دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده برق و کامپیوتر

گزارش تمرین شماره 4
درس یادگیری تعاملی
پاییز 1400

نام و نام خانوادگی	مرصاد اصالتی
شماره دانشجویی	810199089

فهرست

چکیده

در این تمرین به دنبال مطالعه و بررسی روش های Model-Free هستیم که بدون داشتن دینامیک محیط فرایند پیدا کردن سیاست بهینه را انجام می دهند. در انتها نیز به ترکیب روش های Model-Based و Model-Free می پردازیم

سوال 1 - Model-Based

هدف سوال

در این سوال به دنبال پیدا کردن سیاست بهینه با استفاده از الگوریتم Value Iteration هستیم. از سیاست بدست آمده از این قسمت به منظور بررسی همگرایی روش های Model-Free استفاده خواهیم کرد

پیاده سازی Value Iteration

Algorithm 2: Value Iteration for Q-values

Result: Find $Q^{\pi^*}(s,a)$ and $\pi^*(s),\ \forall s$

$Q(s,a) \leftarrow 0,\ \forall s, a \in S \times A$;
$\Delta \leftarrow \infty$;
while $\Delta \geq \Delta_\circ$ **do**
  $\Delta \leftarrow 0$;
  for *each* $s \in S$ **do**
    for *each* $a \in A$ **do**
      $temp \leftarrow Q(s,a)$;
      $Q(s,a) \leftarrow \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma \max_{a'} Q(s',a')]$;
      $\Delta \leftarrow \max_a(\Delta, |temp - Q(s,a)|)$;
    end
  end
end
$\pi^*(s) \leftarrow argmax_a Q^*(s,a), \forall s \in S$;
return $\pi^*(s), \forall s \in S$

رابطه ی بروز رسانی state-action value یا Q:

$$Q(s,a) = E[R(s,a,s') + \gamma \max_{a' \in A} Q(s',a')] = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma \max_{a' \in A} Q(s',a')]$$

مطابق شبه کد Value Iteration از مقدار دهی اولیه صفر برای state-action value ها شروع کرده و تا زمانی که اختلاف بین دو مرحله متفاوت کمتر از 0.01 شود عملیات بروزرسانی را ادامه می دهیم. در نهایت پس از همگرایی سیاست بهینه را به صورت greedy با توجه به مقادیر state-action value یا همان Q بدست می آوریم.

نتایج

```
-----------------------------
| 0.000 | 0.862 | 0.493 | 0.695 |
-----------------------------
| 0.001 | 0.001 | 0.001 | 0.001 |
-----------------------------
| 0.476 | 0.803 | 0.795 | 0.001 |
-----------------------------
| 0.893 | 0.822 | 0.248 | 0.000 |
-----------------------------
```

```
-----------------
| ↓ | ↓ | ↓ | ↓ |
-----------------
| → | → | → | ↓ |
-----------------
| ↑ | → | → | ↓ |
-----------------
| → | → | → | * |
-----------------
```

همانطور که مشاهده می شود روش Value Iteration که یک روش Model-Based می باشد توانسته برای تمامی خانه های نقشه سیاست بهینه را به درستی پیدا کند. برای بدست آوردن cumulative reward بهینه به تعداد 10000 بار از نقطه 0,0 شروع کرده و با استفاده از سیاست بهینه تصمیم گیری می کنیم :

Optimal Cumulative Reward: 34.75

سوال 2 - Model-Free

هدف سوال

در این سوال به دنبال یافتن سیاست بهینه با استفاده از روش های Model-Free هستیم. روش های پیاده سازی شده Off-Policy MC , Q-Learning , SARSA, N-Step Tree Backup می باشند.

