تمرین سری چهارم: فرآیندهای تصمیم‌گیری مارکوف

لطفاً به نکات زیر توجه کنید:

- مهلت ارسال این تمرین برای هر دو گروه **۴ آذر ماه** است.
- در صورتی که به اطلاعات بیشتری نیاز دارید می‌توانید به صفحه‌ی تمرین در وب‌سایت درس مراجعه کنید.
- ما همواره هم‌فکری و هم‌کاری را برای حلِ تمرین‌ها به دانشجویان توصیه می‌کنیم. اما هر فرد باید تمامی سوالات را به تنهایی تمام کند و پاسخ ارسالی حتماً باید توسط خود دانش‌جو نوشته‌شده باشد. لطفاً اگر با کسی هم‌فکری کردید نام او را ذکر کنید.
- لطفاً برای ارسال پاسخ‌های خود از راهنمای موجود در صفحه‌ی تمرین استفاده کنید.
- هر سؤالی درباره‌ی این تمرین را می‌توانید در گروه درس مطرح کنید و یا از دستیاران حلِ تمرین بپرسید.

  - آدرس صفحه‌ی تمرین: https://iust-courses.github.io/ai97/assignments/04_mdp
  - آدرس گروه درس: https://groups.google.com/forum/#!forum/ai97

موفق باشید

سؤال‌ها

سوال یک (۲۵ نمره)

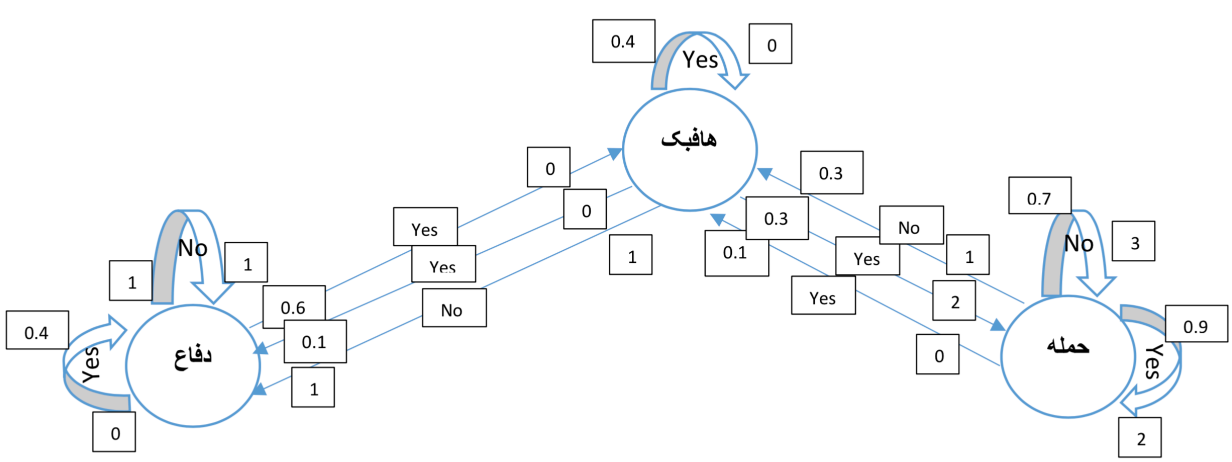
هدف یک عامل شبیه‌ساز فوتبال ۲ بعدی در زمین، کمک کردن به تیم برای رسیدن به پیروزی در بازی است. بیشترین سود برای تیم رسیدن توپ به خط حمله هست اما در این مکان احتمال از دست دادن توپ وجود دارد و برای برگرداندن توپ به خط حمله نیاز به صرف انرژی می‌باشد (در هر مرحله از دست دادن توپ، توپ به یک منطقه عقب‌تر در صورت وجود برمی‌گردد).

