به نام خدا

یادگیری تقویتی

نیم‌سال بهار ۴۰۲-۱۴۰۱

اساتید: دکتر رهبان، آقای حسنی

دانشکده‌ی مهندسی کامپیوتر

تمرین سری اول — MDP, Tabular Methods, Value Approximation — زمان تحویل: ۱۹ اسفند

لطفا نکات زیر را رعایت کنید:

- سوالات خود را از طریق پست مربوط به تمرین در Quera مطرح کنید.
- پاسخ ارسالی واضح و خوانا باشد.
- در هر کدام از سوالات، اگر از منابع خاصی استفاده کرده‌اید، آن را ذکر کنید.
- اگر با افرادی همفکری کرده‌اید، نام ایشان را ذکر کنید.
- پاسخ ارسالی باید توسط خود شما نوشته شده باشد. به اسکرین‌شات از منابع یا پاسخ افراد دیگر نمره‌ای تعلق نمی‌گیرد.
- تمام پاسخ‌های خود را در یک فایل با فرمت RL_HW#_[SID]_[Fullname].zip روی کوئرا قرار دهید.
- برای ارسال هر تمرین تا ساعت ۲۳:۵۹ روز ددلاین فرصت دارید. علاوه بر آن، در هر تمرین می توانید تا سقف ۵ روز از تأخیر مجاز باقیمانده‌ی خود استفاده کنید.

سوال ۱: پاشنه‌ی ابیل (۲۵ نمره)

می‌خواهیم اثباتی را که برای همگرایی روش Value Iteration در کلاس مطرح شد دقیقتر بررسی کنیم و آن را اندکی گسترش دهیم. همانطور که می‌دانید V_k^* حداکثر مقدار مجموع پاداشی است که در k مرحله می‌توانیم به دست آوریم و در رابطه‌ی بلمن صدق می‌کند.

(آ) ابتدا مقدار پاداش‌ها را نامنفی در نظر بگیرید. یک کران بالا برای V_k^* بیابید.

پاسخ: فرض کنید که R یک کلان بالا برای پاداش‌ها باشد و ضریب کاهش γ باشد. (فرض کنید $\gamma < 1$)

$$V_k^* \leq \sum_{i=1}^{k} \gamma^{i-1} R = \frac{1-\gamma^k}{1-\gamma} R \leq \frac{R}{1-\gamma}$$

(ب) می‌خواهیم ثابت کنیم که V_k^* نسبت به k صعودی است. با در نظر گرفتن یک policy خاص تا مرحله‌ی k ام یک V_{k+1}^{π} پیدا کنید که

$$V_{k+1}^{\pi} \geq V_k^*$$

سپس به کمک تعریف V_{k+1}^* صعودی بودن تابع V^* را ثابت کنید و به کمک قسمت قبل همگرایی الگوریتم Value iteration را نتیجه‌گیری کنید.

پاسخ: V_k^* تابع بهینه‌ی ارزش در افق k گامه است. π را policy در نظر بگیرید که متناظر با این تابع ارزش است، برای مرحله‌ی $k+1$ فرض کنید π یک اقدام دلخواه را انتخاب می‌کند. از آن‌جایی که پاداش‌ها نامنفی هستند، داریم:

$$V_k^* = V_k^{\pi}$$

$$V_{k+1}^{\pi} = V_k^{\pi} + \gamma^{k-1} r_{k+1} = V_k^* + \gamma^{k-1} r_{k+1} \geq V_k^*$$

حال میتوانیم ببینیم که V_k^* بر حسب k صعودی است.

$$V_{k+1}^* \geq V_{k+1}^{\pi} \geq V_k^*$$

و چون که کران بالا دارد (قسمت قبلی سوال) پس همگرا می‌شود.

(ج) با میل دادن دو طرف معادله‌ی بلمن و به دست آوردن V^* ثابت کنید که جواب به دست آمده بهینه است.

پاسخ: تابع V_k^* به صورت بازگشتی از طریق رابطه‌ی زیر به دست می‌آید.

$$V_{k+1}^*(s) = \max_a \sum_{s'} P(s'|a,s)[R(s',a,s) + \gamma V_k^*(s')]$$

در قسمت قبل ثابت کردیم V_k^* همگرا می‌شود. مقدار حد آن را V_∞^* بنامید. با میل دادن دوطرف معادله‌ی بالا به بینهایت با توجه به پیوستگی عملگر بلمن، مقدار V_k^* و V_{k+1}^* را با V_∞^* جایگزین می‌کنیم.

$$\lim_{k\to\infty} V_{k+1}^*(s) = \lim_{k\to\infty} \max_a \sum_{s'} P(s'|a,s)[R(s',a,s) + \gamma V_k^*(s')]$$

که باتوجه به پیوسته بودن عملگر max حد به داخل عبارت می‌رود:

$$\lim_{k\to\infty} V_{k+1}^*(s) = \max_a \sum_{s'} \lim_{k\to\infty} P(s'|a,s)[R(s',a,s) + \gamma V_k^*(s')]$$

$$\lim_{k\to\infty} V_{k+1}^*(s) = \max_a \sum_{s'} P(s'|a,s)[R(s',a,s) + \gamma \lim_{k\to\infty} V_k^*(s')]$$

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

چون که V_∞^* در رابطه‌ی بلمن صدق می‌کند پس بهینه است.

