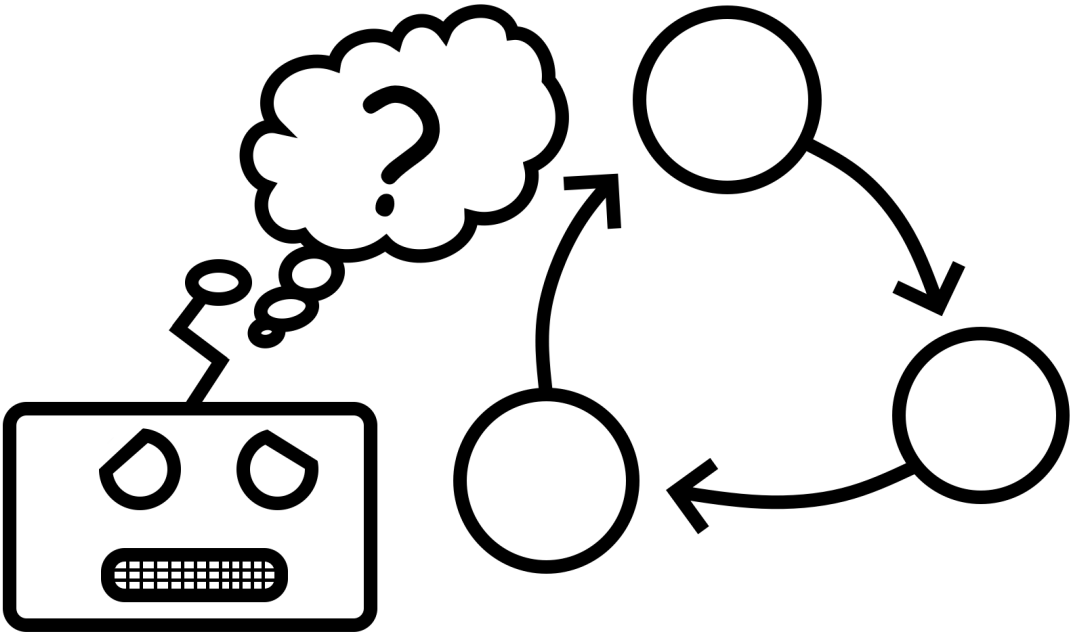
Model-free Prediction & Control



John Williams - 1790472

A. Monte-Carlo policy evaluation

I. *Bepaal de optimale policy π_* en gebruik deze voor de evaluatie. Start elke simulatie op de startpositie in het doolhof. Wat is de value function na een oneindig aantal iteraties als $\gamma = 1$? Visualiseer dit op een inzichtelijke manier.*

Links de optimale policy π_* met een rode pijl om de start positie aan te geven, en rechts de value matrix V geïnitialiseerd met willekeurige reële getallen (0.0 voor terminal states). De states worden aangeduid met een (x, y) coördinaat, waarbij (0, 0) links boven is.

$$\pi_* = \begin{bmatrix} \rightarrow & \rightarrow & \rightarrow & \phi \\ \rightarrow & \uparrow & \uparrow & \uparrow \\ \rightarrow & \uparrow & \leftarrow & \leftarrow \\ \phi & \uparrow & \uparrow & \leftarrow \end{bmatrix} \qquad V = \begin{bmatrix} 6.83 & 9.69 & -9.93 & 0.0 \\ -9.55 & -1.99 & -4.14 & 2.45 \\ -1.27 & 6.33 & 4.09 & 6.08 \\ 0.0 & -0.7 & -8.37 & 0.34 \end{bmatrix}$$

De episode die met deze policy altijd uitgevoerd zal worden volgt altijd deze states:

(2, 3) → (2, 2) → (1, 2) → (1, 1) → (1, 0) → (2, 0) → (3, 0)

Met behulp van de First-visit MC Prediction algoritme kunnen de returns van de states die door de episodes bezocht worden berekend, om daarna tot value omgezet te worden door het gemiddelde te nemen.

$$Returns(s) = \begin{Bmatrix} (2,0): & [40] \\ (1,0): & [39] \\ (1,1): & [38] \\ (1,2): & [37] \\ (2,2): & [36] \\ (2,3): & [35] \end{Bmatrix} \xrightarrow{next} \begin{Bmatrix} (2,0): & [40,40] \\ (1,0): & [39,39] \\ (1,1): & [38,38] \\ (1,2): & [37,37] \\ (2,2): & [36,36] \\ (2,3): & [35,35] \end{Bmatrix} \xrightarrow{next} \begin{Bmatrix} (2,0): & [40,40,40] \\ (1,0): & [39,39,39] \\ (1,1): & [38,38,38] \\ (1,2): & [37,37,37] \\ (2,2): & [36,36,36] \\ (2,3): & [35,35,35] \end{Bmatrix} \xrightarrow{etc}$$

Aangezien de episode met deze policy altijd hetzelfde is, zijn de returns van een state altijd hetzelfde. Als hiervan dan de gemiddelde genomen worden, komen de volgende values hieruit:

$$V = \begin{bmatrix} 6.83 & 39.0 & 40.0 & 0.0 \\ -9.55 & 38.0 & -4.14 & 2.45 \\ -1.27 & 37.0 & 36.0 & 6.08 \\ 0.0 & -0.7 & 35.0 & 0.34 \end{bmatrix}$$

II. *Wat is de value function na een oneindig aantal iteraties als $\gamma = 0.5$? Visualiseer weer op een inzichtelijke manier.*

Hetzelfde proces als hierboven wordt uitgevoerd, maar dan met een discount factor van 0.5.

$$Returns(s) = \begin{Bmatrix} (2,0): & [40] \\ (1,0): & [19] \\ (1,1): & [8.5] \\ (1,2): & [3.25] \\ (2,2): & [0.63] \\ (2,3): & [-0.69] \end{Bmatrix} \xrightarrow{next} \begin{Bmatrix} (2,0): & [40,40] \\ (1,0): & [19,19] \\ (1,1): & [8.5,8.5] \\ (1,2): & [3.25,3.25] \\ (2,2): & [0.63,0.63] \\ (2,3): & [-0.69,-0.69] \end{Bmatrix} \xrightarrow{etc}$$

Hiervan worden vervolgens weer de gemiddelden genomen voor de values:

$$V = \begin{bmatrix} 6.83 & 19.0 & 40.0 & 0.0 \\ -9.55 & 8.5 & -4.14 & 2.45 \\ -1.27 & 3.25 & 0.63 & 6.08 \\ 0.0 & -0.7 & -0.69 & 0.34 \end{bmatrix}$$

III. *Als het goed is, kom je met deze policy tot een onvolledige value function. Waarom is dat het geval? Hoe kan je dat oplossen?*

Deze policy kan meerdere states nooit bereiken doordat het altijd dezelfde episode produceert, en zal dus van deze states nooit de value kunnen berekenen. Dit kan opgelosd worden door een policy in te stellen die meer willekeur biedt, bijvoorbeeld met een volledig willekeurige policy of een kans om een willekeurige actie te kiezen.

IV. *Stel, je maakt gebruik van any-visit Monte-Carlo prediction (in plaats van first-visit). Maakt dat een verschil voor het resultaat? Waarom wel/niet?*

In het geval van deze policy maakt het geen verschil. Any-visit zorgt er niet voor dat er plots andere states bezocht worden. En de episode die altijd uit deze policy komt komt nooit twee keer langs dezelfde state, dus is first-visit praktisch hetzelfde als any-visit.

B. Temporal difference learning

I. *In de pseudocode van temporal difference learning hierboven worden alle values een willekeurige waarde toegewezen, behalve de terminal states -- deze worden geïnitialiseerd op 0. Bij Monte-Carlo policy evaluation was deze uitzondering niet nodig. Waarom is er een andere initialisatie nodig?*

Bij Monte-Carlo policy evaluation wordt de value van een terminal state nooit meegenomen in de berekeningen, dus het maakt daar niet uit wat voor waarde het heeft. Echter bij TD wordt er per stap gekeken naar de value van de volgende state, en als dat een terminal state is maakt de value van de terminal state dus wel uit.

II. *Noem een voordeel en een nadeel van dit algoritme ten opzichte van het Monte-Carlo algoritme.*

Een voordeel van TD is dat het om kan gaan met oneindige episodes. TD hoeft niet te wachten tot er een terminal state bereikt is voordat het de policy kan evalueren, terwijl als de MC algoritme in een oneindige episode terecht komt kan de policy nooit geëvalueerd worden. Een nadeel van TD is dat het niet weet wat de returns zijn van de states, deze moeten “gegokt” worden wat de mogelijkheid tot onnauwkeurigheid, en dus de mogelijkheid tot een biased antwoord, vergroot.

C. On-policy first-visit Monte-Carlo Control

I. *Leg in eigen woorden uit wat een Q-function is.*

Een Q-function is een functie die voor een gegeven state en een gegeven actie een value teruggeeft, de verwachte reward van de state waar de agent terecht komt door het uitvoeren van de actie in de gegeven state.

II. *Beargumenteer waarom we bij model-free control geen gebruik kunnen maken van een value function.*

Bij Model-Free Control worden de keuzes van acties niet gebaseerd op de verwachte rewards van states (dat wat de value functie levert).Deze zijn gebaseerd op de rewards gekregen vanuit eerdere exploratie.

III. *Leg in eigen woorden uit wat 'on-policy' in de naam van het algoritme betekent.*

On-policy betekent dat er een policy gebruikt wordt, die telkens tussen episodes door aangepast wordt. Tegenover off-policy, dat tussen elke episode een nieuwe policy maakt.

IV. *Dit algoritme geeft een deterministische policy. Leg uit of dit een nadeel is voor de agent in onze doolhof.*

Als de policy deterministisch is betekent het dat misschien niet elke state gevonden wordt tijdens exploratie.