این عامل می‌تواند در یکی از ۳حالت ۱- خط دفاع ۲- خط هافبک ۳- خط حمله باشد. در هر یکی از این حالت‌ها ۲ اکشن ۱- حرکت کردن ۲- حرکت نکردن را می‌تواند انجام بدهد. حرکت کردن معادل یک واحد صرف انرژی است و یک امتیاز منفی برای عامل در پی دارد. وقتی عامل در خط حمله قرار می‌گیرد ۳ امتیاز مثبت دریافت می‌کند و در حالت‌های دیگر ۱ امتیاز . برای مثال اگر عامل در خط حمله باشد و در حال حرکت (برای حفظ توپ) پاداش ۲=۱-۳ را دریافت می‌کند.

احتمال موقعیت بعدی عامل بعد هر عمل در هر منطقه از زمین معادل جدول زیر است:

	حرکت کردن	حرکت نکردن
دفاع	۰.۶ هافبک / ۰.۴ دفاع	۱ دفاع
هافبک	۰.۳ حمله / ۰.۶ هافبک / ۰.۱ دفاع	۱ دفاع
حمله	۰.۹ حمله / ۰.۱ هافبک	۰.۷ حمله / ۰.۳ هافبک

الف) MDP این مسئله را رسم کنید.



ب) با استفاده از ضریب تخفیف[1] ۰.۸ و روش «تکرار ارزش[2]»، این MDP را حل کنید. با مجموعه مقادیر صفر باید این کار را آغاز کنید، و راهبرد[3] و مقدار بهینه را نشان دهید.

برای V1 داریم:

دفاع:

No: 1(0+1) = 1

Yes: 0.4(0)+0.6(0)=0 max=1

هافبک:

No: 1(1+0)=1

Yes: 0.1(0)+0.6(0+0.8*0)+0.3(2+0)=0.6 max=1

حمله:

No: 0.7(3+0)+0.3(1+0)=2.4

Yes: 0.9(2+0)+0.1(0+0)=1.8 max=2.4

برای V2 داریم:

دفاع:

Yes: 0.4(0+0.8)+0.6(0+0.8)=0.8

No: 1(1+0.8*1)=1.8 max=1.8

هافبک:

Yes: 0.1(0+0.8)+0.6(0+0.8)+0.2(2+0.8*2.4)=1.73

No: 1(1.8)=1.8 max=1.8

حمله:

Yes: 0.9(2+0.8*2.4)+0.1(0+0.8)=3.528

No: 0.7(3+0.8*2.4)+0.3(1+0.8*1)=3.98 max=3.98

---

[1] Discount Factor
[2] Value Iteration
[3] Policy

به همین ترتیب مقادیر را برای V های بیشتر ادامه داده تا به عددی همگرا شویم.

همچنین در این سوال می توانستید از کد استفاده کنید.

Policy=(دفاع=Not Move, هافبک=Move, حمله=Not Move)

	دفاع	هافبک	حمله
V0	0	0	0
V1	1	1	2.4
V2	1.8	1.8	3.98
...	...	...	...
V25	4.98	5.91	8.67
...	...	...	...
V1000	~5	~6	~8.7

پ) با استفاده از ضریب تخفیف ۰.۸ و روش «تکرار راهبرد[4]» این MDP را حل کنید. حل را با راهبرد اولیه و غیر بهینه‌ی «حرکت نکردن» آغاز کنید. راهبرد و مقدار بهینه را نشان دهید.