پیاده سازی Off-Policy MC

Off-policy MC control, for estimating $\pi \approx \pi_*$

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
$Q(s,a) \in \mathbb{R}$ (arbitrarily)
$C(s,a) \leftarrow 0$
$\pi(s) \leftarrow \operatorname{argmax}_a Q(s,a)$ (with ties broken consistently)

Loop forever (for each episode):
$b \leftarrow$ any soft policy
Generate an episode using b: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
$G \leftarrow 0$
$W \leftarrow 1$
Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
$G \leftarrow \gamma G + R_{t+1}$
$C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)} [G - Q(S_t, A_t)]$
$\pi(S_t) \leftarrow \operatorname{argmax}_a Q(S_t, a)$ (with ties broken consistently)
If $A_t \neq \pi(S_t)$ then exit inner Loop (proceed to next episode)
$W \leftarrow W \frac{1}{b(A_t|S_t)}$

مشکل اصلی این روش sample efficiency می باشد. این مشکل به دو دلیل رخ می دهد:

1. بریده شدن اپیزود تولیدی به علت عدم پوشش سیاست رفتاری (e-greedy) توسط سیاست مورد ارزیابی (greedy)

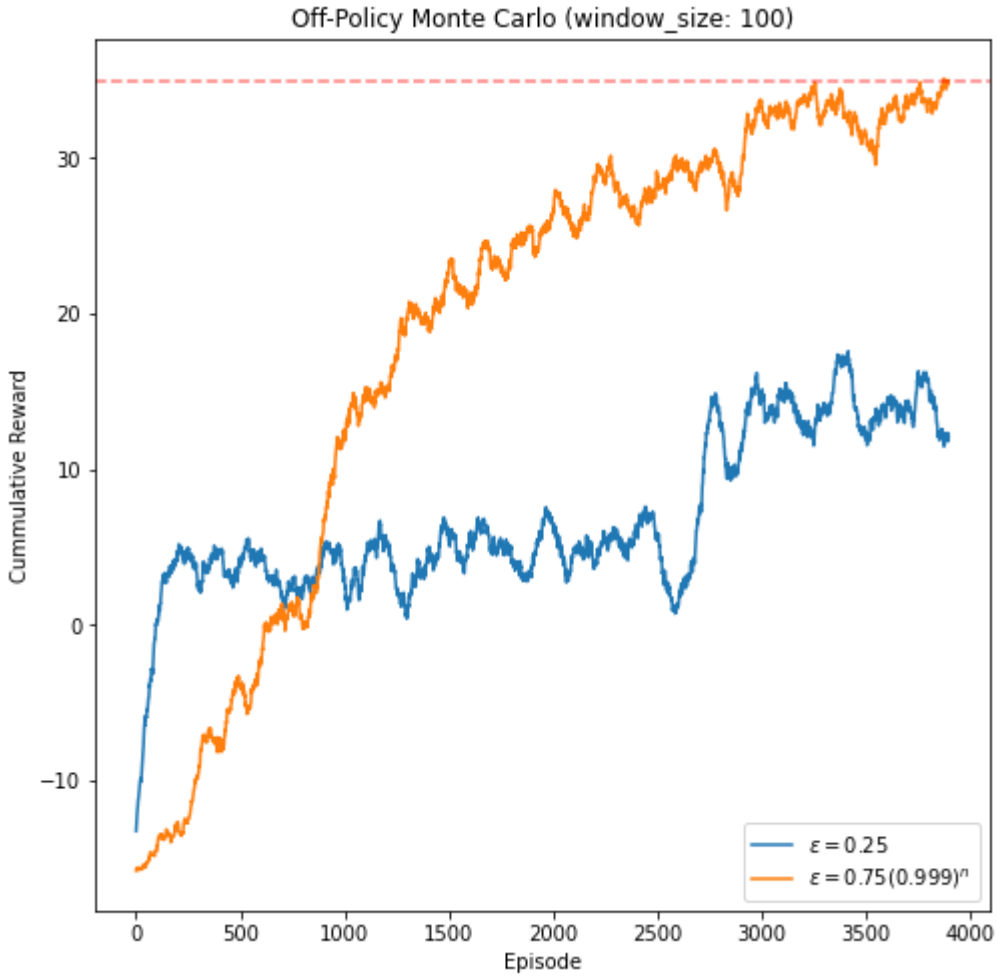
2. عدم توانایی ساخت اپیزود هایی با احتمال رخداد غیر صفر به صورت ذهنی

نتایج

الگوریتم فوق را برای دو حالت epsilon ثابت و epsilon کاهشی اجرا کرده و نتایج بدست آمده را مقایسه می کنیم (به دلیل احتمالاتی بودن محیط هر یک از روش ها را به تعداد 20 تکرار و به مدت 4000 اپیزود اجرا کرده و میانگین بدست آمده را پس از اعمال پنجره با ابعاد 100 بررسی میکنیم)

epsilon constant: 0.25

epsilon decay: 0.75*(1-1e-3)**eps



همانطور که مشاهده می شود با استفاده از epsilon کاهشی فرایند پیدا کردن سیاست بهینه به صورت کامل انجام می شود و agent به optimal cumulative reward دست پیدا میکند ولی با مقدار epsilon ثابت و برابر 0.25 مدل به سیاست بهینه همگرا نمی شود علت این مشکل نداشتن exploration کافی می باشد. از طرفی سرعت همگرایی در حالت epsilon ثابت بالا تر بوده به عبارتی دیگر در اپیزود های آغازی توانسته به نتایج بهتری دست پیدا کند. لازم به ذکر است برای حالت epsilon ثابت با اجرا های مختلف نتایج متفاوتی حاصل می شود !!! که بیانگر وابستگی شدید این الگوریتم به sample ها یا episode های تولید شده می باشد. (مقدار **epsilon** ثابت نسبت به صورت سوال تغییر کرده تا نتایج قابل تحلیل باشند)

پیاده سازی Q-Learning

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

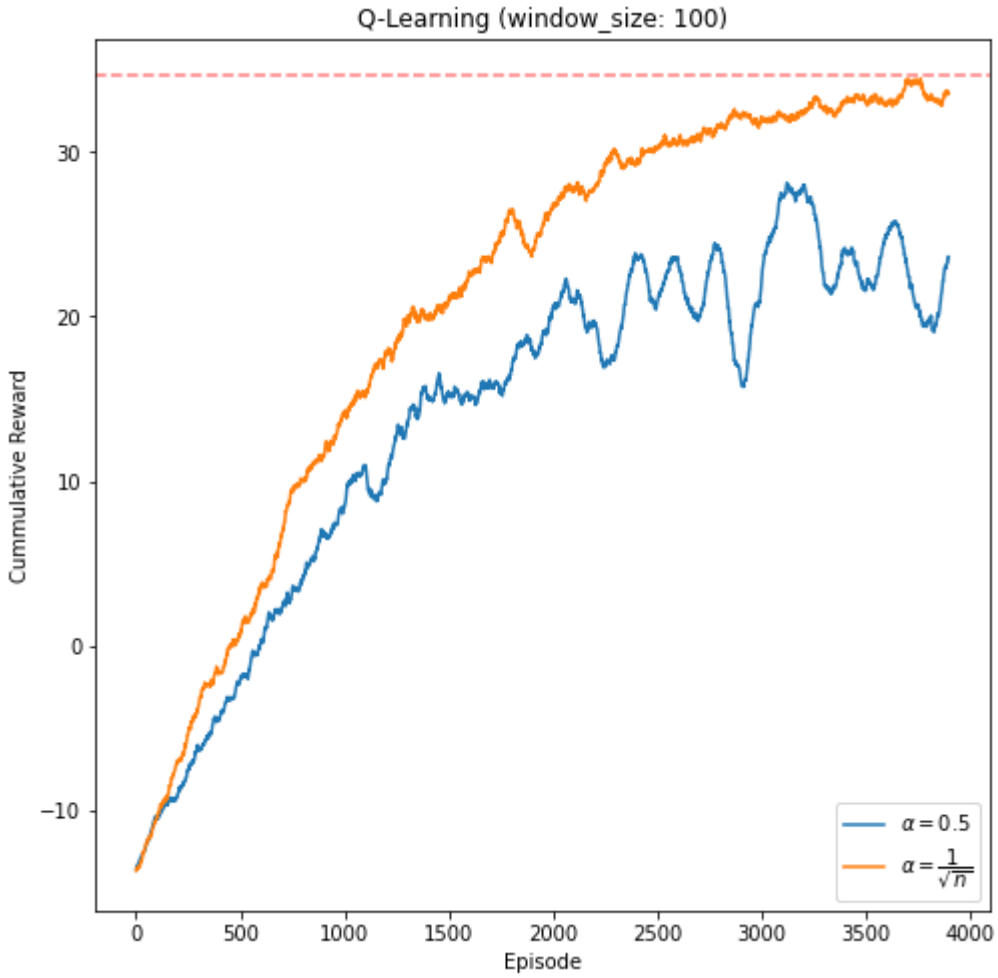
Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
    $S \leftarrow S'$
  until S is terminal

نتایج

الگوریتم فوق را به صورت epsilon کاهشی مشابه روش Off-Policy MC و برای دو حالت learning rate ثابت و learning rate کاهشی پیاده سازی و اجرا کرده و نتایج بدست آمده را مقایسه می کنیم (به دلیل احتمالاتی بودن محیط هر یک از روش ها را به تعداد 20 تکرار و به مدت 4000 اپیزود اجرا کرده و میانگین بدست آمده را پس از اعمال پنجره با ابعاد 100 بررسی میکنیم)

learning rate constant: 0.5

learning rate decay: $\frac{1}{\sqrt{eps}}$



همانطور که مشاهده می شود در حالت learning rate کاهشی agent با سرعت همگرایی تقریبا مناسبی موفق به دستیابی به optimal cumulative reward شده است اما در حالت استفاده از learning rate ثابت همگرایی به صورت کامل صورت نگرفته است. (مقدار **learning rate** ثابت نسبت به صورت سوال تغییر کرده تا نتایج قابل تحلیل باشند)

پیاده سازی SARSA

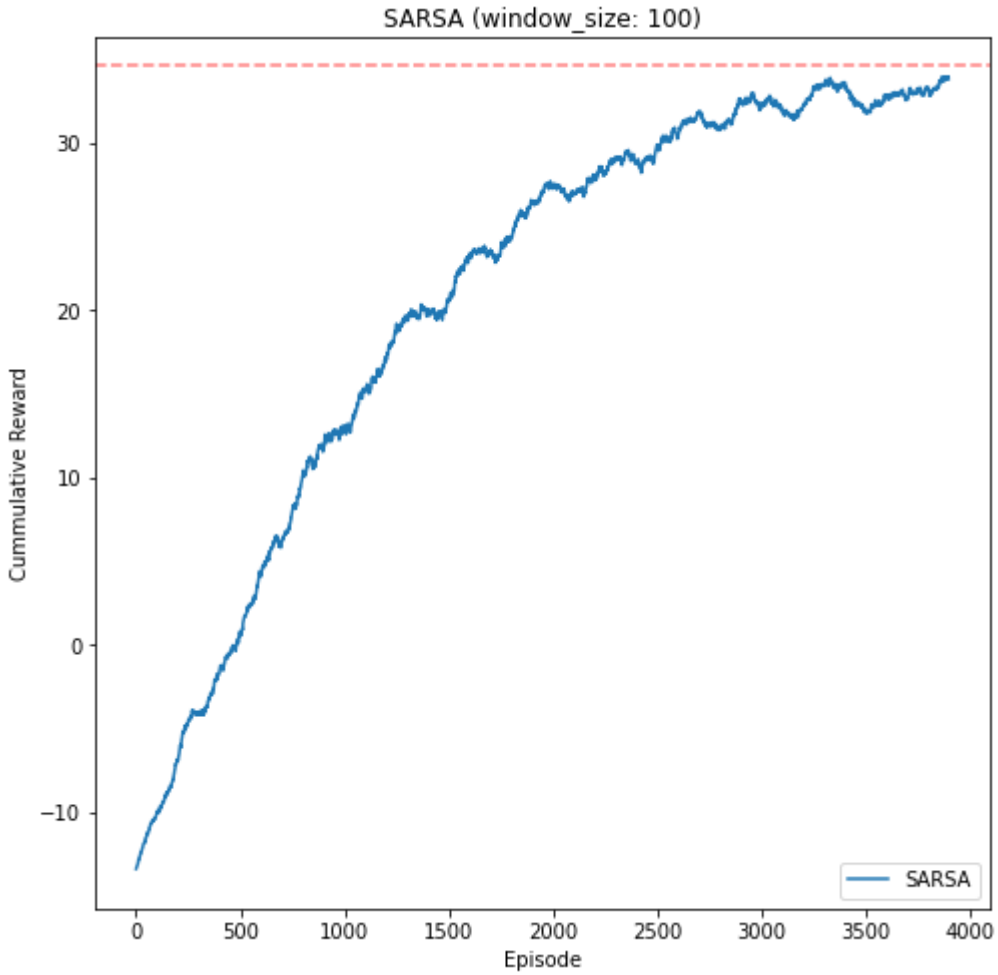
Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Choose A from S using policy derived from Q (e.g., ε-greedy)
  Loop for each step of episode:
    Take action A, observe R, S'
    Choose A' from S' using policy derived from Q (e.g., ε-greedy)
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma Q(S', A') - Q(S, A)\big]$
    $S \leftarrow S'$; $A \leftarrow A'$;
  until S is terminal

