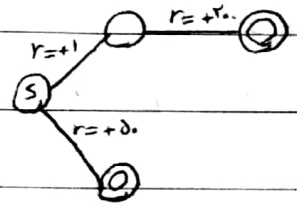
حمیدرضا امیرزاده
۴۰۱۲۰۶۹۹

موضوع: تمارین سری ششم درس یادگیری ژرف تاریخ:

(۱) آ) غلط است. به عنوان مثال نقض می‌توان MDP زیر را در نظر گرفت. اگر $\gamma=0.1$ باشد آنگاه $Q^*(s,down)=50$ و $Q^*(s,up)=1+0.1\times 200=21$ می‌شود. بنابراین اکشن بهینه حرکت down است. درحالی که اگر $\gamma=0.9$ باشد آنگاه $Q^*(s,down)=50$ و $Q^*(s,up)=1+0.9\times 200=181$ می‌شود. بنابراین اکشن بهینه حرکت up می‌شود. مشاهده می‌شود که سیاست بهینه در حالت s با تغییر ضریب تخفیف، تغییر می‌کند.



ب) مسئله: طراحی یک سیستم پیشنهاد دهنده برای یک وبسایت فروش کتاب. سیستم باید با استفاده از تاریخچه خرید و امتیازهای کاربران به آنها کتاب پیشنهاد کند.

مدل ۱: عامل را سیستم پیشنهاد دهنده در نظر می‌گیریم و محیط را مشتریان. در این مدل حالات محیط برابر تاریخچه خرید و امتیازهای مشتریان است. اکشن‌هایی که عامل می‌تواند انجام دهد، کتاب‌هایی است که می‌تواند پیشنهاد دهد. پاداش نیز نزدیکی است که مشتری (محیط) به سیستم (عامل) می‌دهد.

مدل ۲: عامل را مشتری در نظر می‌گیریم و محیط را سیستم پیشنهاد دهنده. در این مدل حالات محیط برابر لیست کتاب‌های موجود است. اکشن‌هایی که عامل می‌تواند انجام دهد کتاب‌هایی است که حاضرند. پاداش نیز میزان رضایت مشتری از خرید است.

اگر بخواهیم سیستم پیشنهاد دهنده را براساس بازخوردهای مشتری بهینه کنیم، مدل ۱ بهتر است. اما اگر بخواهیم رفتار مشتری و علایق وی را دریابیم، مدل ۲ بهتر است.

ج) الگوریتم Value Iteration با استفاده از معادلات بلمن، معیاری کوتاه‌مدتی بهینه را به دست می‌آورد. پس با استفاده از این معیاری کوتاه‌مدتی بهینه، سیاست بهینه را استخراج می‌کند. این روش اگر تعداد حالات زیاد باشد و یا تابع پاداش به‌خوبی تعریف نشده باشد به کندی همگرا می‌شود. الگوریتم Policy Iteration ابتدا با استفاده از یک سیاست دلخواه π معیاری کوتاه‌مدتی بهینه را می‌یابد سپس با استفاده از این معیاری، سیاست خود را به‌روزرسانی می‌کند. در این روش اگر آپدیت سیاست منجر به افزایش کوتاه‌مدتی نشود ممکن است به کندی همگرا شود. به‌طور کلی این دو الگوریتم به دلیل نیاز به ذخیره معیاری کوتاه‌مدتی و سیاست‌ها نیاز به حافظه زیادی هستند. همچنین به‌طور کلی الگوریتم Policy Iteration نسبت به Value Iteration از لحاظ محاسباتی بهینه‌تر است و معمولاً زودتر همگرا می‌شود.

موضوع : ..... تاریخ : .....