دفاع:

V^*_1(دفاع)=1(1+0)=1   policy(دفاع)=arg max (not move(1+0.8),move(0.6+0.8+0.6+(0.8(0))))

هافبک

V^*_1(هافبک) = 1*(1+0)=1 policy(هافبک)=arg max (not move(1.8), move(0.6*0.8+0.3(2+0.8*2.4)))

حمله:

V^*_1(حمله) = 0.7*0.3+0.3(1)=2.4policy(حمله)=arg max (not move(0.3+(1+0.8)+0.7(3+0.8(2.4))), move(0.1(0.8+1)+0.9(2+0.8(2.4)))

policy(1) = (دفاع=Not move, هافبک=Not move, حمله=Not move)

V^*_2(دفاع) = 1.8   policy(دفاع) =arg max(not move(2.44), move(1.44))

V^*_2(هافبک) = 1.8   policy(هافبک)=arg max(not move(2.44), move(2.564))

[4] Policy Iteration

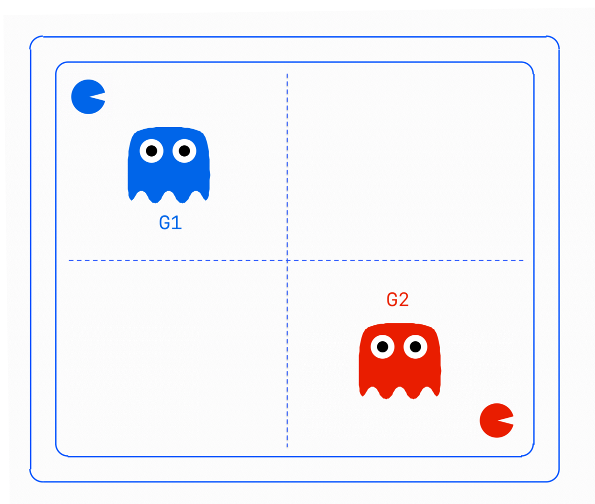
V*₂(حمله) = 3.984   policy(حمله)=arg max(not move(5.063), move(4.812))

policy(2) = (دفاع=Not move, هافبک=move, حمله=Not move)

با محاسبه‌ی policy(3) نیز به همین مقدار policy(2) خواهیم رسید و مقادیر تقریبا همگرا می‌شوند و policy* همان policy(2) است

سوال دو (۲۵ نمره)

در این سوال می‌خواهیم از MDP در شکلی از بازی پک‌من که یک بازی دو نفره‌ی نوبتی با مجموع صفر است استفاده کنیم. فرض کنید در یک صفحه‌ی ۲در ۲، دو روح G1 و G2 به شکل زیر قرار دارند. وظیفه‌ی دو روح خوردن پک‌من طرف مقابل است. مکان پک‌من‌ها در نقشه مشخص شده‌اند و با رسیدن هر روح به پک‌من مورد نظرش آن را می‌خورد و بازی تمام می‌شود. هم‌چنین دو روح هم نمی‌توانند هم‌زمان در یک خانه از جدول باشند. هر روح فقط می‌تواند در جهت عمودی و یا افقی حرکت کند. R(s) تابعی است که پاداش بودن در هر حالت را از دید روح اول (G1) مشخص می‌کند به این صورت که اگر روح اول زودتر پک‌من طرف مقابل را بخورد مقدار (+۱) و اگر روح دوم زودتر پک‌من طرح مقابل را بخورد مقدار (−۱) و در غیر این صورت صفر برگردانده می‌شود. شروع بازی با روح اول است.



الف) در حالت کلی اگر $V_{G1}(s)$ ارزش حالت s وقتی نوبت حرکت G1 باشد و هم‌چنین $V_{G2}(s)$ ارزش حالت s وقتی نوبت حرکت G2 باشد، معادلات بِل‌من[5] را برای V_{G1} و V_{G2} بنویسید (دقت کنید که تمامی محاسبات امتیاز و ارزش‌ها بر اساس دید روح اول از بازیست).