نتایج

الگوریتم فوق را به صورت epsilon کاهشی مشابه روش Off-Policy MC اجرا کرده و نتیجه بدست آمده را بررسی می کنیم.



همانطور که مشاهده می شود agent با سرعت همگرایی مناسبی موفق به دستیابی به optimal cumulative reward شده است.

پیاده سازی N-Step Tree Backup

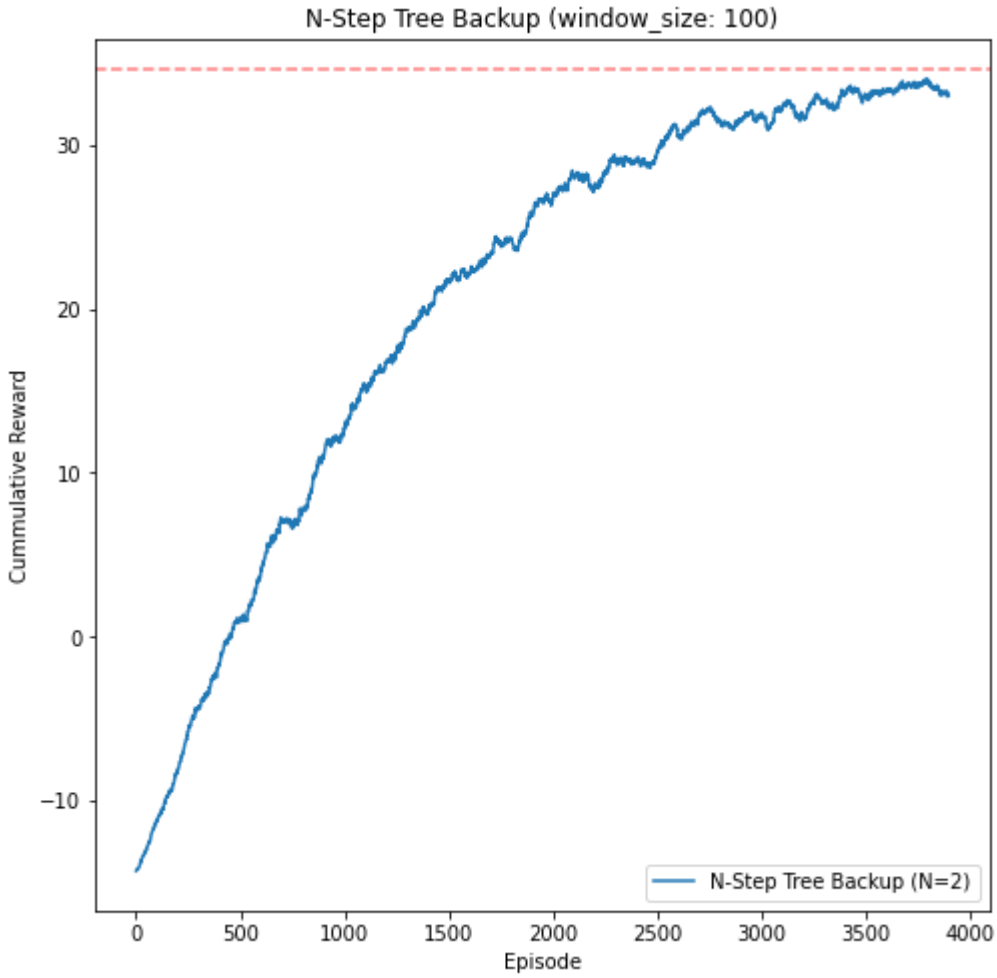
n-step Tree Backup for estimating $Q \approx q_*$ or q_π