۴) $Q(1)=0$ $Q(2)=0$ $Q(3)=0$ $Q(4)=0$ $Q(5)=0$

$Q(1)=\frac{1}{-2}$ $Q(2)=\frac{1}{3}$ $Q(3)=\frac{1}{-1}$ $Q(4)=\frac{1}{5}$

$Q(2)=\frac{1}{2,5}$ $Q(3)=\frac{1}{-0,5}$

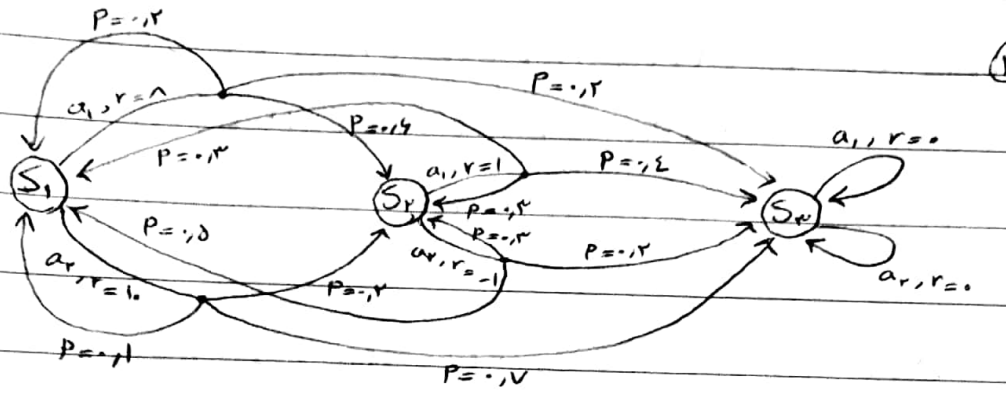
	ε random	1-ε greedy	
$(A_1=1, R_1=-2)$	✓	✓	
$(A_2=2, R_2=3)$	✓	✓	
$(A_3=3, R_3=-1)$	✓	X	→ در اینجا بهترین حالت اکشن ۲ بود
$(A_4=2, R_4=2)$	✓	✓	
$(A_5=3, R_5=0)$	✓	X	→ در اینجا بهترین حالت اکشن ۲ بود
$(A_6=4, R_6=5)$	✓	X	→ در اینجا بهترین حالت اکشن ۲ بود

در مراحل ۳، ۵ و ۶ حتماً ε رخ داده و اکشن تصادفی انتخاب شده. در باقی مراحل هر دوی ε و 1-ε یعنی انتخاب اکشن تصادفی و حریصانه ممکن بوده انتخاب شده باشند.

تاریخ :

موضوع :

(۲۰)



(۲۰) $\gamma = 1$

$V_0(S_1) = 10$ $\quad V_0(S_2) = 1$ $\quad V_0(S_3) = 0$

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s')\left[R(s,a,s') + \gamma V_k(s')\right]$$

$$V_1(S_1) = \max\{ \underbrace{0.2(8 + V_0(S_1)) + 0.4(8 + V_0(S_2)) + 0.2(8 + V_0(S_3))}_{10.4},$$
$$\underbrace{0.1(10 + V_0(S_1)) + 0.2(10 + V_0(S_2)) + 0.7(10 + V_0(S_3))}_{11.2} \}$$

$\Rightarrow V_1(S_1) = 11.2 \quad Q_1(S_1, a_1) = 10.4 \quad Q_1(S_1, a_2) = 11.2 \quad \pi_1(S_1) = a_2$

$$V_0(S_2) = \max\{ \underbrace{0.4(1 + V_0(S_1)) + 0.4(1 + V_0(S_2)) + 0.2(1 + V_0(S_3))}_{4.4},$$
$$\underbrace{0.8(-1 + V_0(S_1)) + 0.2(-1 + V_0(S_2)) + 0.2(-1 + V_0(S_3))}_{4.2} \}$$

$\Rightarrow V_1(S_2) = 4.4 \quad Q_1(S_2, a_1) = 4.4 \quad Q_1(S_2, a_2) = 4.2 \quad \pi_1(S_2) = a_1 \text{ or } a_2$

$$V_1(S_3) = \max\{ 1(0 + V_0(S_3)),\ 1(0 + V_0(S_3)) \}$$

$\Rightarrow V_1(S_3) = 0 \quad Q_1(S_3, a_1) = 0 \quad Q_1(S_3, a_2) = 0 \quad \pi_1(S_3) = a_1 \text{ or } a_2$

موضوع : ..... تاریخ : .....

$V_2(s_1) = \max \{ 0.2(8 + V_1(s_1)) + 0.6(8 + V_1(s_2)) + 0.2(8 + V_1(s_3)) ,$ (۱۲٫۸۲)

$0.1(10 + V_1(s_1)) + 0.2(10 + V_1(s_2)) + 0.7(10 + V_1(s_3)) \}$ (۱۱٫۹۸)

الف) $V_2(s_1) = 12.82 \quad Q_2(s_1, a_1) = 12.82 \quad Q_2(s_1, a_2) = 11.98 \quad \pi_2(s_1) = a_1$

$V_2(s_2) = \max \{ 0.3(1 + V_1(s_1)) + 0.3(1 + V_1(s_2)) + 0.4(1 + V_1(s_3)) ,$ (۵٫۴۵)

$0.5(-1 + V_1(s_1)) + 0.3(-1 + V_1(s_2)) + 0.2(-1 + V_1(s_3)) \}$ (۵٫۱۹)

ب) $V_2(s_2) = 5.89 \quad Q_2(s_2, a_1) = 5.45 \quad Q_2(s_2, a_2) = 5.89 \quad \pi_2(s_2) = a_2$

