گزارش تمرین سوم

یادگیری تقویتی

پاییز ۱۴۰۰

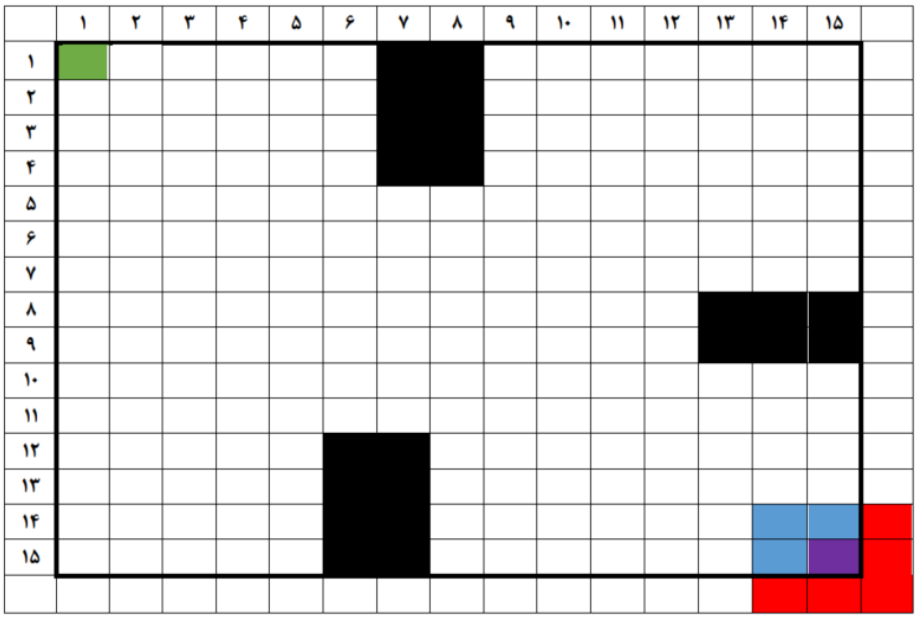
بخش دوم – پیاده سازی

علی ساعی زاده – ۸۱۰۱۹۶۴۷۷

فهرست

مقدمه

پیاده سازی مسئله مسیریابی بطور کامل طبق خواسته های موجود در صورت سوال اجرا شد و در فایل env.ipynp ضمیمه شده است.



شکل 1 نقشه مسئله

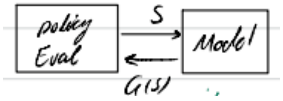
برای مقایسه پیاده سازی های این قسمت از متوسط تعداد حرکت برای رسیدن به مقصد استفاده شد.

همچنین برای ساده سازی نقسه به صورت یک ماترس دو بعدی در آمد که 0 نشان دهنده موانع، 1 به معنای خانه خالی و 2 به معنای مقصد مورد نظر است که در فایل maze.map قابل مشاهده است.

همچنین تصاویر سیر حرکت و همچنین فیلم آن به پروژه ضمیمه شده است.

شبیه کد مونت کارلو

از این روش برای حالتی استفاده می‌شود که اطلاعات محیط را به طور کامل نداشته باشیم و از طریق زندگی واقعی در محیط بخواهیم اطلاعات را بیرون بکشیم. دو شبه کد در کتاب ساتون ارائه شده است. در این روش چون احتمال و پاداش هارا نداریم باید به نوعی ارزش هر استیت را تخمین بزنیم با زندگی کردن واقعی در آن محیط. یعنی نیاز به توسعه مدلی به کمک تجربه داریم که بتوانیم policy evaluation را انجام دهیم.



$$V_\pi = E[G_t|S_t = s] = \frac{\sum_{i=0}^{L} G_i}{L}$$

First-visit MC prediction, for estimating $V \approx v_\pi$

Input: a policy π to be evaluated
Initialize:
  $V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$
  $Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):
  Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$
  $G \leftarrow 0$
  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
    $G \leftarrow \gamma G + R_{t+1}$
    Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:
      Append G to $Returns(S_t)$
      $V(S_t) \leftarrow$ average$(Returns(S_t))$