(د) حال می‌خواهیم شرط نامنفی بودن پاداش را برداریم. فرض کنید که terminating state نداریم. یک MDP جدید که از اضافه شدن مقدار پاداش r_0 به تمامی پاداش‌های MDP فعلی به دست می‌آید در نظر بگیرید، با یافتن مقدار V_k^* و action بهینه بر حسب مقادیر MDP قبلی و r_0 ثابت کنید که در حالت r منفی نیز الگوریتم Value iteration به policy بهینه قبلی می‌رسد. V^* جدید را نیز محاسبه کنید.

پاسخ:
مقدار $R^{'}(s^{'},a,a) = R(s^{'},a,s) + r_0$ را مقدار پاداشی بگیرید که از اضافه کردن r_0 به تمام پاداش‌ها به دست می‌آید. به استقرا ثابت می‌کنیم که تابع ارزش جدید از رابطه‌ی زیر به دست می‌آید:

$$V_k^{'*}(s) = \frac{1-\gamma^k}{1-\gamma} r_0 + V_k^*(s), k > 0$$

اثبات:

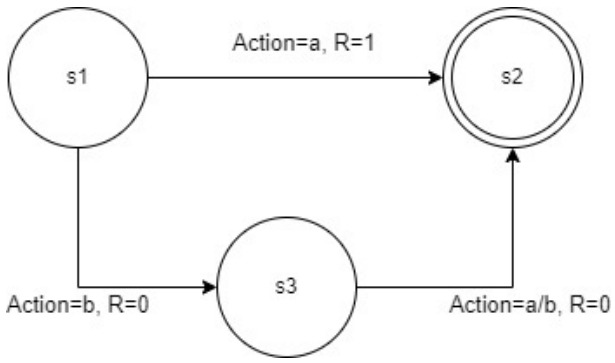
$$\begin{aligned}
V_{k+1}^{'*}(s) = & \max_a \sum_{s'} P(s'|a,s)[R^{'}(s',a,s) + \gamma V_k^{'*}(s')] \\
&= \max_a \sum_{s'} P(s'|a,s)[(R(s^{'},a,s) + r_0) + \gamma(\frac{1-\gamma^k}{1-\gamma} r_0 + V_k^*(s))] \\
&= \max_a \sum_{s'} P(s'|a,s)[R(s^{'},a,s) + \frac{1-\gamma^{k+1}}{1-\gamma} r_0 + \gamma V_k^*(s)] \\
&= \frac{1-\gamma^{k+1}}{1-\gamma} r_0 + \max_a \sum_{s'} P(s'|a,s)[R(s^{'},a,s) + \gamma V_k^*(s)] \\
&= \frac{1-\gamma^{k+1}}{1-\gamma} r_0 + V_{k+1}^*(s)
\end{aligned}$$

در خط اول مقدار پاداش جدید و ارزش جدید را به ترتیب با پاداش قدیم و رابطه‌ي به دست آمده از فرض استقرا جایگزین می‌کنیم.
در خط یکی مانده به آخر، چون که عبارت بیشینه کننده نسبت به ستاپ قبلی مسئله تغییری نکرده است action ی که برای maximize کردن انتخاب می‌شود با حالت قبل فرق ندارد پس policy فرقی نمی‌کند.
می‌دانیم که چون $V_k^{'*}$ همگرا می‌شود، تابع V_k^* نیز همگرا می‌شود. همچنین دیدیم که action بهینه با افزایش یک پاداش ثابت تغییر نمی‌کند، پس چون که میدانیم در حالت جدید نیز همگرا می‌شویم، policy بهینه نیز تغییر نمی‌کند. برای محاسبه‌ی $V^{*'}$ نیز $V^{k'}$ را حد میگیریم و خواهیم داشت:

$$V^{'*}(s) = \frac{1}{1-\gamma} r_0 + V^*(s)$$

(ه) چرا لازم است شرط نداشتن terminating state را داشته باشیم؟ سعی کنید با یک مثال نقض توضیح دهید.

پاسخ:



شکل ۱: مثال نقض برای حالت terminating state

در اینجا با در نظر گرفتن ضریب کاهش $\gamma = 0.9$ بدون تغییر پاداش‌ها در حالت s1 اقدام a انتخاب می‌شود. و مقدار ارزش برابر ۱ خواهد بود ولی اگر به پاداش‌ها مقدار ۲ را اضافه کنیم، آنگاه می‌بینیم که در حالت بهینه در s1 باید اقدام b را انتخاب کنیم و ارزش برابر ۳/۸ خواهد بود که از مقدار ۳ که از انتخاب اقدام a در این حالت به دست می‌آمد بیشتر است.

