به نام خدا

تمرین پنجم درس هوش مصنوعی

استاد: دکتر رهبان

نویسنده: سید علیرضا میررکنی

شماره دانشجویی: ۴۰۱۱۰۶۶۱۷

دانشکده مهندسی کامپیوتر دانشگاه صنعتی شریف – بهار ۱۴۰۳

سوال ۱:

الف) نادرست. MDP زیر را در نظر بگیرید که در آن $state$ ها، خانه های یک نوار با $T+3$ خانه هستند:

s_{T+3}	s_1	...	s_{T+1}	s_{T+2}

همچنین برای این MDP داریم (دقت کنید که سمت راست s_1، استیت s_2 قرار گرفته است):

$$S = \{s_1, \dots, s_{T+1}, s_{T+2}, s_{T+3}\}$$

$$A = \{r\ (move\ to\ right), l(move\ to\ left)\}$$

$$T(s_i, D, s_j) = \begin{cases} 1 & D = r \wedge 1 \le i \le T+1 \wedge j = i+1 \\ 1 & D = l \wedge i = 1 \wedge j = T+3 \\ 0 & O.W. \end{cases}$$

$$R(s_i, D, s_j) = \begin{cases} 10 & D = r \wedge j = i+1 = T+2 \\ 1 & D = l \wedge i = 1 \wedge j = T+3 \\ 0 & O.W. \end{cases}$$

$$(start\ state, final\ states) = (s_1, \{s_{T+2}, s_{T+3}\})$$

به وضوح در این MDP داریم: (برای همه استیت های غیر ترمینال، سیاست بهینه حرکت به سمت راست است)

$$\forall 1 \le i \le T+1; V^*(s_i) = 10 \Longrightarrow \pi^*(s) = r$$

حال دقت کنید که در حین اجرای $value\ iteration$ برای این MDP خواهیم داشت: ($\gamma = 1$)

$$\forall 1 \le i \le T+1; V_0(s_i) = 0$$

$$V_{k+1}(s_i) = \max_a Q(s_i, a) = Q(s_i, r) = \sum_{1 \le j \le T+1} T(s_i, r, s_j)\left(R(s_i, r, s_j) + \gamma V_k(s_j)\right)$$

بنابراین می توان نوشت:

$$\forall 1 \le i \le T+1; V_k(s_i) = \begin{cases} 10 & k \ge T+2-i \\ 1 & i = 1 \wedge 1 \le k \le T \\ 0 & O.W. \end{cases}$$

بنابراین در T مرحله ابتدایی اجرای $value\ iteration$ بر روی این MDP، سیاست به دست آمده برای استیت s_1 حرکت به سمت چپ می باشد (با $value$ برابر 1) که با سیاست بهینه برای این استیت (حرکت به سمت راست) یکسان نمی باشد. پس این عبارت نادرست است.

ب) نادرست. در الگوریتم $Q-learning$ که یکی از روش های $passive\ RL$ می باشد، ما در ابتدا یک سیاست دلخواه (نه لزوما سیاست بهینه؛ به طور مثال این سیاست می تواند انتخاب تصادفی یک کنش در هر استیت باشد) را در اختیار $agent$ قرار می دهیم تا با استفاده از آن، مقدار $Q-value$ ها را یاد بگیرد. سپس با استفاده از $Q-value$ های به دست آمده، سیاست بهینه را به دست می آوریم (به عبارت دیگر پس از به دست آمدن $Q-value$ ها توسط $agent$، خواهیم داشت $V^*(s)=\max_a Q(s,a)$ و $\pi(s)=\underset{a}{\operatorname{argmax}}\, Q(s,a)$).

ج) نادرست. می دانیم که در $Q-learning$ مقادیر به شکل زیر بروزرسانی می شوند:

$$Q^\pi(s,a)=(1-\alpha)Q^\pi(s,a)+\alpha(sample)$$

به وضوح اگر قرار دهیم $\alpha=1$، خواهیم داشت:

$$Q^\pi(s,a)=sample$$

به عبارت دیگر، در هر بروزرسانی نمونه گیری های گذشته و مقداری که تا بحال برای $Q^\pi(s,a)$ محاسبه شده است را فراموش می کنیم و تنها نمونه جدید را در نظر می گیریم. این عملکرد، منجر به عدم همگرایی $Q-value$ ها به مقادیری ثابت و یادگیری نادرست آن ها توسط عامل یادگیرنده می شود. به عبارت دیگر برای یادگیری صحیح $agent$، باید $0 \ll \alpha < 1$ انتخاب شود (انتخاب α به این شکل، منجر به فراموشی تقریبی نمونه های اولیه که نادقیق بودند شده و سرعت و دقت یادگیری عامل یادگیرنده را افزایش می دهد. دقت کنید که با کاهش تدریجی α، می توان همگرایی الگوریتم را نیز تضمین نمود).

د) درست. می دانیم که پیچیدگی زمانی $value\ iteration$ در هر $iteration$ برابر $O(|S|^2|A|)$ می باشد؛ در حالی که در $policy\ iteration$ پیچیدگی زمانی هر مرحله $policy\ evaluation$ برابر $O(|S|^2)$ بوده و پیچیدگی زمانی هر مرحله $policy\ improvement$ برابر $O(|S|^2|A|)$ است. حال اگر $|S| \gg |A|$ باشد، می توان در عبارت $|S|^2|A|$ از مقدار $|A|$ صرف نظر کرد (چرا که در مقایسه با $|S|^2$ ناچیز است) و بنابراین خواهیم داشت $O(|S|^2|A|) \approx O(|S|^2)$. در نتیجه با فرض $|S| \gg |A|$، پیچیدگی زمانی اجرای هر $iteration$ در $value\ iteration$ و $policy\ iteration$ تقریبا با یکدیگر برابر خواهند شد (هر دو برابر $O(|S|^2)$ می شوند).