شکل 2 *Monte Carlo Policy Evaluation*

Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

Initialize:
  $\pi(s) \in \mathcal{A}(s)$ (arbitrarily), for all $s \in \mathcal{S}$
  $Q(s, a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$
  $Returns(s, a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Loop forever (for each episode):
  Choose $S_0 \in \mathcal{S}$, $A_0 \in \mathcal{A}(S_0)$ randomly such that all pairs have probability > 0
  Generate an episode from S_0, A_0, following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
  $G \leftarrow 0$
  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
    $G \leftarrow \gamma G + R_{t+1}$
    Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:
      Append G to $Returns(S_t, A_t)$
      $Q(S_t, A_t) \leftarrow$ average$(Returns(S_t, A_t))$
      $\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

شکل 3 *Monte Carlo learning (control)*

سیاست بهینه برای حالت پایه با استفاده از روش Policy Iteration

در این قسمت به کمک روش policy iteration عامل مسئله را برای حالت پایه به درستی حل کرد که سیاست بهینه بدست امده برای حالت پایه آن در پوشه output_policies قابل دسترسی است.

جدول 1 شرایط حالت پایه

احتمال انجام درست عمل انتخاب شده	0.8
هزینه برخورد با مانع	-1
هزینه انجام هر عمل	-0.01
پاداش رسیدم به خانه هدف	1000

این الگوریتم با استفاده از شبه کد زیر اجرا شد.

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

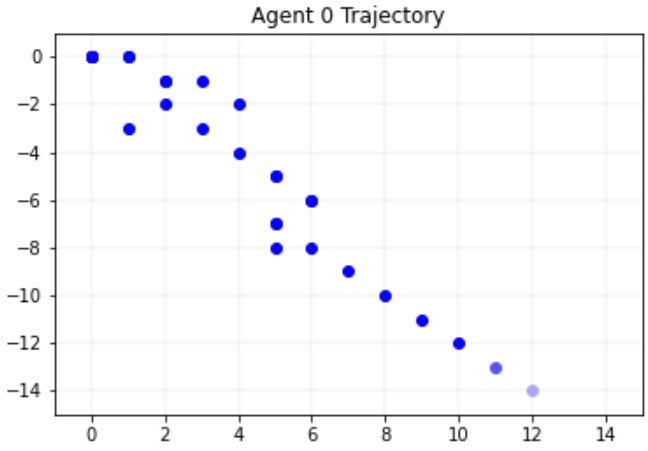
1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Loop:
   $\Delta \leftarrow 0$
   Loop for each $s \in \mathcal{S}$:
   $v \leftarrow V(s)$
   $V(s) \leftarrow \sum_{s',r} p(s', r|s, \pi(s))[r + \gamma V(s')]$
   $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   old-action $\leftarrow \pi(s)$
   $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r|s, a)[r + \gamma V(s')]$
   If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

شکل 4 شبه کد Policy Iteration

یک مسیر طی شده توسط عامل در شکل ۵ مشخص شده است. همچنین فیلم حرکت این عامل در پیوست تحت عنوان /GIFs/Agent1.gif آمده است.



شکل 5 مسیر عامل در حالت پایه

بطور متوسط در 1000 بار اجرای این مسئله به کمک سیاست بدست آمده بطور متوسط این عامل برای رسید به مقصد 18.975 حرکت انجام داده است.

سیاست بهینه برای حالت بدون اصطکاک با استفاده از روش Policy Iteration

برای این قسمت هم از روش policy iteration استفاده شد که در قست قبل توضیح داده شد اما شرایط مسئله در این سمت تغییر کرده است.

جدول 2 شرایط حالت بدون اصطکاک

احتمال انجام درست عمل انتخاب شده	0.8
هزینه برخورد با مانع	-0.01
هزینه انجام هر عمل	0
پاداش رسیدم به خانه هدف	1000

در این قسمت نیز یک مسیر حرکت طی شده توسط عامل به شکل زیر است. همچنین فیلم حرکت این عامل در پیوست تحت عنوان GIFs/Agent2.gif/ آمده است.