سوال ۲: Mutated Policy Iteration (۲۵ نمره)

فرض کنید که در یک MDP مقدار پاداش‌ها نامنفی باشد. همانطور که می‌دانید الگوریتم policy iteration از دو بخش بهبود پالیسی و ارزیابی

پالیسی تشکیل شده‌است. با شروع از یک پالیسی مانند π_0 ، در مرحله‌ی t ام از الگوریتم ابتدا برای پالیسی t ام مقدار Valueها برای π_t را به کمک Policy Evaluation می‌یابیم. در Policy Evaluation برای پالیسی π_t مقدار ارزش به صورت بازگشتی از رابطه‌ی زیر به دست می‌آید.

$$V_0^{\pi_t}(s) = 0$$

$$V_{k+1}^{\pi_t}(s) = \sum_{s'} P(s'|\pi_t(s), s)[R(s', \pi_t(s), s) + \gamma V_k^{\pi_t}(s')] \tag{۱}$$

در ادامه پس از همگرا شدن $V_k^{\pi_t}$ به $V_\infty^{\pi_t}$ ، به کمک Policy Improvement یک پالیسی جدید به دست می‌آوریم که بین actionها آن یکی را انتخاب کند که بیشترین ارزش را به دست آورد.

$$\pi_{t+1}(s) = \arg\max_a \sum_{s'} P(s'|\pi_t(s), s)[R(s', \pi_t(s), s) + \gamma V_\infty^{\pi_t}(s')] \tag{۲}$$

فرض کنید که می‌دانیم در الگوریتم policy iteration در هر مرحله‌ی iteration مقدار Value همگرا شده صعودی است. یعنی:

$$\forall s \;\; V_\infty^{\pi_{t+1}}(s) \geq V_\infty^{\pi_t}(s) \tag{۳}$$

(آ) ثابت کنید که اگر برای دو policy متوالی π_t و π_{t+1} مقدار $V_\infty^{\pi}(s)$ به ازای هر state برابر شود به policy بهینه رسیده‌ایم.

پاسخ:
در ابتدا باید اشاره کنیم که رابطه‌ي ۲ ایراد دارد و باید به صورت زیر باشد:

$$\pi_{t+1}(s) = \arg\max_a \sum_{s'} P(s'|a, s)[R(s', a, s) + \gamma V_\infty^{\pi_t}(s')]$$

از آنجایی که policy جدید رابطه‌ی بر حسب ارزش‌های قبلی را بیشینه می‌کند، سعی می‌کنیم که رابطه‌ی بلمن را بسازیم:

$$\begin{aligned} V_\infty^{\pi_t}(s) &= V_\infty^{\pi_{t+1}}(s) \\ &= \sum_{s'} P(s'|\pi_{t+1}(s), s)[R(s', \pi_{t+1}(s), s) + \gamma V_\infty^{\pi_{t+1}}(s')] \\ &= \sum_{s'} P(s'|\pi_{t+1}(s), s)[R(s', \pi_{t+1}(s), s) + \gamma V_\infty^{\pi_t}(s')] \\ &\overset{(2)}{=} \max_a \sum_{s'} P(s'|a, s)[R(s', a, s) + \gamma V_\infty^{\pi_t}(s')] \end{aligned}$$

پس $V_\infty^{\pi_t}(s)$ در رابطه‌ی بلمن صدق می‌کند و بهینه است.

(ب) با فرض اینکه مجموعه‌ی actionها یک مجموعه‌ی متناهی مانند A و مجموعه‌ی stateها یک مجموعه‌ی متناهی مانند S بوده، یک کران بالا روی تعداد مراحل Policy Evaluation بیابید و ثابت کنید که الگوریتم Policy Iteration تمام شده و به مقدار بهینه همگرا می‌شود.

پاسخ: تعداد policyهای ممکن متناهی است چرا که برای هر state $|A|$ روش مختلف برای انتخاب action داریم پس طبق اصل ضرب در کل $|A|^{|S|}$ تا policy متفاوت می‌تواند وجود داشته باشد. از آن‌جایی که در هر iteration طبق رابطه‌ی ۳ ارزش مربوط به policy کمتر نمی‌شود و تعداد متناهی policy داریم، مرحله‌ای وجود دارد که ارزش‌ها تغییر نمی‌کند. اگر در یک مرحله ارزش‌ها تغییر نکند، تا ابد ارزش‌ها تغییر نخواهد کرد چرا که الگوریتم مستقل از شماره‌ی iteration است و تنها به مرحله‌ی قبل وابسته است. پس به یک policy همگرا می‌شویم و در بخش قبل دیدیم که اگر ارزش‌ها تغییر نکند policy به دست‌آمده بهینه است.