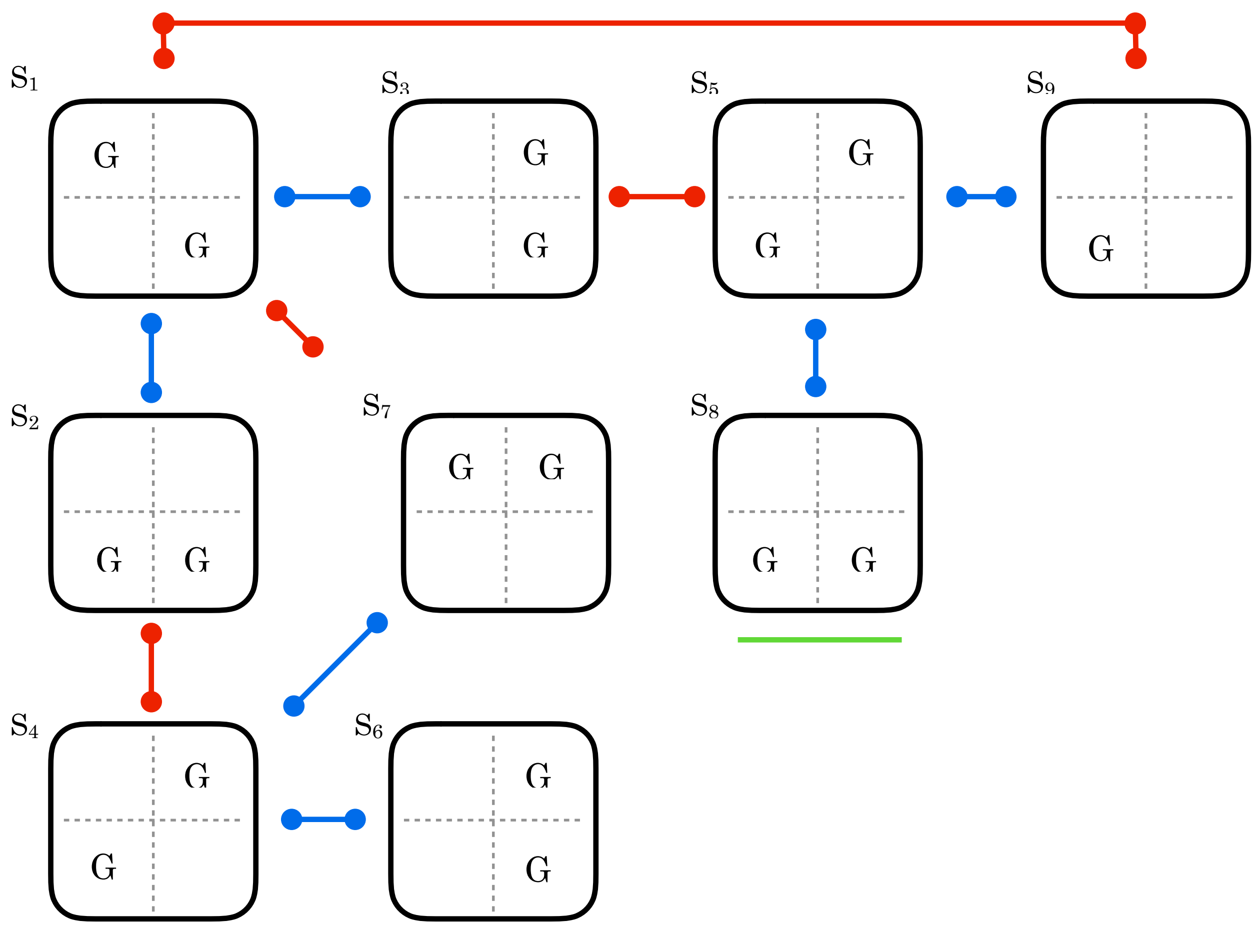
[5] Bellman Equations

$$V_{G_1}(s) = \max_a \sum T(s,a,s')\ [R(s,a,s') + \lambda V_{G_2}(s')]$$
$$V_{G_2}(s) = \min_a \sum T(s,a,s')\ [R(s,a,s') + \lambda V_{G_1}(s')]$$

ب) توضیح دهید چگونه با استفاده از این دو معادله‌ای که به دست آوردید می‌توان الگوریتم «تکرار ارزش» را به صورت دو نفره انجام داد. ضمناً شرط پایان مناسب را هم تعیین کنید.