Initialize $Q(s,a)$ arbitrarily, for all $s \in \mathcal{S}, a \in \mathcal{A}$
Initialize π to be greedy with respect to Q, or as a fixed given policy
Algorithm parameters: step size $\alpha \in (0,1]$, a positive integer n
All store and access operations can take their index mod $n+1$

Loop for each episode:
  Initialize and store $S_0 \neq$ terminal
  Choose an action A_0 arbitrarily as a function of S_0; Store A_0
  $T \leftarrow \infty$
  Loop for $t = 0, 1, 2, \ldots$:
  | If $t < T$:
  | Take action A_t; observe and store the next reward and state as R_{t+1}, S_{t+1}
  | If S_{t+1} is terminal:
  | $T \leftarrow t+1$
  | else:
  | Choose an action A_{t+1} arbitrarily as a function of S_{t+1}; Store A_{t+1}
  | $\tau \leftarrow t+1-n$ (τ is the time whose estimate is being updated)
  | If $\tau \geq 0$:
  | If $t+1 \geq T$:
  | $G \leftarrow R_T$
  | else
  | $G \leftarrow R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) Q(S_{t+1}, a)$
  | Loop for $k = \min(t, T-1)$ down through $\tau + 1$:
  | $G \leftarrow R_k + \gamma \sum_{a \neq A_k} \pi(a|S_k) Q(S_k, a) + \gamma \pi(A_k|S_k) G$
  | $Q(S_\tau, A_\tau) \leftarrow Q(S_\tau, A_\tau) + \alpha \left[G - Q(S_\tau, A_\tau) \right]$
  | If π is being learned, then ensure that $\pi(\cdot|S_\tau)$ is greedy wrt Q
  Until $\tau = T - 1$