۵) نادرست. MDP زیر را در نظر بگیرید که در آن $state$ ها، خانه های یک نوار با ۴ خانه هستند:

s_4	s_3	s_1	s_2

همچنین برای این MDP داریم: (دقت کنید تمامی حالاتی که برای T و R آورده نشده اند، خروجی صفر دارند)

$$S = \{s_1, s_2, s_3, s_4\}$$

$$A = \{r\ (move\ to\ right), l\ (move\ to\ left\}$$

$$T = \{\big((s_1, r, s_2), 1\big), \big((s_1, l, s_3), 1\big), \big((s_3, l, s_4), 1\big)\}$$

$$R = \{\big((s_1, r, s_2), 1\big), \big((s_1, l, s_3), 1\big), \big((s_3, l, s_4), -1\big)\}$$

$$(start\ state, final\ states) = (s_1, \{s_2, s_4\})$$

$$\gamma = \frac{1}{2}$$

به وضوح در این MDP، سیاست بهینه برای استیت s_1 حرکت به سمت راست می باشد؛ چرا که اگر $agent$ در این استیت به سمت راست حرکت کند مقدار $utility$ در نهایت برابر $R(s_1, r, s_2) = 1$ می شود، اما اگر به سمت راست حرکت کند مقدار $utility$ نهایی برابر $R(s_1, l, s_3) + \gamma R(s_3, l, s_4) = 1 + \frac{1}{2} \cdot (-1) = \frac{1}{2}$ خواهد شد که کمتر می باشد.

حال دقت کنید که اگر همه پاداش ها را با عدد ثابت $c = 2$ جمع کنیم، تابع پاداش به شکل زیر خواهد شد:

$$R = \{\big((s_1, r, s_2), 2\big), \big((s_1, l, s_3), 2\big), \big((s_3, l, s_4), 1\big)\}$$

به وضوح در این MDP، سیاست بهینه برای استیت s_1 حرکت به سمت چپ می باشد؛ چرا که اگر $agent$ در این استیت به سمت چپ حرکت کند مقدار $utility$ در نهایت برابر $R(s_1, l, s_3) + \gamma R(s_3, l, s_4) = 2 + \frac{1}{2} \cdot 1 = \frac{5}{2}$ می شود، اما اگر به سمت راست حرکت کند مقدار $utility$ نهایی برابر $R(s_1, r, s_2) = 2$ خواهد شد که کمتر است.

بنابراین مثالی یافت شد که در آن گزاره داده شده نادرست می باشد.

سوال ۲:

الف) می دانیم که در $value\ iteration$، مقادیر به شکل زیر بروزرسانی می شوند:

$$\forall s \in S; V_0(s) = 0$$

$$V_{k+1}(s) = \max_a \sum_{s'} T(s,a,s')\big(R(s,a,s') + \gamma V_k(s')\big)$$

با توجه به روابط بالا، مراحل بروزرسانی $value$ استیت ها (ی غیر ترمینال) را می نویسیم (در هر کدام از max ها، آرگومان اول مربوط به کنش بالا رفتن، آرگومان دوم مربوط به کنش راست رفتن، آرگومان سوم مربوط به کنش پایین رفتن و آرگومان چهارم مربوط به کنش چپ رفتن می باشد).

مرحله اول:

$$V_1\big((1,1)\big) = \max\begin{Bmatrix} 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 0)+0.1\times(0+0.9\times 0) \end{Bmatrix} = 0$$

$$V_1\big((2,1)\big) = \max\begin{Bmatrix} 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 0)+0.1\times(0+0.9\times 0) \end{Bmatrix} = 0$$

$$V_1\big((2,2)\big) = \max\begin{Bmatrix} 0.8\times(0+0.9\times 0)+0.1\times(5+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(5+0.9\times 0)+0.1\times(0+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(5+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 0)+0.1\times(0+0.9\times 0) \end{Bmatrix} = 4$$

$$V_1\big((1,2)\big) = \max\begin{Bmatrix} 0.8\times(0+0.9\times 0)+0.1\times(-5+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(-5+0.9\times 0)+0.1\times(0+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(-5+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 0)+0.1\times(0+0.9\times 0) \end{Bmatrix} = 0$$

مرحله دوم:

$$V_2\big((1,1)\big) = \max\begin{Bmatrix} 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 0)+0.1\times(0+0.9\times 0) \end{Bmatrix} = 0$$

$$V_2((2,1)) = \max\begin{Bmatrix} 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 4)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 4)+0.1\times(0+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 4)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 0)+0.1\times(0+0.9\times 0) \end{Bmatrix} = 2.88$$

$$V_2((2,2)) = \max\begin{Bmatrix} 0.8\times(0+0.9\times 4)+0.1\times(5+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(5+0.9\times 0)+0.1\times(0+0.9\times 4)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(5+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 4)+0.1\times(0+0.9\times 0) \end{Bmatrix} = 4.36$$

$$V_2((1,2)) = \max\begin{Bmatrix} 0.8\times(0+0.9\times 4)+0.1\times(-5+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(-5+0.9\times 0)+0.1\times(0+0.9\times 4)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(-5+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 4)+0.1\times(0+0.9\times 0) \end{Bmatrix} = 2.38$$

مرحله سوم:

$$V_3((1,1)) = \max\begin{Bmatrix} 0.8\times(0+0.9\times 2.88)+0.1\times(0+0.9\times 2.38)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 2.38)+0.1\times(0+0.9\times 2.88)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 2.38)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 2.88)+0.1\times(0+0.9\times 0) \end{Bmatrix}$$

$$= 2.2878\ (up)$$

$$V_3((2,1)) = \max\begin{Bmatrix} 0.8\times(0+0.9\times 2.88)+0.1\times(0+0.9\times 4.36)+0.1\times(0+0.9\times 2.88) \\ 0.8\times(0+0.9\times 4.36)+0.1\times(0+0.9\times 2.88)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 4.36)+0.1\times(0+0.9\times 2.88) \\ 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 2.88)+0.1\times(0+0.9\times 0) \end{Bmatrix}$$

$$= 3.3984\ (right)$$

$$V_3((2,2)) = \max\begin{Bmatrix} 0.8\times(0+0.9\times 4.36)+0.1\times(5+0.9\times 0)+0.1\times(0+0.9\times 2.88) \\ 0.8\times(5+0.9\times 0)+0.1\times(0+0.9\times 4.36)+0.1\times(0+0.9\times 2.38) \\ 0.8\times(0+0.9\times 2.38)+0.1\times(5+0.9\times 0)+0.1\times(0+0.9\times 2.88) \\ 0.8\times(0+0.9\times 2.88)+0.1\times(0+0.9\times 4.36)+0.1\times(0+0.9\times 2.38) \end{Bmatrix}$$

$$= 4.6066\ (right)$$

$$V_3((1,2)) = \max\begin{Bmatrix} 0.8\times(0+0.9\times 4.36)+0.1\times(-5+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(-5+0.9\times 0)+0.1\times(0+0.9\times 4.36)+0.1\times(0+0.9\times 2.38) \\ 0.8\times(0+0.9\times 2.38)+0.1\times(-5+0.9\times 0)+0.1\times(0+0.9\times 0) \\ 0.8\times(0+0.9\times 0)+0.1\times(0+0.9\times 4.36)+0.1\times(0+0.9\times 2.38) \end{Bmatrix}$$

$$= 2.6392\ (up)$$

دقت کنید که $value$ برای $terminal\ state$ ها همواره برابر صفر در نظر گرفته می شود، به عبارت دیگر داریم:

$$V_0((2,3)) = V_1((2,3)) = V_2((2,3)) = V_3((2,3)) = 0$$

$$V_0((1,3)) = V_1((1,3)) = V_2((1,3)) = V_3((2,3)) = 0$$

ب) روش های متفاوتی برای یادگیری سیاست بهینه وجود دارد. این روش ها عبارتند از:

- $model-based\ learning$: در روش های $model-based$، $agent$ با استفاده از مشاهدات به دست آمده از محیط مقدار $transition\ probability$ ها و $reward$ ها را تخمین می زند ($\hat{T}$ و $\hat{R}$) و با استفاده از این مقادیر، سیاست بهینه را با استفاده از روش های مورد استفاده برای MDP ها (مانند $value\ iteration$ و $policy\ iteration$) می یابد.
- $passive\ RL$: در این روش ها، یک سیاست اولیه در اختیار $agent$ قرار می گیرد و $agent$ با استفاده از این سیاست، مقدار $value$ استیت ها (با استفاده از میانگین گیری در $episode$ های مختلف و یا با استفاده از $temporal\ difference\ learning$) و یا مقدار $Q-value$ ها ($Q-learning$) را یاد می گیرد. در نهایت با استفاده از مقادیری که یاد گرفته است، با اعمال $expectimax$ یک مرحله ای، $policy\ improvement$ را انجام می دهد و سیاست بهینه را می یابد. دقت کنید که در این روش ها، $agent$ در حین یادگیری همواره از همان سیاست اولیه استفاده می کند و سیاست خود را بهبود نمی بخشد.
- $active\ RL$: در این روش ها، همانند روش های $passive\ RL$، یک سیاست اولیه در اختیار $agent$ قرار می گیرد، با این تفاوت که در $active\ RL$ بر خلاف $passive\ RL$، عامل یادگیرنده همزمان با یادگیری سیاست مورد استفاده خود را نیز بهبود می بخشد.

یکی از این روش ها، $Q-learning$ با استفاده از $greedy-\varepsilon$ می باشد. در این روش، عامل یادگیرنده در هر $time\ step$ به احتمال ε عمل $exploration$ (انجام یک کنش تصادفی) را انجام می دهد و به احتمال $1-\varepsilon$، عمل $exploitaion$ (حریصانه عمل کردن و انجام بهترین کنشی که تاکنون یافت شده است) را انجام خواهد داد. یکی دیگر از این روش ها، $Q-learning$ با استفاده از $exploration\ function$ می باشد.

به طور کلی روش های $active\ RL$ نسبت به روش های $passive\ RL$، $regret$ کمتری دارند و در $time\ step$ های کمتری به سیاست بهینه $converge$ می کنند.