(ج) با توجه به قسمت‌های بالا و مقایسه با الگوریتم Value Iteration توضیح دهید که Policy Iteration چه مزیتی نسبت به Value Iteration دارد؟

پاسخ: هر دو الگوریتم به policy بهینه همگرا می‌شوند و از این نظر تفاوتی ندارند. در الگوریتم Policy Iteration تعداد مراحل متناهی است و تضمین خوبی برای تشخیص زمان همگرایی الگوریتم داریم، در بخش Policy Evaluation این الگوریتم برعکس Value Iteration از عملگر max استفاده نمی‌شود که به همگرایی سریعتر و سادگی محاسبات الگوریتم بسیار کمک می‌کند و از عملگر هزینه‌بر max تنها در Policy Improvement استفاده می‌شود. این معماری الگوریتم در الگوریتم‌های دیگر این حوزه نیز بسیار کاربرد دارد. در شبیه‌سازی‌ها همگرایی سریعتر این الگوریتم نیز بررسی شده است.

(د) می‌خواهیم الگوریتم Policy Evaluation را اندکی تغییر دهیم. به جای صفر گرفتن $V_0^{\pi_t}(s)$ مقدار آن را به صورت زیر به دست می‌آوریم:

$$V_0^{\pi_{t+1}}(s) = \sum_{s'} P(s'|\pi_{t+1}(s), s)[R(s', \pi_{t+1}(s), s) + \gamma V_\infty^{\pi_t}(s')] \tag{۴}$$

می‌خواهیم فرض اول سوال یعنی عبارت ۳ را ثابت کنیم.
ابتدا ثابت کنید که

$$\forall s \;\; V_0^{\pi_{t+1}}(s) \geq V_\infty^{\pi_t}(s) \tag{۵}$$

حال با فرض صعودی بودن مقادیر در Policy Evaluation :

$$\forall s \;\; V_{k+1}^{\pi_{t+1}}(s) \geq V_k^{\pi_{t+1}}(s) \tag{۶}$$

ثابت کنید که در این حالت تغییریافته‌ی Policy Iteration نیز عبارت ۳ برقرار است. حال ثابت کنید در حالتی که $V_0^{\pi_t}(s)$ تغییر نیافته باشد هم عبارت ۳ برقرار است.

در صورتی که علاقه‌مند هستید می‌توانید تلاش کنید عبارت ۶ را نیز مشابه قسمت ب سوال ۱ یا به کمک نوشتن تساوی بلمن ثابت کنید (نمره‌ای ندارد).

پاسخ:

$$\begin{aligned}
V_\infty^{\pi_t}(s) = & \sum_{s'} P(s'|\pi_t(s), s)[R(s', \pi_t(s), s) + \gamma V_\infty^{\pi_t}(s')] \\
& \leq \max_a \sum_{s'} P(s'|a, s)[R(s', a, s) + \gamma V_\infty^{\pi_t}(s')] \\
& = \sum_{s'} P(s'|\pi_{t+1}(s), s)[R(s', \pi_{t+1}(s), s) + \gamma V_\infty^{\pi_t}(s')] \\
& \overset{(4)}{=} V_0^{\pi_{t+1}}(s)
\end{aligned}$$

حال به کمک رابطه‌ی ۶ داریم:

$$V_\infty^{\pi_t}(s) \leq V_0^{\pi_{t+1}}(s) \leq V_\infty^{\pi_{t+1}}(s)$$

پس صورت مسئله برای حالت تغییر یافته ثابت شد.
برای حالت تغییر نیافته می‌دانیم که تنها یک ارزش بهینه در معادله‌ی بلمن صدق می‌کند که یکتا است. پس با شروع از ارزش اولیه‌ی صفر نیز به یک ارزش که همان ارزش نهایی روش تغییریافته است می‌رسیم که برای آن حالت ثابت کردیم تابع ارزش نسبت به iteration صعودی است.

سوال ۳: Max-Gini (۲۵ نمره)

فرض کنید یک بازوی رباتی داریم که وظیفه دارد تعداد جسم را از یک جعبه بردارد. بازوی رباتی بعد از مدت زمان مشخصی کار خود را خاتمه می‌دهد. ابتدا به دنبال این هستیم که policy ی را پیدا کنیم که مقدار $R = \sum_{t=0}^{H} r_t$ را بیشینه کند. حال فرض کنید که می‌فهمیم شکل اشیا داخل جعبه در آزمایش‌های مختلف تغییر می‌کند. در اینجا برای اینکه policy مطمئن‌تری داشته باشیم قصد داریم از policy های تصادفی استفاده کنیم یعنی برای انتخاب action از توزیع احتمال روی آن استفاده می‌کنیم (مانند epsilon-greedy). یعنی

$$\pi_s(a) = P(a|s)$$

حال به جای جمع پاداش‌ها امید ریاضی جمع آن‌ها برای بیشینه کردن در نظر می‌گیریم.