$V_2(s_3) = \max \{ 1(0 + V_1(s_3)) , 1(0 + V_1(s_3)) \}$

ج) $V_2(s_3) = 0 \quad Q_2(s_3, a_1) = 0 \quad Q_2(s_3, a_2) = 0 \quad \pi_2(s_3) = a_1 \text{ or } a_2$

ب) برای اینکه ثابت کنیم که سیاست بهینه در s_1 برای a_1 است و دیگر عوض نمی‌شود، باید داشته باشیم:

$0.2(8 + V_k(s_1)) + 0.6(8 + V_k(s_2)) + 0.2(8 + V_k(s_3)) \geq$

$0.1(10 + V_k(s_1)) + 0.2(10 + V_k(s_2)) + 0.7(10 + V_k(s_3))$

$\Rightarrow 0.1 V_k(s_1) + 0.4 V_k(s_2) \geq 2$

از آنجایی که $V_k(s_1) \geq 11.2$ و $V_k(s_2) \geq 4.3$ بنابراین رابطه بالا به ازای $k \geq 2$ همواره برقرار است.

برای اینکه ثابت کنیم که سیاست بهینه در s_2 برای a_2 است و دیگر عوض نمی‌شود، باید داشته باشیم:

$0.5(-1 + V_k(s_1)) + 0.2(-1 + V_k(s_2)) + 0.3(-1 + V_k(s_3)) \geq$

$0.3(1 + V_k(s_1)) + 0.3(1 + V_k(s_2)) + 0.4(1 + V_k(s_3))$

$\Rightarrow V_k(s_1) \geq 10$

از آنجایی که $V_k(s_1) \geq 11.2$ بنابراین رابطه بالا به ازای $k \geq 2$ همواره برقرار است.

از آنجایی که مقدار سودهی در s_3 همواره برابر صفر است بنابراین سیاست بهینه در آن همیشه تغییری نمی‌کند.

موضوع : ..... تاریخ : .....

۴) معماری I2A از ترکیب دو روش Model free و Model based بهره می برد که روش مبتنی بر مدل آن بر پایه مدل های محیطی است که مطابق Imagination و دور نگری عامل است. هدف از این قسمت این است که عامل قبل از اینکه در یک وضعیت خاص، یک اکشن خاص را انجام دهد، بتواند با استفاده از این ماژول Imagination یک تصوری از عواقب و حالاتی که پس از انجام یک اکشن مشخص به آن می رسد دست پیدا کند.

ساختار این ماژول به این صورت است که حالت فعلی یا $\hat{O}_t$ را به یک Policy net می دهیم و یک اکشن مناسب $\hat{a}_t$ می گیریم. حال برای اینکه بتوانیم به یک تصوری از عواقب انجام این اکشن برسیم، $\hat{O}_t$ و $\hat{a}_t$ را به یک Environment Model که اساساً یک شبکه عصبی است داده، و $\hat{O}_{t+1}$ و $\hat{r}_{t+1}$ را به دست می آوریم. به همین که این پروسه را انجام می دهد Imag. core می گوییم.

می توان با کنار هم گذاردن تعدادی از این Imag. core ها یک پشت سر هم، time step های بیشتری از آینده را تصور کنیم. در مرحله بعد خروجی هر یک از این Imag. core ها را به یک Encoder که می تواند بلاک LSTM باشد می دهیم، خروجی این encoder های پشت سر هم، یک Rollout encoding می سازد که نشان دهنده عواقب انجام اکشن a_t در O_t است. به هر یک از این بلوک ها Rollout encoder می گویند. با کنار هم گذاردن تعدادی از این بلوک ها می توان به تصورات خوبی رسید و عملکرد را این گونه بهبود داد. در نهایت خروجی این Rollout encoder ها توسط راهی به نام Aggregator باهم concat می کنند و به همراه خروجی شبکه Model free به شبکه استخراج کننده سیاست داده می شوند.

* مزایا: دستیابی عامل به تخمینی از عواقب انجام اکشن هایش می تواند منجر به انجام exploration بهینه تر و در نتیجه همگرایی سریع تر شود. / استفاده از این ماژول می تواند به عامل در انتخاب و پیش بینی اکشن هایی که منجر به پاداش نهایی بیشتری می شوند کمک کند

* معایب: استفاده از این ماژول امکان دارای حریصانه محاسباتی بالایی است و آموزش مدل را کندتر می کند. / اگر این ماژول با استفاده از یک مدل محیط دارای پایاس آموزش دیده باشد، ممکن است منجر به عملکرد غیر بهینه شود.