ج) می دانیم که در $temporal\ difference-learning$، مقادیر به شکل زیر بروزرسانی می شوند:

$$\forall s\in S; V^{\pi}(s)=0$$
$$V^{\pi}(s)=V^{\pi}(s)+\alpha\big(sample-V^{\pi}(s)\big)$$
$$sample=R(s,\pi(s),s')+\gamma V^{\pi}(s)$$

با توجه به روابط بالا، مراحل بروزرسانی $value$ استیت ها را می نویسیم:

نمونه اول ($(1,1)\longrightarrow(1,2)\longrightarrow(1,3)$):

$$(1,1)\longrightarrow(1,2)\Longrightarrow V^{\pi}\big((1,1)\big)=0+0.1(0+0.9\times 0-0)=0$$

$$(1,2)\longrightarrow(1,3)\Longrightarrow V^{\pi}\big((1,2)\big)=0+0.1(-5+0.9\times 0-0)=-0.5$$

نمونه دوم ($(1,1)\longrightarrow(1,2)\longrightarrow(2,2)\longrightarrow(2,3)$):

$$(1,1)\longrightarrow(1,2)\Longrightarrow V^{\pi}\big((1,1)\big)=0+0.1(0+0.9\times -0.5-0)=-0.045$$

$$(1,2)\longrightarrow(2,2)\Longrightarrow V^{\pi}\big((1,2)\big)=-0.5+0.1\big(0+0.9\times 0-(-0.5)\big)=-0.45$$

$$(2,2)\longrightarrow(2,3)\Longrightarrow V^{\pi}\big((2,2)\big)=0+0.1(5+0.9\times 0-0)=0.5$$

بنابراین مقادیر نهایی استیت ها، به شکل زیر خواهد بود:

$$V^{\pi}\big((1,1)\big)=-0.045$$
$$V^{\pi}\big((1,2)\big)=-0.45$$
$$V^{\pi}\big((2,2)\big)=0.5$$
$$V^{\pi}\big((2,1)\big)=V^{\pi}\big((1,3)\big)=V^{\pi}\big((2,3)\big)=0$$

سوال ۳:

الف) عامل در محیط $grid\ world\ 4\times 4$، با شروع از خانه $(0,0)$، در هر مرحله با توجه به $sample$ ای که در محیط مشاهده می کند، مقدار $Q-value$ استیت قبلی را بروزرسانی می کند. این بروزرسانی، با توجه به $Bellman\ Equations$ که به شکل زیر هستند، انجام می شود:

$$\forall s\in S; V^*(s)=\max_a Q^*(s,a)$$
$$\forall s\in S, a\in A; Q^*(s,a)=\sum_{s'}T(s,a,s')\big(R(s,a,s')+\gamma V^*(s')\big)$$
$$\Longrightarrow Q^*(s,a)=\sum_{s'}T(s,a,s')\left(R(s,a,s')+\gamma\max_{a'}Q^*(s',a')\right)$$

برای پیدا کردن مقادیر تابع Q، می توان از $value\ iteration$ به شکل زیر استفاده کرد: (یافتن نقطه ثابت Q^*)

$$Q_k(s,a)=\sum_{s'}T(s,a,s')\left(R(s,a,s')+\gamma\max_{a'}Q_{k-1}(s',a')\right)$$

دقت کنید که می توان اثبات کرد که در این روش $iterative$، $Q-value$ ها در نهایت به Q^* همگرا می شوند.

در روش های RL، عامل یادگیرنده مستقیما به توابع T و R دسترسی ندارد و به همین دلیل نمی تواند مستقیما از $value\ iteration$ استفاده نماید. در نتیجه، عامل یادگیرنده با استفاده از نمونه گیری ($sampling$) به یادگیری $Q-value$ ها می پردازد؛ به این شکل که با استفاده از معادلات بالا و $sample$ به شکل زیر استفاده می کند:

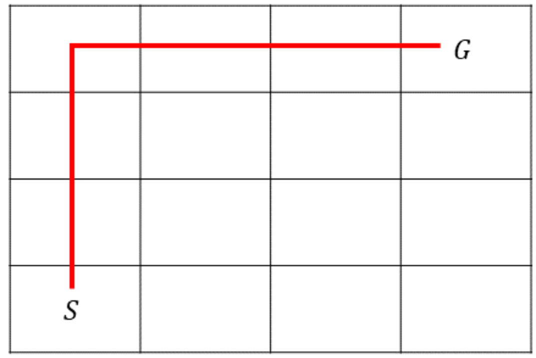
$$Q(s,a)=Q(s,a)+\alpha\big(sample-Q(s,a)\big)$$
$$sample=R(s,a,s')+\gamma\max_{a'}Q(s',a')$$

مقدار تمامی Q ها در ابتدا صفر می باشد. در این روش از این نکته استفاده می کنیم که برای تخمین زدن میانگن (امید ریاضی) یک جامعه آماری، نیازی به بدست آوردن توزیع احتمال دقیق روی آن جامع آماری نیست و می توان با استفاده از میانگین نمونه، پارامتر جامعه را به شکل نقطه ای تخمین زد. به همین خاطر می توان بدون داشتن مقادیر دقیق احتمالات انتقال و تنها با میانگین گیری (وزن دار) از نمونه های مشاهده شده، مقدار $Q-value$ ها را محاسبه نمود.

در نهایت، توجه کنید که برای اجرای $Q-learning$ در این $grid\ 4\times 4$، باید در ابتدا یک سیاست در اختیار عامل قرار دهیم تا با استفاده از آن کنش ها را انتخاب کند. در ادامه عامل با نمونه گیری از محیط، مقدار $Q-value$ خانه های $grid$ را بروزرسانی می کند (در هر مرحله از خانه ای که در آن قرار گرفته به یکی از خانه های مجاور می رود و مقدار $Q-value$ خانه قبلی را بروزرسانی می کند) در نهایت و پس آن که عامل مقدار $Q-value$ ها را یاد گرفت، با پیدا کردن $\operatorname{argmax}_a Q(s,a)$ برای هر استیت s سیاست بهینه را می یابیم.

ب) فرض می کنیم که عامل یادگیرنده، دو مسیر زیر را از خانه شروع به خانه هدف طی می کند.