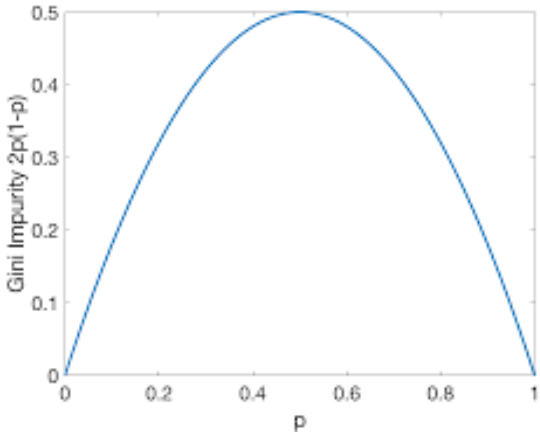
$$R = \mathbb{E}[\sum_{t=0}^{H} r_t] \tag{۷}$$

همچنین دوست داریم که در انتخاب actionهایمان تفاوت تخصیص احتمال کمتر شود یعنی به تعداد کمی از actionها احتمال بالا برای انتخاب شدن و به سایر actionها احتمال کمی نسبت داده نشود. برای تحقق اینکار از شاخصه‌ی Gini استفاده می‌کنیم. روی توزیع احتمال گسسته‌ی P شاخصه‌ی Gini به صورت زیر تعریف می‌شود.:

$$Gini(P) = \sum_k p_k(1 - p_k) \tag{۸}$$

(آ) به طور مختصر توضیح دهید که برای حالتی که دو action داریم چگونه شاخصه‌ی Gini به کم شدن تفاوت احتمال انتخاب شدن action ها کمک می‌کند.

پاسخ:
احتمال کنش اول را p و کنش دوم را $1-p$ در نظر بگیرید. در این حالت شاخصه‌ی gini برابر $Gini(P) = 2p(1-p)$ می‌شود. نمودار مربوط به شاخصه را در شکل ۲ می‌بینیم. تنها زمانی مقدار بیشینه را خواهیم داشت که احتمال دو کنش برابر هم شود.



شکل ۲: تغییرات شاخصه‌ی gini نسبت به احتمال انتخاب کنش اول

حال مسئله را به یافتن یک تابع توزیع احتمال روی actionها تغییر می‌دهیم. همچنین برای سادگی به جای بررسی تمام پاداش‌ها تا بینهایت تنها به پاداش‌های لحظه‌ای توجه می‌کنیم. مسئله به صورت روبه‌رو بازنویسی می‌شود:

$$\max_{\pi_A} \mathbb{E}_{\pi_A}[r(a)] + \beta Gini(\pi_A) \tag{۹}$$

که π_A همان تابع توزیع احتمال روی مجموعه‌ی action هاست که به عنوان policy معرفی می‌کنیم.

(ب) به کمک KKT تابع لاگرانژ مربوط به بهینه‌سازی عبارت بالا را بنویسید. توجه کنید که π_A یک تابع توزیع احتمال است.

پاسخ: مسئله‌ی بهینه‌سازی به صورت روبه‌رو است:

$$\max_{\pi_A}[\mathbb{E}_{\pi_A}[r(a)] + \beta Gini(\pi_A)]$$
$$s.t. \quad \forall a \in A : -\pi_A(a) \leq 0$$
$$1 - \sum_a^A \pi_A(a) = 0$$

تابع لاگرانژ به صورت زیر خواهد بود:

$$\begin{aligned} L(\pi_A, \lambda, \delta) &= \sum_a^A \pi_A(a) r(a) + \beta \sum_a^A \pi_A(a)(1 - \pi_A(a)) - \lambda(1 - \sum_a^A \pi_A(a)) - \sum_a^A \delta_a(-\pi_A(a)) \\ &= \sum_a^A \pi_A(a) r(a) + \beta \sum_a^A \pi_A(a)(1 - \pi_A(a)) + \lambda(\sum_a^A \pi_A(a) - 1) + \sum_a^A \delta_a \pi_A(a) \end{aligned}$$

شروط KKT به صورت زیر است:

- Stationarity: $\frac{\partial L}{\partial \pi_A} = 0$ جواب داشته باشد.
- Primal feasibility:

$$\forall a \in A : -\pi_A(a) \leq 0$$
$$1 - \sum_a^A \pi_A(a) = 0$$

- Dual feasibility:

$$\forall a : \delta_a \geq 0$$

- Complementary Slackness:

$$\forall a : \delta_a \pi_A(a) = 0$$

(ج) با بهینه‌سازی عبارت به دست آمده، توزیع احتمال π_A بهینه را بیابید. فرض کنید، می‌دانیم که به مجموعه‌ی G از actionها احتمال غیر صفر نسبت داده‌شده است.

پاسخ: اگر $\beta = 0$ آنگاه پاسخ مسئله‌ی بهینه‌سازی مشخص است و پاسخ همان کنش با پاداش بیشینه است. پس $\beta > 0$ است.

$$\max_{\pi_A} \min_{\lambda,\delta\geq 0} L(\pi_A, \lambda, \delta)$$

توابع داده‌شده convex هستند همچنین یک نقطه‌ی درونی در ناحیه‌ی feasible وجود دارد(مثلا توزیع یکنواخت روی actionها)
دو شرط بالا شروط کافی برای strong duality هستند (Slater's condition) پس روی این مسئله strong duality داریم. (بررسی خیلی دقیق این موضوع جز اهداف مسئله نیست و در محاسبه‌ی نمره تاثیری نخواهد داشت ولی جهت مطالعه‌ی بیشتر می‌توانید Slater's condition را مطالعه نمایید.)
به خاطر strong duality معادلا بهینه‌سازی پایین را حل می‌کنیم.