برای انجام الگوریتم تکرار ارزش، هر کدام از معادلات قسمت بالا را به فرمول آپدیت بل‌من تبدیل می‌کنیم، سپس آن‌ها را به پشت‌سر هم (فرمول۱، فرمول۲، فرمول ۱ و ...) رو تمامی استیت‌ها اعمال می‌کنیم. شرط پایان آن است که ارزش‌های یک روح، با ارزش‌های همان روح در مرحله‌ی قبل برابر باشد

پ) نقشه‌ی حالات MDP را برای این بازی رسم کنید و هم‌چنین امتیاز هر حالت را برای آن بنویسید.



ت) (امتیازی / ۱۰ نمره) حال الگوریتم «تکرار ارزش»ی که در قسمت (ب) به دست آوردید را برای این بازی اعمال کنید و راهبرد بهینه را به دست آورید.

سوال سه (۳۰ نمره)

تا به این‌جا از الگوریتم «تکرار ارزش» برای حل MDPها استفاده کردیم و می‌دانیم با استفاده از آن حتماً به جواب بهینه می‌رسیم. اما در این سوال می‌خواهیم دلیل هم‌گرایی این الگوریتم را بیابیم. در ریاضی مفهومی وجود دارد به نام نگاشت انقباضی[۶]. اگر بخواهیم حدودی آن را تعریف کنیم، این نگاشت یک تابع با یک ورودیست که اگر آن را به طور مجزا روی دو عدد اعمال کنیم، این تابع خروج‌هایی برای این دو تولید می‌کند که به هم نزدیکترند. به طور مثال «تقسیم بر دو» یک نگاشت انقباضیست. توجه کنید که این نوع عملگر یک نقطه‌ی ثابت هم دارد که در اثر اعمال تابع، بدون تغییر می‌ماند (مثلاً عدد صفر در نگاشت «تقسیم بر دو» یک نقطه‌ی ثابت است).

این نگاشت دو خاصیت مهم دارد: ۱- تنها یک نقطه‌ی ثابت دارد و ۲- اگر به صورت حدی آن را بر روی یک ورودی مرتباً اعمال کنیم به به نطقه‌ی ثابت آن می‌رسیم (چرا؟) بنابراین اگر ثابت کنیم رویه‌ی آپدیت بِل‌من[۷] از این نوع است هم‌گرایی و رسیدن به جواب یکتا برای آن اثبات می‌شود.

اگر V_i یک بردار باشد که هر خانه‌ی آن بیان‌گر یک ارزش در مرحله‌ی iام الگوریتم تکرار ارزش باشد، و هم‌چنین اگر رویه‌ی بل‌من را به صورت یک عملگر ببینیم که به صورت همزمان روی تمام خانه‌ی این بردار اعمال می‌شود، رابطه‌ی آپدیت بل‌من را می‌توان به صورت زیر نوشت:

$$V_{i+1} \leftarrow B[V_i]$$

به علاوه برای اثبات، به روشی برای اندازه‌گیری «فاصله» بین دو بردار ارزش نیاز داریم، برای این منظور از MaxNorm استفاده می‌کنیم. اگر این عملگر روی یک بردار اعمال شود بزرگ‌ترین قدر مطلق از مقادیر آن را برمی‌گرداند:

$$||V|| = \ max_s \ |V_s|$$

با این تعریف، بیشترین تفاوت بین هر دو المان دو بردار برابر با $||V - V'||$ خواهد شد، در ادامه اگر با فرض اینکه V_i و V_i' دو بردار ارزش باشد برای رسیدن به انقباضی بودن رابطه‌ی بل‌من شما باید رابطه زیر را اثبات کنید:

$$||B[V_i] - B[V_i']|| \leq \gamma ||V_i - V_i'||$$

راهنمایی: ابتدا اثبات کنید به ازای هر دو تابع g و f رابطه‌ی زیر برقرار است، سپس رابطه بل‌من را در نامساوی بالا قرار دهید.