مسیر ۱



مسیر ۲

می دانیم که در $Q-learning$، مقادیر به شکل زیر بروزرسانی می شوند: ($Q-value$ ها در ابتدا صفر هستند)

$$Q(s,a) = Q(s,a) + \alpha\big(sample - Q(s,a)\big)$$
$$sample = R(s,a,s') + \gamma \max_{a'} Q(s',a')$$

با توجه به روابط بالا، مراحل بروزرسانی $Q-value$ ها را می نویسیم.

مسیر ۱:

$$Q\big((0,0),\rightarrow\big) = 0 + 0.1(-1 + 0.9 \times 0 - 0) = -0.1$$
$$Q\big((0,1),\rightarrow\big) = 0 + 0.1(-1 + 0.9 \times 0 - 0) = -0.1$$
$$Q\big((0,2),\rightarrow\big) = 0 + 0.1(-1 + 0.9 \times 0 - 0) = -0.1$$
$$Q\big((0,3),\uparrow\big) = 0 + 0.1(-1 + 0.9 \times 0 - 0) = -0.1$$
$$Q\big((1,3),\uparrow\big) = 0 + 0.1(-1 + 0.9 \times 0 - 0) = -0.1$$
$$Q\big((2,3),\uparrow\big) = 0 + 0.1(10 + 0.9 \times 0 - 0) = 1$$

مسیر ۲:

$$Q\big((0,0),\uparrow\big) = 0 + 0.1(-1 + 0.9 \times 0 - 0) = -0.1$$
$$Q\big((1,0),\uparrow\big) = 0 + 0.1(-1 + 0.9 \times 0 - 0) = -0.1$$
$$Q\big((2,0),\uparrow\big) = 0 + 0.1(-1 + 0.9 \times 0 - 0) = -0.1$$
$$Q\big((3,0),\rightarrow\big) = 0 + 0.1(-1 + 0.9 \times 0 - 0) = -0.1$$
$$Q\big((3,1),\rightarrow\big) = 0 + 0.1(-1 + 0.9 \times 0 - 0) = -0.1$$
$$Q\big((3,2),\rightarrow\big) = 0 + 0.1(10 + 0.9 \times 0 - 0) = 1$$

ج) می دانیم که در استراتژی $\varepsilon - greedy$ در $Q - learning$، عامل در هر استیت به شکل زیر عمل می کند:

- به احتمال ε یک کنش $action$ تصادفی انجام می دهد ($exploration$ یا اکتشاف).
- به احتمال $1 - \varepsilon$ به شکل حریصانه عمل می کند و کنش بهینه (که تا الان پیدا شده است) را انجام می دهد ($exploitation$ یا استفاده از اطلاعات).

نقش اکتشاف و استفاده از اطلاعات به شکل زیر می باشد:

- اکتشاف: با اکتشاف، عامل یادگیرنده می تواند اطلاعات بیشتری درباره کنش های مختلف و پاداش احتمالی آن ها جمع آوری کند که منجر به شناخت بهتر محیط خواهد شد. اگر عامل هیچ گاه اکتشاف نکند و همواره به صورت حریصانه عمل کند، ممکن است برخی از حالات را (که از قضا پاداش بسیار زیادی دارند) هیچ گاه مشاهده نکند و مقادیر $Q - value$ ها را به درستی یاد نگیرد (برخی از $Q - value$ ها را اصلا یاد نگیرد!).
- استفاده از اطلاعات: نقش استفاده از اطلاعات، بیشینه کردن پاداش به شکل فوری و لحظه ای با انجام کنشی است که تاکنون به عنوان کنش بهینه شناخته می شود. اگر عامل یادگیرنده، هیچ گاه به شکل حریصانه عمل نکند و همواره به صورت تصادفی کنش را انتخاب کند، تعداد اپیزود های لازم برای همگرایی الگوریتم بسیار زیاد شده و $regeret$ نیز افزایش خواهد یافت (کارایی الگوریتم بسیار پایین می آید).

نحوه تنظیم ε باید به این صورت باشد که در ابتدا مقدار نسبتا بزرگی داشته باشد (چرا که در ابتدا شناخت بسیار کمی از محیط داریم و باید محیط را اکتشاف کنیم تا به تدریج محیط را شناسایی نماییم) و سپس باید آن را به تدریج کاهش دهیم (چرا که هر چه پیش تر می رویم، شناخت بیشتری از محیط کسب می کنیم و کنش بهینه را دقیق تر شناسایی می کنیم. بنابراین بهتر است بجای حرکت کردن به شکل تصادفی که می تواند منجر به افزایش تعداد مراحل اجرای الگوریتم و $regret$ بسیار زیادی شود، به شکل حریصانه عمل کنیم تا فرایند همگرایی سریع تر صورت بگیرد). بنابراین یک روش بهینه می تواند انتخاب ε اولیه نسبتا بزرگ و سپس کاهش آن به شکل نمایی باشد.

سوال ۴:

الف) دقت کنید که به وضوح با توجه به تعریف، می توان نوشت:

$$\pi^*(s) = \operatorname*{argmax}_a Q^*(s,a)$$
$$V^*(s) = \max_a Q^*(s,a) = Q^*\big(s,\pi^*(s)\big)$$

همچنین طبق متن سوال، داریم:

$$\pi(s) = \operatorname*{argmax}_a \tilde{Q}(s,a)$$
$$\big\|\tilde{Q} - Q^*\big\|_\infty \le \varepsilon \Longrightarrow \max_{s,a}\big|\tilde{Q}(s,a) - Q^*(s,a)\big| \le \varepsilon \qquad (*)$$
$$\Longrightarrow \big|\tilde{Q}\big(s,\pi(s)\big) - Q^*\big(s,\pi(s)\big)\big| \le \varepsilon \Longrightarrow -Q^*\big(s,\pi(s)\big) \le \varepsilon - \tilde{Q}\big(s,\pi(s)\big)$$

حال دقت کنید که با توجه به نامساوی بالا، داریم:

$$V^*(s) - Q^*\big(s,\pi(s)\big) = Q^*\big(s,\pi^*(s)\big) - Q^*\big(s,\pi(s)\big) \le Q^*\big(s,\pi^*(s)\big) + \varepsilon - \tilde{Q}\big(s,\pi(s)\big)$$

حال دقت کنید که با توجه به تعریف $\pi(s)$، می توان نوشت:

$$\tilde{Q}\big(s,\pi^*(s)\big) \le \tilde{Q}\big(s,\pi(s)\big) \Longrightarrow -\tilde{Q}\big(s,\pi(s)\big) \le -\tilde{Q}\big(s,\pi^*(s)\big)$$
$$\Longrightarrow Q^*\big(s,\pi^*(s)\big) - \tilde{Q}\big(s,\pi(s)\big) \le Q^*\big(s,\pi^*(s)\big) - \tilde{Q}\big(s,\pi^*(s)\big)$$
$$(*) \Longrightarrow \big|\tilde{Q}\big(s,\pi^*(s)\big) - Q^*\big(s,\pi^*(s)\big)\big| \le \varepsilon \Longrightarrow Q^*\big(s,\pi^*(s)\big) - \tilde{Q}\big(s,\pi^*(s)\big) \le \varepsilon$$
$$\Longrightarrow Q^*\big(s,\pi^*(s)\big) - \tilde{Q}\big(s,\pi(s)\big) \le Q^*\big(s,\pi^*(s)\big) - \tilde{Q}\big(s,\pi^*(s)\big) \le \varepsilon$$

در نهایت، با ترکیب نامساوی های سبز رنگ و آبی رنگ می توان نوشت:

$$V^*(s) - Q^*\big(s,\pi(s)\big) \le Q^*\big(s,\pi^*(s)\big) + \varepsilon - \tilde{Q}\big(s,\pi(s)\big) \le \varepsilon + \varepsilon$$
$$\Longrightarrow V^*(s) - Q^*\big(s,\pi(s)\big) \le 2\varepsilon$$

که این همان حکم مسئله می باشد.

ب) در ابتدا توجه کنید که داریم:

$$\forall s \in S, a \in A;\ \sum_{s'} T(s,a,s') = 1$$

حال دقت کنید که طبق تعریف $\pi(s)$ و با استفاده از آنچه که در قسمت الف اثبات کردیم، برای هر استیت دلخواه مانند $s \in S$ می توان نوشت:

$$V^*(s) - V_\pi(s) = V^*(s) - \tilde{Q}(s,\pi(s)) = \left(V^*(s) - Q^*(s,\pi(s))\right) + \left(Q^*(s,\pi(s)) - \tilde{Q}(s,\pi(s))\right)$$
$$\le 2\varepsilon + \sum_{s'} T(s,\pi(s),s')\left(R(s,\pi(s),s') + \gamma V^*(s')\right) - \sum_{s'} T(s,\pi(s),s')\left(R(s,\pi(s),s') + \gamma V_\pi(s')\right)$$
$$= 2\varepsilon + \sum_{s'} \gamma T(s,\pi(s),s')\left(V^*(s') - V_\pi(s')\right)$$

حال اگر قرار دهیم $s_m = \underset{s}{\operatorname{argmax}}\, V^*(s) - V_\pi(s)$، خواهیم داشت:

$$\forall s \in S; \sum_{s'} \gamma T(s,\pi(s),s')\left(V^*(s') - V_\pi(s')\right) \le \sum_{s'} \gamma T(s,\pi(s),s')\left(V^*(s_m) - V_\pi(s_m)\right)$$
$$= \gamma\left(V^*(s_m) - V_\pi(s_m)\right)\sum_{s'} T(s,\pi(s),s') = \gamma\left(V^*(s_m) - V_\pi(s_m)\right)$$

حال با ترکیب دو نامساوی اخیر می توان نوشت: (در نامساوی اول قرار دهید $s = s_m$)

$$V^*(s_m) - V_\pi(s_m) \le 2\varepsilon + \sum_{s'} \gamma T(s,\pi(s),s')\left(V^*(s') - V_\pi(s')\right) \le 2\varepsilon + \gamma\left(V^*(s_m) - V_\pi(s_m)\right)$$
$$\Longrightarrow (1-\gamma)\left(V^*(s_m) - V_\pi(s_m)\right) \le 2\varepsilon \Longrightarrow V^*(s_m) - V_\pi(s_m) \le \frac{2\varepsilon}{1-\gamma}$$

حال دقت کنید که طبق تعریف s_m می توان نوشت:

$$\forall s \in S;\ V^*(s) - V_\pi(s) \le V^*(s_m) - V_\pi(s_m) \le \frac{2\varepsilon}{1-\gamma} \Longrightarrow V^*(s) - \frac{2\varepsilon}{1-\gamma} \le V_\pi(s)$$

که این همان حکم مسئله می باشد.