$$\min_{\lambda,\delta\geq 0} \max_{\pi_A} L(\pi_A, \lambda, \delta)$$

$$\begin{aligned}
&\Rightarrow \frac{\partial L}{\partial \pi_A(a)} = 0 \\
&\Rightarrow r(a) + \beta(1 - 2\pi_A(a)) + \lambda + \delta_a = 0 \\
&\Rightarrow \pi_A(a) = \frac{r(a) + \beta + \lambda + \delta_a}{2\beta}
\end{aligned}$$

برای هر action مانند a در G داریم که $\pi_A(a) > 0$ پس طبق شرط Complementary Slackness داریم $\delta_a = 0$.

$$\Rightarrow \pi_A(a) = \begin{cases} \frac{r(a)+\beta+\lambda}{2\beta} & \text{if } a \in G \\ 0 & \text{if } a \notin G \end{cases}$$

$$\begin{aligned}
&\sum_a^A \pi_A(a) = 1 \\
&\Rightarrow \sum_a^G \pi_A(a) = 1 \\
&\Rightarrow \sum_a^G \frac{r(a) + \beta + \lambda}{2\beta} = 1 \\
&\Rightarrow \lambda = \frac{2\beta - |G|\beta - \sum_a^G r(a)}{|G|} \\
&\Rightarrow \pi_A(a) = \begin{cases} \frac{r(a)+\beta+\frac{2\beta-|G|\beta-\sum_b^G r(b)}{|G|}}{2\beta} & \text{if } a \in G \\ 0 & \text{if } a \notin G \end{cases} \\
&\Rightarrow \pi_A(a) = \begin{cases} \frac{|G|r(a)+2\beta-\sum_b^G r(b)}{2\beta|G|} & \text{if } a \in G \\ 0 & \text{if } a \notin G \end{cases} \\
&\Rightarrow \pi_A(a) = \begin{cases} \frac{|G|r(a)-\sum_b^G r(b)}{2\beta|G|} + \frac{1}{|G|} & \text{if } a \in G \\ 0 & \text{if } a \notin G \end{cases}
\end{aligned}$$

(د) فرض کنید که مجموعه‌ی G را نمی‌دانستیم. با تبدیل مسئله‌ی بهینه‌سازی به یک مسئله‌ی QP ، روشی برای یافتن مجموعه‌ی G ارائه دهید.

پاسخ:

$$\pi_A(a) = \frac{r(a)+\beta+\lambda+\delta_a}{2\beta}$$
$$\Rightarrow^{PrimalFeasibility} \sum_a^A \frac{r(a)+\beta+\lambda+\delta_a}{2\beta} = 1$$
$$\Rightarrow \sum_a^A \frac{r(a)+\beta+\lambda+\delta_a}{2\beta} = 1$$
$$\Rightarrow \lambda^* = \frac{2\beta - |A|\beta - \sum_b^A r(b) - \sum_b^A \delta_b}{|A|}$$
$$\Rightarrow \pi_A^*(a) = \frac{|A|\delta_a - \sum_b^A \delta_b + |A|r(a) - \sum_b^A r(b)}{2\beta|A|} + \frac{1}{|A|}$$

پس $\pi_A(a)$ و λ بهینه، یک ترکیب خطی از δ_a ها است. با جایگذاری $\pi_A^*(a)$ در رابطه‌ی بهینه‌سازی یک رابطه‌ی QP به دست می‌آوریم(اشاره به فرم چندجمله‌ای نهایی برای کسب نمره کافی می‌باشد):

$$\min_{\lambda,\delta\geq 0} \max_{\pi_A} L(\pi_A, \lambda, \delta)$$
$$\Rightarrow \min_{\delta\geq 0} \sum_a^A \pi_A^*(a)r(a) + \beta\sum_a^A \pi_A^*(a)(1-\pi_A^*(a)) + \lambda^*(\cancel{\sum_a^A \pi_A^*(a) - 1}) + \sum_a^A \delta_a \pi_A^*(a)$$
$$\Rightarrow \min_{\delta\geq 0} \sum_a^A -\beta\pi_A^*(a)^2 + \pi_A^*(a)(r(a)+\delta_a+\beta)$$
$$\Rightarrow \min_{\delta\geq 0} \sum_a^A \pi_A^*(a)(-\beta\pi_A^*(a) + (r(a)+\delta_a+\beta))$$
$$\Rightarrow \min_{\delta\geq 0} \sum_a^A \frac{\lambda^*+r(a)+\beta+\delta_a}{2\beta}(-\beta\frac{\lambda^*-r(a)-\beta-\delta_a}{2\beta})$$
$$\Rightarrow \min_{\delta\geq 0} -\frac{1}{4\beta}\sum_a^A(\lambda^*+r(a)+\beta+\delta_a)(\lambda^*-r(a)-\beta-\delta_a)$$
$$\Rightarrow \max_{\delta\geq 0} \frac{1}{4\beta}\sum_a^A \lambda^{*2} - (r(a)+\beta+\delta_a)^2$$
$$\Rightarrow \max_{\delta\geq 0} \frac{1}{4\beta}(|A|\lambda^{*2} - \sum_a^A (r(a)+\beta+\delta_a)^2)$$
$$\Rightarrow \max_{\delta\geq 0} -\sum_a^A \frac{1}{2\beta}(\frac{\sum_b^A r(b) + |A|\beta - 2\beta}{|A|} + r(a) + \beta)\delta_a + \sum_a^A \frac{1}{4\beta}(\frac{1}{|A|}-1)\delta_a^2 + \sum_a^A\sum_b^A(\frac{1}{2\beta|A|})\delta_a\delta_b$$

با حل QP بالا می‌توانیم مجموعه‌ی G را بیابیم.

جالب است بدانید که اگر به جای شاخصه‌ی Gini از انتروپی استفاده می‌کردیم، انگاه توزیع احتمال ما برای حالت بیشتر از یک مرحله به یک softmax روی Q-value ها تبدیل می‌شد.

سوال ۴: هم‌ارزی نگاه Forward و Backward در TD(λ) (۲۵ نمره)

در این سوال معادل بودن دو نگاه Forward و Backward را در الگوریتم TD(λ) مورد بررسی قرار خواهیم داد. فرض کنید $\Delta V_t^\lambda(s_t)$ میزان تغییر تابع value برای حالت s_t در زمان t را با استفاده از نگاه Forward و $\Delta V_t^{TD}(s)$ میزان تغییر تابع value برای حالت s در زمان t را با توجه به نگاه Backward مشخص کند. در این حالت می‌خواهیم بررسی کنیم آیا مجموع میزان تغییر تابع value برای هر حالت در یک اپیزود در دو حالت گفته شده برابر است یا خیر. به عبارت دیگر هدف بررسی برقراری تساوی زیر است:

$$\forall s \in S, \sum_{t=0}^{T-1} \Delta V_t^{TD}(s) = \sum_{t=0}^{T-1} \Delta V_t^\lambda(s_t) I_{ss_t}$$

(آ) اگر داشته باشیم:

$$\begin{cases} E_{-1}(s) = 0 \\ E_t(s) = \gamma\lambda E_{t-1}(s) + I_{ss_t} \end{cases} \tag{۱۰}$$

که

$$I_{ss_t} = \begin{cases} 0; s \neq s_t \\ 1; s = s_t \end{cases} \tag{۱۱}$$

ثابت کنید:

$$E_t(s) = \sum_{k=0}^{t} (\gamma\lambda)^{t-k} I_{ss_k} \tag{۱۲}$$

پاسخ: طبق رابطه بازگشتی داده شده داریم:

$$\begin{aligned} E_t(s) &= \gamma\lambda E_{t-1}(s) + I_{ss_t} \\ &= \gamma\lambda(\gamma\lambda E_{t-2}(s) + I_{ss_{t-1}}) + I_{ss_t} = (\gamma\lambda)^2 E_{t-2}(s) + (\gamma\lambda) I_{ss_{t-1}} + I_{ss_t} \\ &= (\gamma\lambda)^2(\gamma\lambda E_{t-3}(s) + I_{ss_{t-2}}) + (\gamma\lambda) I_{ss_{t-1}} + I_{ss_t} = (\gamma\lambda)^3 E_{t-3}(s) + (\gamma\lambda)^2 I_{ss_{t-2}} + (\gamma\lambda) I_{ss_{t-1}} + I_{ss_t} \\ &= ... \\ &= (\gamma\lambda)^{t+1} E_{-1}(s) + (\gamma\lambda)^t I_{ss_0} + ... + (\gamma\lambda) I_{ss_t} \\ &= \sum_{k=0}^{t} (\gamma\lambda)^{t-k} I_{ss_k} \end{aligned}$$

(ب) از رابطه قسمت قبل استفاده کنید و اثبات کنید:

$$\sum_{t=0}^{T-1} \Delta V_t^{TD}(s) = \sum_{t=0}^{T-1} \alpha I_{ss_t} \sum_{k=1}^{T-1} (\gamma\lambda)^{k-t} \delta_k \tag{۱۳}$$

پاسخ: سمت چپ را ساده می‌کنیم:

$$\begin{aligned} \sum_{t=0}^{T-1} \Delta V_t^{TD}(s) &= \sum_{t=0}^{T-1} \alpha \delta_t \sum_{k=0}^{t} (\gamma\lambda)^{t-k} \mathcal{I}_{ss_k} \\ &= \sum_{k=0}^{T-1} \alpha \sum_{t=0}^{k} (\gamma\lambda)^{k-t} \mathcal{I}_{ss_t} \delta_k \\ &= \sum_{t=0}^{T-1} \alpha \sum_{k=t}^{T-1} (\gamma\lambda)^{k-t} \mathcal{I}_{ss_t} \delta_k \\ &= \sum_{t=0}^{T-1} \alpha \mathcal{I}_{ss_t} \sum_{k=t}^{T-1} (\gamma\lambda)^{k-t} \delta_k. \end{aligned}$$

