Adaptive system

AS2.1 - Model-free prediction control - theorievragen

Naam : Hussin Almoustafa
Studentnummer : 1776495

May 19, 2023

1 Model-free prediction

1.1 Monte-Carlo policy evaluation

1.1.1 Optimale beleidsbepaling ($\pi*$)

Het is duidelijk that de agent de snelste route moet nemen naar de cel met de hoogste beloning, wat naar de cel (0,3) moet zijn vanaf de startcel (3,2). Dit is de optimale doolhof-policy. Om de beloning te ontvangen, moet de agent naar boven en naar rechts bewegen. Om de totale negatieve beloning te verminderen, zal de agent de kortste route naar het doel volgen omdat de beloning in elke cel die niet de doelcel is negatief is. De waardefunctie na oneindige iteraties met een discount factor van 1 (γ = 1) zou er as volgt uitzien:

$$\begin{bmatrix} 40 & -1 & -2 & -3 \\ 30 & -1 & -12 & -2 \\ 20 & -1 & -2 & -3 \\ 10 & -1 & -2 & -3 \end{bmatrix}$$

1.1.2 Value function met $\gamma = 0.5$

De waarde van toekomstige beloningen wordt met de helft verminderd as de disconteringsfactor γ gelijk is aan 0,5. Dit betekent dat de beloning van de terminal state voor elke stap terug van de terminal state wordt gehalveerd. Als gevolg hiervan zou de waarde van de staat net voor de uiteindelijke staat op het optimale pad 40 * 0,5 = 20 zijn, en zo verder.

De toekomstige beloningen worden verminderd als de discount factor wordt gesteld op 0,5 (= 0,5). Na ontelbare iteraties zou de waardefunctie er zo uitzien:

$$\begin{bmatrix} 20.0 & -1.0 & -1.5 & -1.75 \\ 10.0 & -0.5 & -6.0 & -1.0 \\ 5.0 & -0.25 & -1.5 & -1.5 \\ 2.5 & -1.0 & -1.0 & -1.5 \end{bmatrix}$$

1.1.3 Onvolledige value function

Het beleid bereikt niet alle staten in dit specifieke geval. Dit is het resultaat van onze focus op het bereiken van de doelcel met de hoogste beloning, terwijl andere cellen worden genegeerd vanwege hun negatieve beloningen. We hebben dus geen volledige waardefunctie. Het kan nodig zijn voor exploratie te stimuleren zodat het alle omgevingsstaten kan bereiken om een volledige waardefunctie te bereiken.

1.1.4 Any-visit vs. First-visit Monte Carlo

Of er een verschil is tussen first-visit en any-visit Monte Carlo hangt af van de aard van de taak en het beleid. In ons specifieke geval, omdat het beleid deterministisch is en elke staat niet meer dan één keer bezoekt in een aflevering, zal er geen verschil zijn tussen first-visit en any-visit Monte Carlo. Als de beleidskeuzes echter stochastisch waren of de

agent kon terugkeren naar eerdere staten, dan kunnen de resultaten verschillen tussen de twee methoden.

1.2 Temporal difference learning

1.2.1 Het verschil in initialisatie tussen Monte-Carlo en Temporal Difference (TD) learning

Het verschil in initialisatie tussen Monte-Carlo en Temporal Difference (TD) learning is te wijten aan de manier waarop ze de waarde functie schatten. Bij Monte-Carlo evaluatie, wordt de beloning voor elke staat berekend aan het einde van een aflevering, op basis van daadwerkelijk ervaren beloningen. Dit betekent dat de waarde van de terminale staten direct uit de ervaring wordt berekend, ongeacht hun initiële waarden. Daarentegen actualiseert TD learning de waarde schattingen op basis van de huidige waarde schatting en de geschatte toekomstige waarde (die is gebaseerd op de huidige waarde van de volgende staat). Dit betekent dat de initiële waarden een impact kunnen hebben op de uiteindelijke schattingen, en daarom worden de terminale staten geïnitialiseerd op 0 in TD learning, aangezien hun ware waarde inderdaad 0 is (er zijn geen toekomstige beloningen vanuit een terminale staat).

1.2.2 Een voordeel van Temporal Difference learning ten opzichte van het Monte-Carlo algoritme

Een voordeel van Temporal Difference learning ten opzichte van het Monte-Carlo algoritme is dat het sneller convergentie kan bereiken. Dit komt omdat TD learning na elke tijd stap een update uitvoert in plaats van aan het einde van een episode, zoals bij Monte-Carlo. Dit betekent dat het kan leren van onvoltooide episodes, wat bijzonder nuttig is in taken met lange of zelfs oneindige episodes. Bovendien is TD minder variabel dan Monte-Carlo, omdat het updates baseert op de huidige schatting van de waarde functie, in plaats van op de volledige terugkeer van een episode.

Een nadeel van TD learning is echter dat het mogelijk is dat het niet convergeert naar de ware waarde functie wanneer we een function approximator gebruiken (zoals neurale netwerken), terwijl Monte-Carlo in staat is om te convergeren onder mildere voorwaarden. Dit is te wijten aan het feit dat de updates van TD learning gebaseerd zijn op andere schattingen, wat kan leiden tot problemen met stabiliteit en convergentie.

2 Model-Free Control

2.1 On-policy first-visit Monte-Carlo Control

2.1.1 Q-functie

Een Q-functie, ook wel bekend als een actie-waarde functie, geeft de verwachte return (totaal van toekomstige gedisconteerde beloningen) weer die een agent kan behalen door een bepaalde actie uit te voeren in een bepaalde state, gegeven dat het volgt van een specifieke policy. Het is een manier om te bepalen welke actie het beste is om te nemen gegeven een

bepaalde staat. Het wordt vaak aangeduid als $Q(s, a)$ waarbij 's' de staat vertegenwoordigt en 'a' de actie.

2.1.2 Waarom we geen gebruik kunnen maken van een value function in model-free control

In model-free control, kennen we de dynamics van de omgeving niet, wat betekent dat we niet weten welke staat we zullen bereiken door het nemen van een bepaalde actie in een bepaalde staat. Een waarde functie geeft ons de waarde van een staat onder een bepaalde policy, maar het vertelt ons niet welke actie we moeten nemen in een bepaalde staat. Aan de andere kant, een Q-functie geeft ons de verwachte return voor elk paar state-actie, waardoor het een betere keuze is voor model-free control.

2.1.3 'On-policy' in de naam van het algoritme

On-policy' betekent dat het algoritme leert en beslissingen neemt op basis van de policy die het momenteel aan het evalueren of verbeteren is. Met andere woorden, het evalueert en verbetert het beleid dat het momenteel volgt. Dit is in tegenstelling tot 'off-policy' methoden, die leren van een ander beleid dan ze verbeteren.

2.1.4 Of een deterministische policy een nadeel is voor de agent in onze doolhof

Een deterministisch beleid kan een nadeel zijn in een complexe omgeving zoals een doolhof omdat het de agent kan beperken tot een bepaalde set van acties, wat kan leiden tot suboptimale prestaties. Een deterministisch beleid kan leiden tot het missen van de optimale oplossing als de agent vastzit in een suboptimaal pad. Het gebruik van een stochastisch beleid kan dit probleem verlichten door het toestaan van enige mate van exploratie, wat kan helpen bij het ontdekken van betere oplossingen. Echter, als de doolhofomgeving eenvoudig en statisch is en er een duidelijk pad is naar het doel, dan zou een deterministisch beleid voldoende zijn en kan het zelfs voordelig zijn vanwege de consistentie in de genomen acties.