ج) به وضوح سیاست بهینه در استیت S_1، انجام کنش رفتن می باشد؛ چرا که مادامی که عامل یادگیرنده کنش ماندن را در این استیت انجام بدهد، پاداشی دریافت نخواهد کرد و تنها در صورتی که کنش رفتن را انجام بدهد، پاداش خواهد گرفت. بنابراین می توان نوشت: (در اینجا مجموعه کنش ها $A = \{s\ (stay), l\ (leave)\}$ می باشد):

$$V^*(S_2) = R(S_2, a \in A, S_2) + \gamma V^*(S_2) = 2\varepsilon + \gamma V^*(S_2) \Longrightarrow (1-\gamma)V^*(S_2) = 2\varepsilon$$

$$\Longrightarrow V^*(S_2) = \frac{2\varepsilon}{1-\gamma}$$

$$V^*(S_1) = \max\begin{Bmatrix} R(S_1, s, S_1) + \gamma V^*(S_1) \\ R(S_1, s, S_2) + \gamma V^*(S_2) \end{Bmatrix} = \max\begin{Bmatrix} 0 + \gamma V^*(S_1) \\ 2\varepsilon + \gamma V^*(S_2) \end{Bmatrix} = 2\varepsilon + \gamma V^*(S_2)$$

$$\Longrightarrow V^*(S_1) = 2\varepsilon + \gamma\left(\frac{2\varepsilon}{1-\gamma}\right) = \frac{2\varepsilon}{1-\gamma}$$

$$Q^*(S_1, s) = R(S_1, s, S_1) + \gamma V^*(S_1) = 0 + \gamma\left(\frac{2\varepsilon}{1-\gamma}\right) = \frac{2\gamma\varepsilon}{1-\gamma}$$

$$Q^*(S_1, l) = R(S_1, l, S_2) + \gamma V^*(S_2) = 2\varepsilon + \gamma\left(\frac{2\varepsilon}{1-\gamma}\right) = \frac{2\varepsilon}{1-\gamma}$$

د) تابع $\tilde{Q}$ را به شکل زیر می سازیم:

$$\tilde{Q}(S_1, s) = \tilde{Q}(S_1, l) = \frac{(1+\gamma)\varepsilon}{1-\gamma}$$

دقت کنید که در این حالت خواهیم داشت:

$$\left|Q^*(S_1, s) - \tilde{Q}(S_1, s)\right| = \frac{(1+\gamma)\varepsilon}{1-\gamma} - \frac{2\gamma\varepsilon}{1-\gamma} = \frac{(1-\gamma)\varepsilon}{1-\gamma} = \varepsilon$$

$$\left|Q^*(S_1, l) - \tilde{Q}(S_1, l)\right| = \frac{2\varepsilon}{1-\gamma} - \frac{(1+\gamma)\varepsilon}{1-\gamma} = \frac{(1-\gamma)\varepsilon}{1-\gamma} = \varepsilon$$

$$\Longrightarrow \left\|\tilde{Q} - Q^*\right\|_\infty = \max\{\varepsilon, \varepsilon\} = \varepsilon \le \varepsilon$$

و در نتیجه خطای $\tilde{Q}$ برابر ε می باشد. حال دقت کنید که با این انتخاب برای تابع $\tilde{Q}$، چون $\tilde{Q}(S_1, s) = \tilde{Q}(S_1, l)$ می باشد، ممکن است کنش ماندن به عنوان $\underset{a}{\operatorname{argmax}}\, \tilde{Q}(S_1, a)$ انتخاب شود و در این صورت خواهیم داشت:

$$\pi(S_1) = \underset{a}{\operatorname{argmax}}\, \tilde{Q}(S_1, a) = s \Longrightarrow V_\pi(S_1) = R(S_1, s, S_1) + \gamma V_\pi(S_1) \Longrightarrow V_\pi(S_1) = 0$$

به عبارت دیگر، چون در سیاست π همواره کنش ماندن توسط عامل در استیت S_1 انتخاب می شود، همواره در این استیت باقی خواهد ماند و پاداشی نخواهد گرفت. در نهایت درستی حکم سوال را می توان به سادگی نمایش داد.

$$V_\pi(S_1) - V^*(S_1) = 0 - \frac{2\varepsilon}{1-\gamma} = -\frac{2\varepsilon}{1-\gamma}$$

سوال ۵:

الف) به وضوح در این حالت می توان نوشت:

$$\sum_{t=0}^{\infty} \gamma^t r_t = R(s_0, a_1, s_1) + \sum_{t=1}^{\infty} \gamma^t R(s_1, a \in A, s_1) = 0 + \sum_{t=1}^{\infty} \gamma^t \times 1 = \frac{\gamma}{1-\gamma}$$

دقت کنید که در عبارت بالا منظور از $R(s_1, a \in A, s_1)$، پاداشی است که اگر در استیت s_1 قرار داشته باشیم و کنش دلخواه $a \in A$ را انجام دهیم دریافت می کنیم (دقت کنید که اگر در این استیت قرار داشته باشیم، با انجام هر کنش دلخواه در همین استیت باقی خواهیم ماند). همچنین دقت کنید که داریم:

$$\sum_{t=1}^{\infty} \gamma^t = \gamma \sum_{t=1}^{\infty} \gamma^{t-1} = \gamma \sum_{t=0}^{\infty} \gamma^t = \gamma \times \frac{1}{1-\gamma} = \frac{\gamma}{1-\gamma}$$

ب) به وضوح در این حالت می توان نوشت:

$$\sum_{t=0}^{\infty} \gamma^t r_t = R(s_0, a_2, s_2) + \sum_{t=1}^{\infty} \gamma^t R(s_2, a \in A, s_2) = \frac{\gamma^2}{1-\gamma} + \sum_{t=1}^{\infty} \gamma^t \times 0 = \frac{\gamma^2}{1-\gamma}$$

دقت کنید که در عبارت بالا منظور از $R(s_2, a \in A, s_2)$، پاداشی است که اگر در استیت s_2 قرار داشته باشیم و کنش دلخواه $a \in A$ را انجام دهیم دریافت می کنیم (دقت کنید که اگر در این استیت قرار داشته باشیم، با انجام هر کنش دلخواه در همین استیت باقی خواهیم ماند).