(ج) حال سمت راست عبارت اولیه را ساده می‌کنیم. برای این کار ابتدا تساوی زیر را اثبات کنید:

$$\frac{1}{\alpha} \Delta V_t^{\lambda}(s_t) = \sum_{k=t}^{\infty} (\gamma\lambda)^{k-t} (r_{k+1} + \gamma V_t(s_{k+1}) - V_t(s_k)) \tag{۱۴}$$

پاسخ: داریم:

$$\begin{aligned}\frac{1}{\alpha}\Delta V_t^{\lambda}(s_t) &= R_t^{\lambda} - V_t(s_t)\\ &= -V_t(s_t) + (1-\lambda)\lambda^0[r_{t+1} + \gamma V_t(s_{t+1})]\\ &\quad + (1-\lambda)\lambda^1[r_{t+1} + \gamma r_{t+2} + \gamma^2 V_t(s_{t+2})]\\ &\quad + (1-\lambda)\lambda^2[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 V_t(s_{t+3})]\\ &\quad \vdots \qquad \vdots \qquad \vdots \qquad \ddots\end{aligned}$$

می‌دانیم مجموع ضرایب براکت‌ها برابر ۱ می‌شود بنابراین ستون اول را از براکت خارج می‌کنیم یعنی r_{t+1} با ضریب ۱ سپس ستون دوم را با شروع از سطر دوم از براکت خارج می‌کنیم یعنی r_{t+2} با ضریب $\lambda\gamma$ و همینطور ادامه می‌دهیم:

$$\begin{aligned}\frac{1}{\alpha}\Delta V_t^{\lambda}(s_t) &= -V_t(s_t)\\ &\quad + (\gamma\lambda)^0\left[r_{t+1} + \gamma V_t(s_{t+1}) - \gamma\lambda V_t(s_{t+1})\right]\\ &\quad + (\gamma\lambda)^1\left[r_{t+2} + \gamma V_t(s_{t+2}) - \gamma\lambda V_t(s_{t+2})\right]\\ &\quad + (\gamma\lambda)^2\left[r_{t+3} + \gamma V_t(s_{t+3}) - \gamma\lambda V_t(s_{t+3})\right]\\ &\quad \vdots\\ &= \quad (\gamma\lambda)^0\left[r_{t+1} + \gamma V_t(s_{t+1}) - V_t(s_t)\right]\\ &\quad + (\gamma\lambda)^1\left[r_{t+2} + \gamma V_t(s_{t+2}) - V_t(s_{t+1})\right]\\ &\quad + (\gamma\lambda)^2\left[r_{t+3} + \gamma V_t(s_{t+3}) - V_t(s_{t+2})\right]\\ &\quad \vdots\end{aligned}$$

که یعنی:

$$\frac{1}{\alpha}\Delta V_t^{\lambda}(s_t) = \sum_{k=t}^{\infty}(\gamma\lambda)^{k-t}(r_{k+1} + \gamma V_t(s_{k+1}) - V_t(s_k)) \tag{۱۵}$$

(د) عبارتی که بدست آوردیم را می‌توانیم به طور تقریبی به صورت زیر بازنویسی کنیم:

$$\sum_{k=t}^{\infty}(\gamma\lambda)^{k-t}(r_{k+1} + \gamma V_t(s_{k+1}) - V_t(s_k)) \approx \sum_{k=t}^{\infty}(\gamma\lambda)^{k-t}\delta_k \tag{۱۶}$$

اگر تقریب بالا به تساوی تبدیل شود آنگاه عبارت مورد نظر اثبات خواهد شد. توضیح دهید از بین دو حالت online update و offline update کدام یک تساوی را بدست می‌آورد. چرا؟

پاسخ: در حالت off-line update مقدار $V_t(s)$ در تمام t ها یکسان است بنابراین در این حالت تقریب بالا به تساوی تبدیل می‌شود اما در حالت کلی در حالت on-line update این موضوع درست نیست و در گام های زمانی مختلف مقدار تابع تغییر میکند بنابراین در عبارت تقریبا برابر خواهند بود.

سوال ۵: (عملی ۴۵ نمره) Q-learning و آشنایی با Gym

در این تمرین قصد داریم که با محیط Open-AI Gym آشنا شویم و روش Q-learning را روی آن پیاده‌سازی کنیم. در انتها نیز با بررسی روش‌های MC و SARSA تفاوت‌های آن‌ها را می‌یابیم.

(آ) ابتدا در مورد محیط Frozen Lake در این لینک مطالعه کنید.

(ب) نوتبوک داده شده را کامل کنید.

سوال ۶: (عملی ۳۵ نمره) Deep Q-Networks

در این تمرین هدف استفاده از الگوریتم DQN برای آموزش یک عامل در محیط Cart Pole است.

(آ) ابتدا در مورد محیط Cart Pole در این لینک مطالعه کنید.

(ب) نوتبوک داده شده را کامل کنید.