دست آمده برای تابع policy قبلی π_{old} ، مجموعه ای از مقادیر x_i را برای تابع softmax در معادله تشکیل می دهد.

$$p_i = \frac{\exp(x_i)}{\sum_{j=1}^n \exp(x_j)}, \quad Z = \sum_{j=1}^n \exp(x_j)$$

بر اساس مقاله ی SAC از آنجایی که در عمل سیاست‌هایی را ترجیح داده می شوند که قابل اجرا باشند، بنابراین policy به مجموعه‌ای از policy های Π محدود می شود که می‌تواند برای مثال با یک خانواده پارامتری از توزیع‌ها مانند Gaussians مطابقت داشته باشد.



مسئله ۴. (۱۳ نمره) معماری Actor Critic و روش‌های Policy Gradient

گرادیان تابع هدف ساده شده روش‌های policy based در ۱ نشان داده شده است. یکی از ویژگی‌های گرادیان این تابع هدف واریانس بالای آن به دلیل ذات تصادفی تولید یک trajectory و دریافت پاداش است، موضوعی که فرآیند آموزش شبکه عصبی را با چالش همراه می‌کند. یکی از رویکردها برای کاهش واریانس این گرادیان استفاده از مقداری تحت عنوان baseline است. در این سوال ابتدا به بررسی تاثیر یک مقدار ثابت به عنوان baseline پرداخته و سپس با مطالعه دسته مهمی از معماری‌های شبکه‌های یادگیری تقویتی با نام actor critic که به دنبال یادگیری این baseline هستند، با دو روش ارائه شده در این شاخه آشنا می‌شویم.

$$\mathbb{E}_{\tau \sim p(\tau; \theta)} [\nabla_{\theta} \log p(\tau; \theta) r(\tau)] \quad (۱)$$

این گرادیان با استفاده از یک مقدار ثابت c تحت عنوان baseline به شکل زیر تغییر می‌کند:

$$\mathbb{E}_{\tau \sim p(\tau; \theta)} [\nabla_{\theta} \log p(\tau; \theta) (r(\tau) - c)] \quad (۲)$$


آ (۲ نمره) گرادیان تابع هدف ۲ را به شکل $\mathbb{E}[f(x) - \phi(x)] + \mathbb{E}[\phi(x)]$ باز نویسی کنید که در آن $\mathbb{E}[f(x)]$ همان عبارت ۱ است. مقدار $\mathbb{E}[\phi(\tau)]$ را نیز محاسبه کنید.

ب (۳ نمره) ثابت کنید c بهینه که سبب کمینه شدن $\text{Var}[f(\tau) - \phi(\tau)]$ می‌گردد برابر است با:

$$c = \frac{\mathbb{E} [(\nabla_{\theta} \log p(\tau; \theta))^{\top} r(\tau)]}{\mathbb{E} [(\nabla_{\theta} \log p(\tau; \theta))^{\top}]}$$



• طبق فرضیات مسئله داریم:


$$\mathbb{E}_{\tau \sim p(\tau; \theta)} [\nabla_{\theta} \log p(\tau; \theta) r(\tau)]$$


$$\mathbb{E}_{\tau \sim p(\tau; \theta)} [\nabla_{\theta} \log p(\tau; \theta) (r(\tau) - c)]$$

$$* \mathbb{E}_{\tau \sim p(\tau; \theta)} [\nabla_{\theta} \log p(\tau; \theta) (r(\tau) - c)] =$$

$$* \mathbb{E}_{\tau \sim p(\tau; \theta)} [\nabla_{\theta} \log p(\tau; \theta) (r(\tau)) - \nabla_{\theta} \log p(\tau; \theta) c]$$

$$* \mathbb{E}[f(\tau)] = \mathbb{E}_{\tau \sim p(\tau; \theta)} [\nabla_{\theta} \log p(\tau; \theta) (r(\tau))]$$

$$* \mathbb{E}[\phi(\tau)] = \mathbb{E}_{\tau \sim p(\tau; \theta)} [\nabla_{\theta} \log p(\tau; \theta) c]$$


$$\mathbb{E}_{\tau \sim p(\tau; \theta)} [\nabla_{\theta} \log p(\tau; \theta) c] = \int p(\tau; \theta) \nabla_{\theta} \log p(\tau; \theta) c \, d\tau$$

$$= \int \nabla_{\theta} p(\tau; \theta) c \, d\tau$$

$$= c \nabla_{\theta} \int p(\tau; \theta) \, d\tau$$

$$= c \nabla_{\theta} \int 1 \, d\tau = 0$$



➡
$$\text{Var} = \bar{E}_{\tau \sim p(\tau; A)} [\nabla_{\theta} \log p(\tau; A) (r(\tau) - c)]^T -$$
$$\bar{E}_{\tau \sim p(\tau; A)} [\nabla_{\theta} \log p(\tau; A) (r(\tau) - c)]^T$$

{
$$\frac{d \text{Var}}{d c} = \frac{d}{d c} \bar{E}_{\tau \sim p(\tau; A)} [\nabla_{\theta} \log p(\tau; A) (r(\tau) - c)]^T$$
$$= \frac{d}{d c} \bar{E} [\nabla_{\theta} \log p(\tau; A)]^T (r(\tau) - c)^T \quad \mathcal{Q}(\tau) = \nabla_{\theta} \log p(\tau; A)$$
$$= \frac{d}{d c} \bar{E} [\mathcal{Q}(\tau)^T (r(\tau) - c)^T] = \frac{d}{d c} (\bar{E} [\mathcal{Q}(\tau)^T r(\tau)^T] - \gamma c^T \bar{E} [\mathcal{Q}(\tau)^T r(\tau)] +$$
$$c^T \bar{E} [\mathcal{Q}(\tau)^T]) = \frac{d}{d c} (-\gamma c^T \bar{E} [\mathcal{Q}(\tau)^T r(\tau)] + c^T \bar{E} [\mathcal{Q}(\tau)^T])$$
$$= -\gamma \bar{E} [\mathcal{Q}(\tau)^T r(\tau)] + \gamma c^T \bar{E} [\mathcal{Q}(\tau)^T]$$



$$\Rightarrow \frac{dvar}{dc} = 0 \Rightarrow -\gamma E[\mathcal{Q}(\tau)^T r(\tau)] + \gamma c E[\mathcal{Q}(\tau)^T] = 0$$

$$\Rightarrow c = \frac{E[\mathcal{Q}(\tau)^T r(\tau)]}{E[\mathcal{Q}(\tau)^T]} \Rightarrow c = \frac{E[(\nabla_{\theta} \log p(\tau; \theta))^T r(\tau)]}{E[(\nabla_{\theta} \log p(\tau; \theta))^T]}$$