به وضوح برای $\gamma < 1$، نابرابری $\frac{\gamma}{1-\gamma} > \frac{\gamma^2}{1-\gamma}$ برقرار می باشد. بنابراین عمل بهنیه در این وضعیت، انجام کنش a_1 و انتقال به استیت s_1 خواهد بود.

ج) می دانیم که در $value\ iteration$، مقادیر به شکل زیر بروزرسانی می شوند:

$$\forall s \in S; V_0(s) = 0$$

$$V_{k+1}(s) = \max_a \sum_{s'} T(s,a,s')\big(R(s,a,s') + \gamma V_k(s')\big)$$

حال با استفاده از استقرا، نشان می دهیم که برای هر $n \geq 1$ تساوی های زیر برقرار می باشند:

$$V_n(s_0) = \max \begin{Bmatrix} \sum_{t=1}^{n-1} \gamma^t \\ \frac{\gamma^2}{1-\gamma} \end{Bmatrix}$$

$$V_n(s_1) = \sum_{t=0}^{n-1} \gamma^t$$

$$V_n(s_2) = 0$$

پایه: به ازای $n = 1$ حکم برقرار می باشد، چرا که داریم: (دقت کنید که $\sum_{t=1}^{0} \gamma^t = 0$ و $\sum_{t=0}^{0} \gamma^t = 1$ است)

$$V_1(s_0) = \max \begin{Bmatrix} R(s_0,a_1,s_1) + \gamma V_0(s_1) \\ R(s_0,a_2,s_2) + \gamma V_0(s_2) \end{Bmatrix} = \max \begin{Bmatrix} 0 + \gamma \times 0 \\ \frac{\gamma^2}{1-\gamma} + \gamma \times 0 \end{Bmatrix} = \max \begin{Bmatrix} 0 \\ \frac{\gamma^2}{1-\gamma} \end{Bmatrix}$$

$$V_1(s_1) = R(s_1, a \in A, s_1) + \gamma V_0(s_1) = 1 + \gamma \times 0 = 1$$

$$V_1(s_2) = R(s_2, a \in A, s_2) + \gamma V_0(s_1) = 0 + \gamma \times 0 = 0$$

گام: فرض کنید حکم به ازای $n - 1$ درست باشد. درستی حکم را برای n نشان می دهیم.

دقت کنید که با توجه به گام استقرا، داریم:

$$V_{n-1}(s_1) = \sum_{t=0}^{(n-1)-1} \gamma^t = \sum_{t=0}^{n-2} \gamma^t$$

$$V_{n-1}(s_2) = 0$$

بنابراین، در مرحله n ام اجرای $value\ iteration$ خواهیم داشت:

$$V_n(s_0) = \max\begin{Bmatrix} R(s_0, a_1, s_1) + \gamma V_{n-1}(s_1) \\ R(s_0, a_2, s_2) + \gamma V_{n-1}(s_2) \end{Bmatrix} = \max\begin{Bmatrix} 0 + \gamma \times \sum_{t=0}^{n-2} \gamma^t \\ \frac{\gamma^2}{1-\gamma} + \gamma \times 0 \end{Bmatrix} = \max\begin{Bmatrix} \sum_{t=1}^{n-1} \gamma^t \\ \frac{\gamma^2}{1-\gamma} \end{Bmatrix}$$

$$V_n(s_1) = R(s_1, a \in A, s_1) + \gamma V_{n-1}(s_1) = 1 + \gamma \times \sum_{t=0}^{n-2} \gamma^t = \gamma^0 + \sum_{t=1}^{n-1} \gamma^t = \sum_{t=0}^{n-1} \gamma^t$$

$$V_n(s_2) = R(s_2, a \in A, s_2) + \gamma V_{n-1}(s_1) = 0 + \gamma \times 0 = 0$$

و در نتیجه حکم برای n نیز برقرار می باشد و اثبات کامل است.

حال دقت کنید که $value\ iteration$ تا زمانی ادامه پیدا می کند که تساوی $\max\begin{Bmatrix} \sum_{t=1}^{n-1} \gamma^t \\ \frac{\gamma^2}{1-\gamma} \end{Bmatrix} = \sum_{t=1}^{n-1} \gamma^t$ یا به طور معادل $\sum_{t=1}^{n-1} \gamma^t \geq \frac{\gamma^2}{1-\gamma}$ برقرار شود؛ چرا که در صورت برقراری این تساوی کنش بهینه (که کنش a_1 می باشد) یافت می شود. بنابراین می توان نوشت:

$$\sum_{t=1}^{n^*-1} \gamma^t \geq \frac{\gamma^2}{1-\gamma} \Rightarrow \gamma\left(\frac{1-\gamma^{n^*}}{1-\gamma}\right) \geq \frac{\gamma^2}{1-\gamma} \Rightarrow 1-\gamma^{n^*} \geq \gamma \Rightarrow \gamma^{n^*} \leq 1-\gamma$$

$$\Rightarrow n^* \log(\gamma) \leq \log(1-\gamma) \xRightarrow{0<\gamma<1 \Rightarrow \log(\gamma)<0} n^* \geq \frac{\log(1-\gamma)}{\log(\gamma)}$$

که این همان حکم مسئله می باشد.