$$|max_a \ f(a) - max_a \ g(a)| \leq max_a \ |f(a) - g(a)|$$

$$\begin{aligned}
|\max_a f(a) - \max_a g(a)| &= \max_a f(a) - \max_a g(a) \qquad \text{(by assumption)} \\
&= f(a^*) - \max_a g(a) \\
&\leq f(a^*) - g(a^*) \\
&\leq \max_a |f(a) - g(a)| \qquad \text{(by definition of max)}
\end{aligned}$$

$$
\begin{aligned}
|(B\,U_i - B\,U_i')(s)| \;&=\; |R(s) + \gamma \max_{a\in A(s)} \sum_{s'} P(s' \mid s,a) U_i(s') \\
&\quad - R(s) - \gamma \max_{a\in A(s)} \sum_{s'} P(s' \mid s,a) U_i'(s')| \\
&=\; \gamma | \max_{a\in A(s)} \sum_{s'} P(s' \mid s,a) U_i(s') - \max_{a\in A(s)} \sum_{s'} P(s' \mid s,a) U_i'(s')| \\
&\leq\; \gamma \max_{a\in A(s)} | \sum_{s'} P(s' \mid s,a) U_i(s') - \sum_{s'} P(s' \mid s,a) U_i'(s')| \\
&=\; \gamma | \sum_{s'} P(s' \mid s,a^*(s)) U_i(s') - \sum_{s'} P(s' \mid s,a^*(s)) U_i'(s')| \\
&=\; \gamma | \sum_{s'} P(s' \mid s,a^*(s)) (U_i(s') - U_i'(s')|
\end{aligned}
$$

بعد ا ز اعمال MaxNorm

$$
\begin{aligned}
||B\,U_i - B\,U_i')|| \;&=\; \max_{s} |(B\,U_i - B\,U_i')(s)| \\
&\leq\; \gamma \max_{s} | \sum_{s'} P(s' \mid s,a^*(s)) (U_i(s') - U_i'(s'))| \\
&\leq\; \gamma \max_{s} |U_i(s) - U_i'(s)| = \gamma ||U_i - U_i'||
\end{aligned}
$$

سوال چهار (۲۰ نمره)

در یک فرآیند تصمیم‌گیری مارکوف با تابع پاداش R(s,a) می‌توان راهبرد بهینه $\pi^*(s)$ را یافت. برای چه مقادیری از c، اعمال تغییر در تابع پاداش، همان راهبرد بهینه قبلی را به ما می دهد؟ هر قسمت را اثبات و یا با یک مثال نقض رد کنید.

الف) R'(s,a) = R(s,a) × c

با ضرب مقدار c در R(s,a)، Q* به شکل:

Q*(s) = cR(s,a)+ꝺV*(s)

خواهد شد.

اگر داشته باشیم:

$C=5$

Action 1:	$T=0.5$	Action 2:	$T=0.5$
	$V=2$		$V=-2$
	$R=2$		$R=5$

$$2+2=4 \quad > \quad 5-2=3$$

$$C=5 \rightarrow 2(5)+2=12 \quad < \quad 5(5)-2=23$$

با مثال نقض این تساوی برقرار نیست.

ب) $R'(s,a) = R(s,a) + c$

$$R_1(s,a) + \gamma_1 V_1 \quad > \quad R_2(s,a)+\gamma_2 V_2$$

$$R_1(s,a) + \gamma_1 V_1 + C \quad > \quad R_2(s,a)+\gamma_2 V_2 + C$$

$$R_1`(s,a) + \gamma_1 V_1 \quad > \quad R_2`(s,a)+\gamma_2 V_2$$

پس این تساوی برقرار است.