به نام خدا

یادگیری تقویتی

نیم‌سال بهار ۱۴۰۱-۱۴۰۲

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.

(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.

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

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

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.

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

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

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.

(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.

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

(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).

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.

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.

(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.

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

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:

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

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

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

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

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

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

$$\sum_{k=t}^{\infty}(\gamma\lambda)^{k-t}(r_{k+1}+\gamma V_t(s_{k+1})-V_t(s_k)) \approx \sum_{k=t}^{\infty}(\gamma\lambda)^{k-t}\delta_k \quad (۱۵)$$

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؟

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.