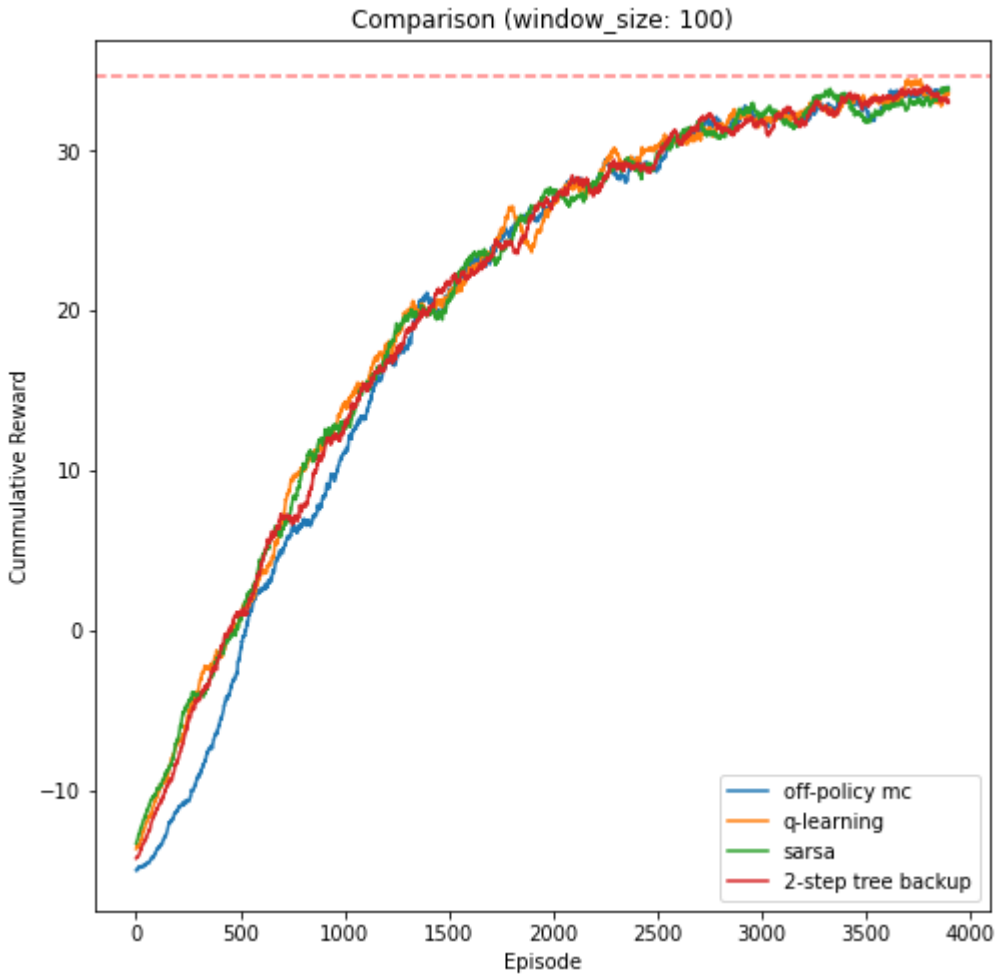
نتایج

الگوریتم فوق را به صورت epsilon کاهشی مشابه روش Off-Policy MC و N برابر با 2 اجرا کرده و نتیجه بدست آمده را بررسی می کنیم.



همانطورکه مشاهده می شود agent با سرعت همگرایی مناسبی موفق به دستیابی به optimal cumulative reward شده است.

مقایسه



همانطور که مشاهده می شود تمامی روش های مطرح شده از نظر سرعت همگرایی و مقدار همگرا شده یکسان می باشند.

نکات مهم و موارد تحویلی

لازم است که به نکات زیر در نوشتن گزارش توجه داشته باشید.

1. ساختار کلی گزارش که در این فایل به آن اشاره شده باید رعایت شود. در صورت تمایل می‌توانید از latex یا هر نرم افزار دلخواه دیگر برای نوشتن گزارش استفاده کنید، به شرط اینکه ساختار کلی گفته شده رعایت شود. لذا در صورت رعایت نکردن ساختار کلی گزارش بخشی از نمره تمرین کم خواهد شد.
2. برای تصاویر موجود در گزارش حتما زیر نویس و برای جداول استفاده شده در گزارش بالانویس (اجباری) قرار داده شود.
3. نتایج و تحلیل‌های شما در روند نمره دهی اهمیت بسیار بالایی دارد، لذا خواهشمندیم کلیه نتایج و تحلیل‌های خواسته شده به صورت کامل و دقیق در گزارش آورده شوند.
4. در صورت مشاهده شباهت بین گزارش شما و افراد مختلف نمره این سری تمرین برای شما در نظر گرفته نمی‌شود.

موارد تحویلی

1. برای هر سری از تمرینات، فقط یک فایل با فرمت PDF آماده کنید.
2. به همراه فایل گزارش، یک پوشه به نام Codes ایجاد کنید و کدها و فایل‌های پیاده‌سازی هر سوال را به صورت تفکیک شده در پوشه‌های جداگانه قرار دهید.
3. هیچ گونه جدول یا تصویر به صورت جداگانه خارج از گزارش ارسال نشود. مگر اینکه به صورت صریح در تمرین از شما خواسته شده باشد.
4. در انتها، لطفا برای هر تمرین گزارش و پوشه کدها را به صورت گفته شده، در یک فایل زیپ با فرمت زیر در سامانه یادگیری الکترونیک بارگذاری نمایید.

HW#_LastName_StudentNumber.zip

به طور مثال:

HW1_Mesbah_810111111.zip

منابع

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA: Bradford Books, 2018.