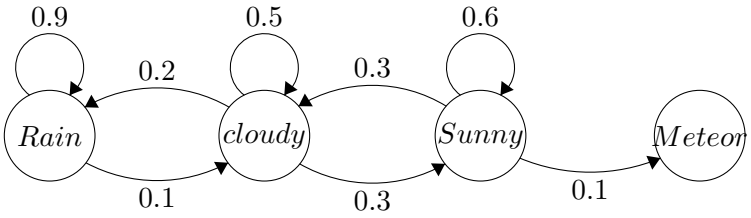
Inleveropgave 1: Model-based Prediction and Control

Max van Kemenade

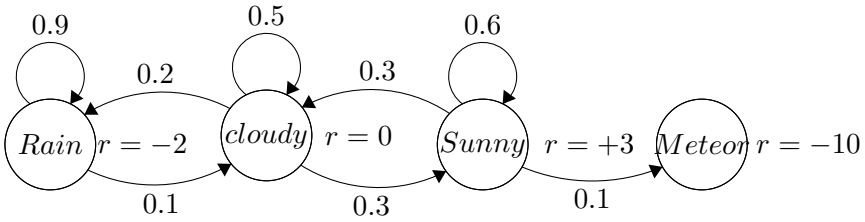
November 24, 2021

1. Prediction

1.1 Markov Chain:



1.2 Markov Reward Process:



1.3 Sampling. Een voorbereiding voor Monte-Carlo Policy Evaluation:
Rain= R, Cloud = C, Sunny = S, Meteor = M

$$C- > R- > C- > S- > S- > S- > S- > M \tag{1}$$

$$Reward = 0 - 2 + 0 + 3 + 3 + 3 + 3 - 10 = 0$$

$$C- > R- > R- > C- > S- > C- > C- > S- > S- > S- > C- > S- > M \tag{2}$$

$$Reward = 0 - 2 - 2 + 0 + 3 + 0 + 0 + 3 + 3 + 3 + 0 + 3 - 10 = 1$$

1.4 De value-function bepalen:

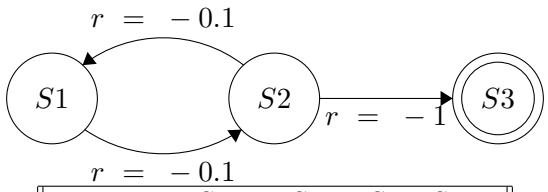
Iteratie	Rain	Cloudy	Sunny	Meteor
0	0	0	0	0
1	-1.8	0.5	0.8	0
2	-3.37	0.64	1.43	0

1.5 Zelf-onderzoek:

1: Als je een discount van 1 hebt zullen alle stappen die gezet zijn evenveel meetellen. Alleen de laatste stappen waarmee daadwerkijkelijk het doel wordt bereikt zullen vaak iets belangrijker zijn dan de stappen die meer in het midden worden gemaakt. Dit zul je bijvoorbeeld zijn bij het boot probleem (link hier: https://www.youtube.com/watch?v=tlOIHko8ySg) Waarbij boosters pakken belangrijker is dan de race finishen.

2: Met een discount van 1 blijft een RL-model oneindig leren, terwijl met een discount tussen 1 en 0 zal die langzaam stoppen met leren.

2. Control met Value Iteration



Iteratie	S1	S2	S3	Stop
0	0	0	0	
1	-0.1	-0.55	0	
2	-0.65	-1	0	
3	-1.1	-1	0	
4	-1.1	-1	0	←

In de eerste stap is de value voor naar links en naar rechts hetzelfde. Daarom neem ik de gemiddelde value van naar S1 gaan en naar S3 gaan voor S2. Daarna wordt het echt duidelijk dat naar S3 gaan beter is, omdat s3 een hogere value heeft. Met deze manier is uiteindelink op iteratie 4 zeker dat de value van de 3 states hetzelfde blijven.