شکل 6 مسیر عامل در حالت بدون اصطکاک

بطور متوسط در 1000 بار اجرای این مسئله به کمک سیاست بدست آمده بطور متوسط این عامل برای رسید به مقصد 19.575 حرکت انجام داده است.

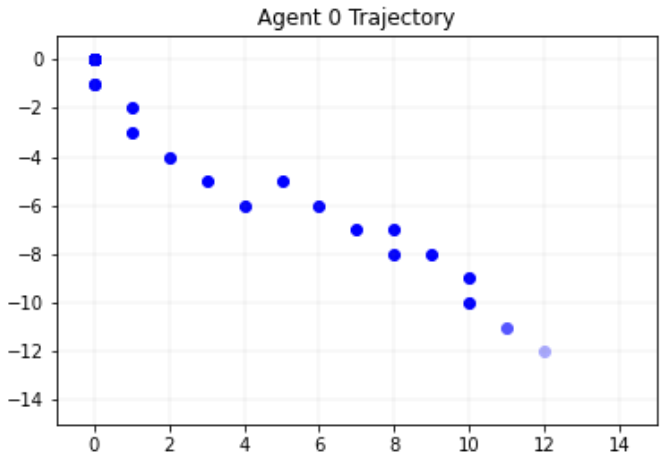
سیاست بهینه برای حالت اصطکاک زیاد با استفاده از روش Policy Iteration

در این قسمت نیز مانند قسمت های قبل از روش policy iteration استفاده شد.

جدول 3 شرایط اصطکاک زیاد

احتمال انجام درست عمل انتخاب شده	0.8
هزینه برخورد با مانع	-10
هزینه انجام هر عمل	-1
پاداش رسیدم به خانه هدف	100

در این قسمت نیز یک مسیر حرکت طی شده توسط عامل به شکل زیر است. همچنین فیلم حرکت این عامل در پیوست تحت عنوان /GIFs/Agent3.gif آمده است.



شکل 7 مسیر هرکت عامل در حالت اصطکاک زیاد

بطور متوسط در 1000 بار اجرای این مسئله به کمک سیاست بدست آمده بطور متوسط این عامل برای رسید به مقصد 18.256 حرکت انجام داده است.

با مقایسه سه حالت پیاده شده و مقایسه آن ها به کمک معیار تعداد حرکت انجام شده تا رسیدن به مقصد، نتیجه گیری می شود حالت با اصطکاک زیاد بهترین حالت برای این مسئله است.

جدول 4 مقایسه عملکرد عامل در حالت های مختلف

حالت	*پایه*	*بدون اصطکاک*	*با اصطکاک زیاد*
متوسط تعداد عمل تا رسیدن به مقصد	18.975	19.575	18.256

تغییر مقادیر Discount Factor

برای حالت کلی 11 مقدار مختلف discount factor بررسی شد که نتایج آن بصورت زیر است. همانظور که مشخص است بهترین مقدار discount factor برابر با 0.8 است.

برای حالاتی که عددی وارد نشده مسئله حل نشده است و عامل به مقصد نرسیده است.

همانطور که در کلاس درس گفته شد این پارامتر نشان میدهد که ارزش اعمال آینده ما در تصمیم گیری الان ما چه مقدار تاثیرگذار است. در مسئله MAZE که نوعی جهت یابی است بسیار آینده نگری نقش مهمی بازی می‌کند. همانطور که در مسیریابی در زندگی واقعی خود، بدون در نظر گرفتن مسیر های پیش رو (آینده) و مقصد نهایی امکان مسیریابی درست برای ما وجود ندارد در این شبیه سازی نیز آینده نگری نقش پر رنگی دارد. پس discount factor های نزدیک 1 نتیجه خوبی را به ما عرضه می‌کنند.

جدول 5 مقایسه عملکرد عامل با Discount Factor های متفاوت

Discount factor	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Avg Actions	-	-	-	-	-	-	48.34	18.13	17.97	18.61	18.04

پیاده سازی به روش Value Iteration

بنابر نتایج بدست آمده در قسمت های قبل بهترین discount factor برابر با 0.8 و بهترین حالت، حالت با اصطکاک زیاد بود. حالت در این قسمت به کمک روش value iteration مسئله ار حل می‌کنیم که جزییات ان در شبه کد زیر وجود است.

Value Iteration, for estimating $\pi \approx \pi_*$

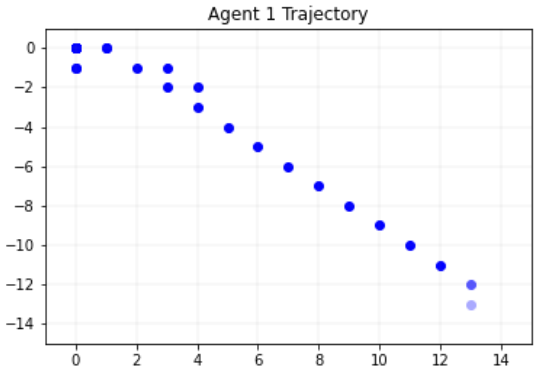
Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
  $\Delta \leftarrow 0$
  Loop for each $s \in \mathcal{S}$:
    $v \leftarrow V(s)$
    $V(s) \leftarrow \max_a \sum_{s',r} p(s', r|s, a)\left[r + \gamma V(s')\right]$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that
  $\pi(s) = \arg\max_a \sum_{s',r} p(s', r|s, a)\left[r + \gamma V(s')\right]$

شکل 8 شبه کد مربوط به Value Iteration

بطور متوسط در 100 بار اجرای این مسئله به کمک سیاست بدست آمده بطور متوسط این عامل برای رسید به مقصد 17.74حرکت انجام داده است. همچنین فیلم حرکت این عامل در پیوست تحت عنوان /GIFs/Agent4.gif آمده است.



شکل 9 مسیر حرکت عامل با روش Value Iteration

با مقایسه دو روش و مقایسه آن ها به کمک معیار تعداد حرکت انجام شده تا رسیدن به مقصد، نتیجه گیری می شود روش value iteration بهترین حالت برای این مسئله است.

جدول 6 مقایسه عملکرد Value Iteration و Policy Iteration

Method	Policy Iteration	Value Iteration
Avg. Actions to Win	18.7	17.74

تفاوت نتایج بخش های قبل

تفاوت حالت های پیاده شده به علت این اسست که در حالت های با اصطکاک کم عامل احتمال بیشتری برای رفتن به مسیر های غیر از مسیر بهینه قائل است به همین دلیل در مسیر خود ممکن است عملی را انتخاب کند که بهینه نیست (در مسیر مستقیم مقصد) به این دلیل که تنبیه شدیدی برای انجام هر حرکت درنظر گرفته نشده است پس احتمال بیشتری وجود دارد که متوسط انجام حرکات بیشتر شود.

همچنین تفاوت در تنبه برخورد به موانع با توجه به اینکه موانع بر سر راه مستقیم عامل نیست تاثیر چندانی ندارد اما اگر موانع بیشتر بود و عامل باید مسیر سختی را پیش میگرفت افزایش تنبیه موانع بسیار کمک کننده خواهد بود زیرا عامل به شدت از برخورد با مانع و گیر کرد در پشت یکی از موانع دوری می‌کرد که این در مسائلی با موانع پیچیده کمک کننده است.

تفاوت در رسیدن به مقصد برای ماندن در مقصد موثر است اما اگر معیار مقایسه تنها رسیدن به مقصد باشد تاثیر چندانی ندارد اما باید فاصله فاحشی با تنبه موانع و حرکت داشته باشد تا مسئله حل شود. همچنین در پیاده سازی ما همانطور که مشاهده شد اگر تنبیه برای حرکت درنظر گرفته نشود متوسط حرکات زیاد خواهد شد.

منابع

[1] Sutton, Richard S., and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.