Artificial Intelligence

Question 1

Taxi-v2 with value-iteration and policy-iteration

Mohammad Reza Ghamkhar 95106494

Gamma

- **گاما = 0.99 و value_iteration**

برای تشخیص همگرایی از متغییر delta استفاده کردم؛ یعنی زمانی که همه مقادیر $V_{k+1}(S) - V_K(S) < delta$ باشد به همگرایی رسیده ایم. حال اگر از روش value iteration با gamma=0.99 و delta=0.01 استفاده کنیم، الگوریتم ما بعد 758 مرحله همگرا می شود(زیرا در نهایت یک جایی $gamma ** n * V(s')$ کمتر از delta می شود). همانطور که در شکل مشاهده می کنید:

```
Hi, Mreza says 'Hello'...   lets go!
==================
finding and optimal policy for 'Taxi-v2' using 'Value_Iteration'
gamma: 0.99
number of states: 500
number of actions: 6
delta: 0.01
Convergence happened after '758' iterations
policy:
[4, 4, 4, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 3, 3, 3, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3
==================
```

- **گاما = 1 و value_iteraion**

در این حالت در هر مرحله مقدار $Q(s,a)$ حداقل به اندازه $V(s')$ تغییر می کند در نتیجه هیچ وقت همگرا نمی شود اگرچه چون در هر مرحله مقدار $V(s)$ به یک نسبت زیاد می شود به یک حالت شبه همگرا می رسد. در این تابع من یک حد بالا برابر 1000 برای مقدار iteration ها برای همگرایی در نظر گرفتم به این معنی که اگر بعد از 1000 مرحله همگرا نشد همان مقدار $V(s)$ را به عنوان مقدار همگرا شده بر می گرداند. تصویر مربوط به این حالت:

```
Hi, Mreza says 'Hello'...   lets go!
==================
finding and optimal policy for 'Taxi-v2' using 'Value_Iteration'
gamma: 1
number of states: 500
number of actions: 6
delta: 0.01
Convergence happened after '1000' iterations
[9500.0, 9437.0, 9458.0, 9416.0, 9374.0, 9437.0, 9353.0, 9374.0, 9416.0, 9374.0, 9458.0,
policy:
[4, 4, 4, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 3, 0, 0, 0, 0, 0,
==================
```

- **گاما = 0.99 و policy_iteration**

هر iteration در این روش شامل policy_evaluation و policy_improvment است.

که مرحله policy_evaluation تقریبا مشابه value iteration با gamma=0.99 است و در نتیجه همگرا است (که البته خیلی زودتر از value iteration همگرا می شود چون فقط براساس policy یک action انتخاب می کند و برای همه action های موجود این کار را انجام نمی دهد درنتیجه در هر مرحله تفاوت نسبت به حالت قبل کمتر می شود)

در مورد policy_improvment هم چون policy_evaluation همگرا است براساس صفحه 69 اسلاید MDP استاد، همگرا می شود. تصویر متناظر با این ادعا که نشانگر همگرا شدن بعد از 17 iteration است:

```
finding and optimal policy for 'Taxi-v2' using 'Policy_Iteration'
gamma: 0.99
number of states: 500
number of actions: 6
delta: 0.01
Convergence happened after '17' iterations
policy:
[4, 4, 4, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 3, 3, 3,
```

- **گاما = 1 و policy_iteration**

در این حالت ،مرحله policy_evaluation به خاطر این که مانند value_iteration از $\max_a$ استفاده نمی کند و همواره از $\pi(s)$ استفاده می کند و gamma=1 است در نتیجه مقدار V_{K+1} همواره مقداری ثابت کمتر از V_k است و هیچ وقت همگرا نمی شود در نتیجه در این حالت مقدار policy_evaluation و policy_improvment هیچ وقت همگرا نمی شود. و جواب بهینه بدست نمی آید(برخلاف gamma=1 value iteration که بازهم بعد از تعداد خوبی مرحله جواب بهینه بدست می آمد)

مقایسه جواب ها با gamma=1 و gamma=0.99

در روش value_iteration هنگامی که gamma=1 است اگر تعداد مناسبی iteration گذشته باشد می توانیم فرض کنیم به همگرایی رسیده ایم و از روی $Q(s,a)$ ها policy بهینه را بدست آوریم که این policy **مشابه** policy بدست آمده از طریق value_iteration با gamma=0.99 است.

اما در مورد policy_iteration قضیه متفاوت است و جواب حاصل از gamma=0.99 جواب بهینه و درست است و با جواب حاصل از gamma=1 **متفاوت** است (که جوابی نادرست و غیربهینه است).

مقایسه زمانی policy_iteration vs value_iteration

در مورد این مسئله (taxi-v2) الگوریتم policy_iteration بسیار سریع تر (حدود 5 برابر) از الگوریتم value_iteration است. (policy_iteration حدود 4 ثانیه و value_iteration حدود 0.8 ثانیه طول می کشد(cpu intel Core i5 2.3

GHz)). با اجرا کردن کد Taxi.py می توانید نتایج را مشاهده کنید: