Wprowadzenie do uczenia ze wzmocnieniem

część 1

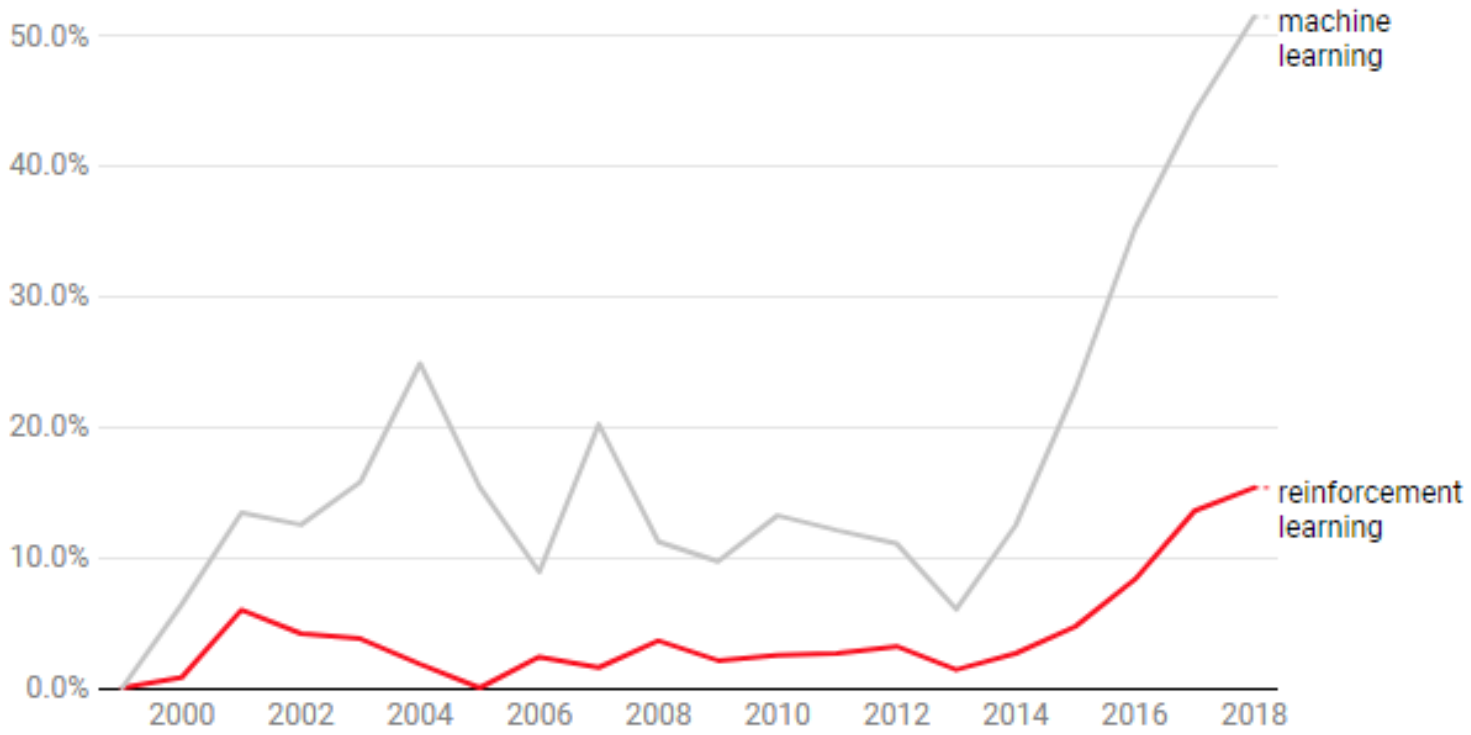


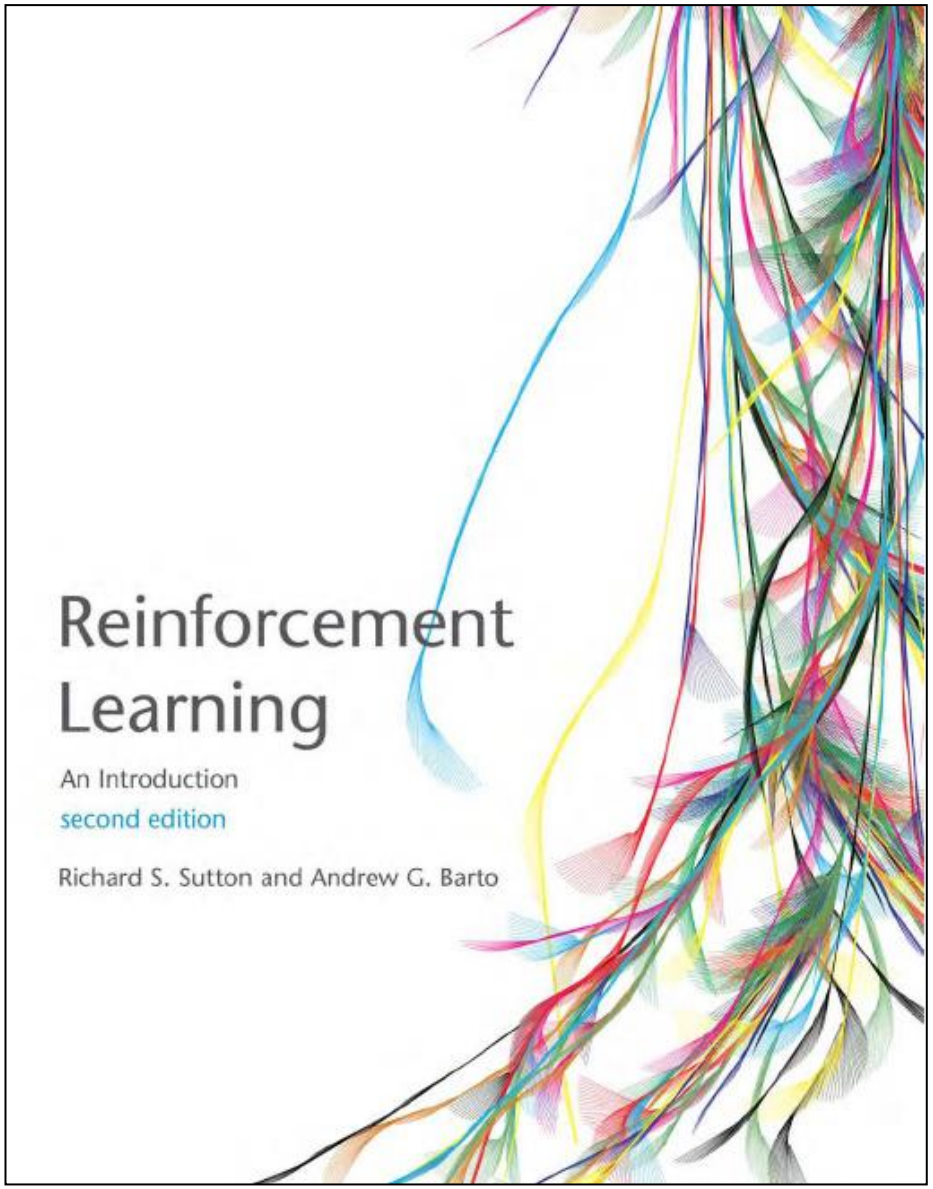
Chart: MIT Technology Review • Source: arXiv.org • Created with Datawrapper

Podręcznik

Richard S. Sutton
Andrew G. Barto

Reinforcement Learning:
An Introduction

(Adaptive Computation and Machine Learning series)



Uczenie maszynowe

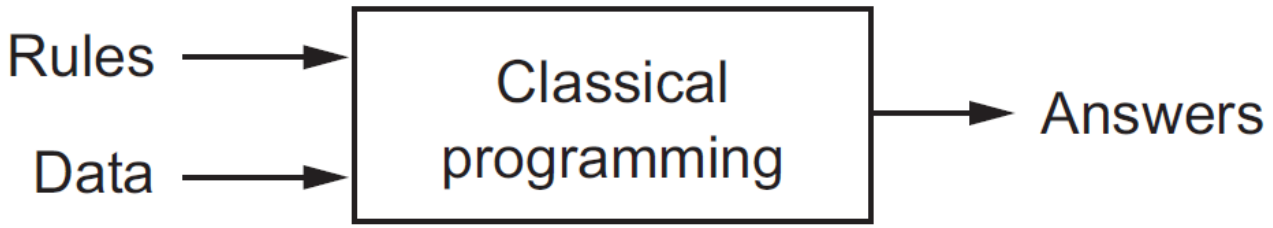
Uczenie maszynowe powstało z pytania:

Czy maszyna może wyjść poza naszą wiedzę (eksperta) i samemu nauczyć się jak wykonać specyficzne zadanie?

Inaczej:

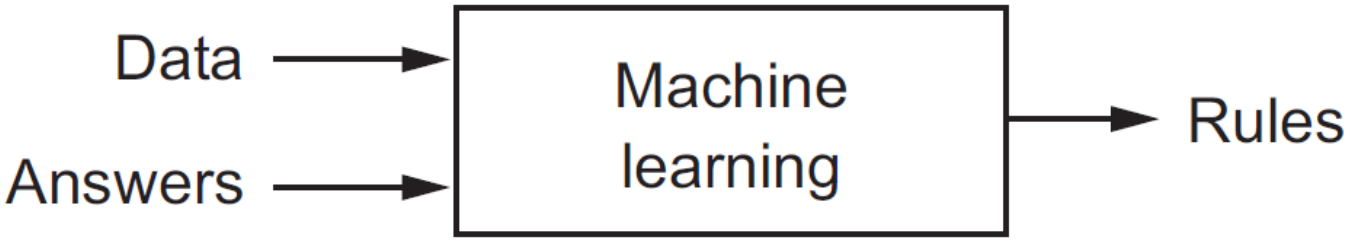
Czy maszyna jest w stanie nas zaskoczyć?

Stary paradygmat programowania:



Uczenie maszynowe

Nowy paradygmat programowania!



System działający w oparciu o uczenie maszynowe jest raczej **trenowany** (**uczony**) niż programowany!

Uczenie maszynowe

Podział uczenia maszynowego:

Uczenie z nauczycielem (*supervised learning*)
Przykłady: generowanie etykiet, detekcja obiektów na zdjęciach.

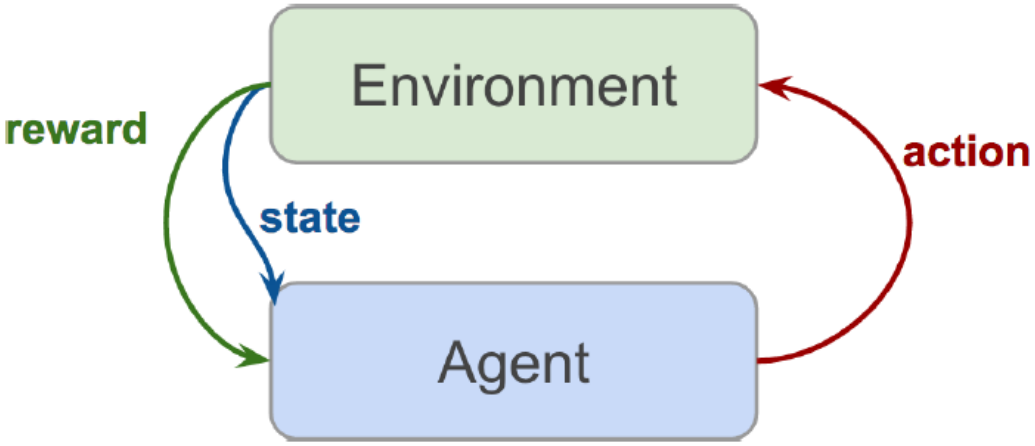
Uczenie bez nauczyciela (*unsupervised learning*)
Przykłady: analiza danych – redukcja wymiarów, grupowanie.

Samouczenie (*self-supervised learning*)
Przykłady: przewidywanie następnego kadru na filmie, przewidywanie następnego słowa w zdaniu.

Uczenie maszynowe

Podział uczenia maszynowego (cd):

Uczenie przez wzmacnianie (*reinforcement learning*)



Uczenie maszynowe

Ważne podsumowanie:

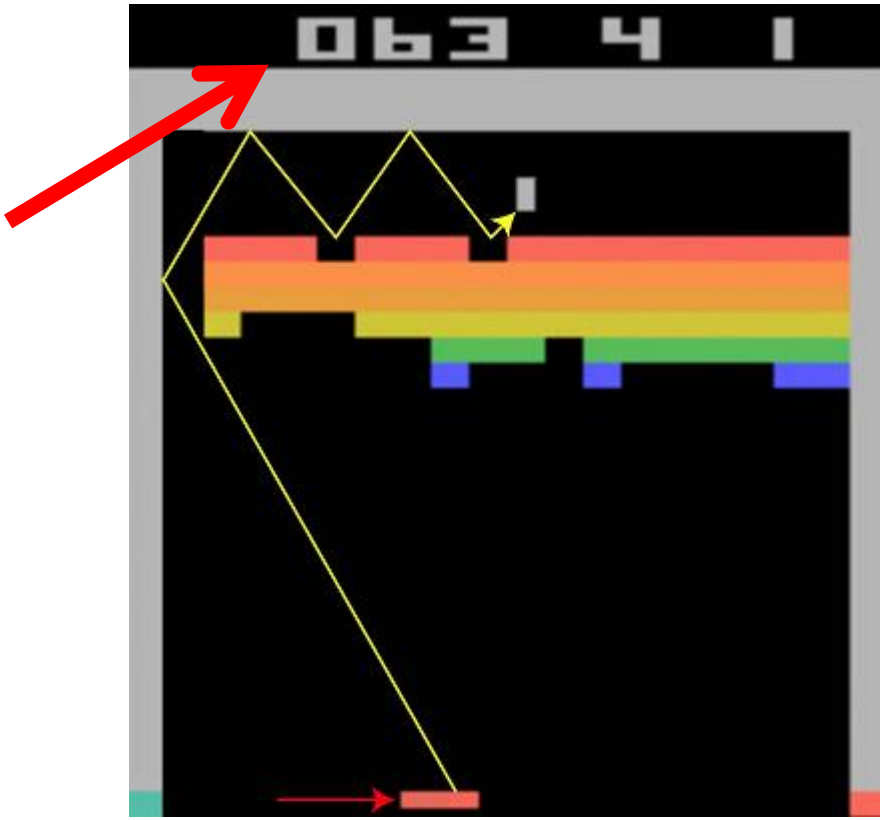
Uczenie	Dane	Wart. oczekiwane
Z nauczycielem	Tak	Tak
Bez nauczyciela	Tak	Nie
Przez wzmacnianie	Nie	Nie

Uczenie przez wzmacnianie

Uczenie przez wzmacnianie

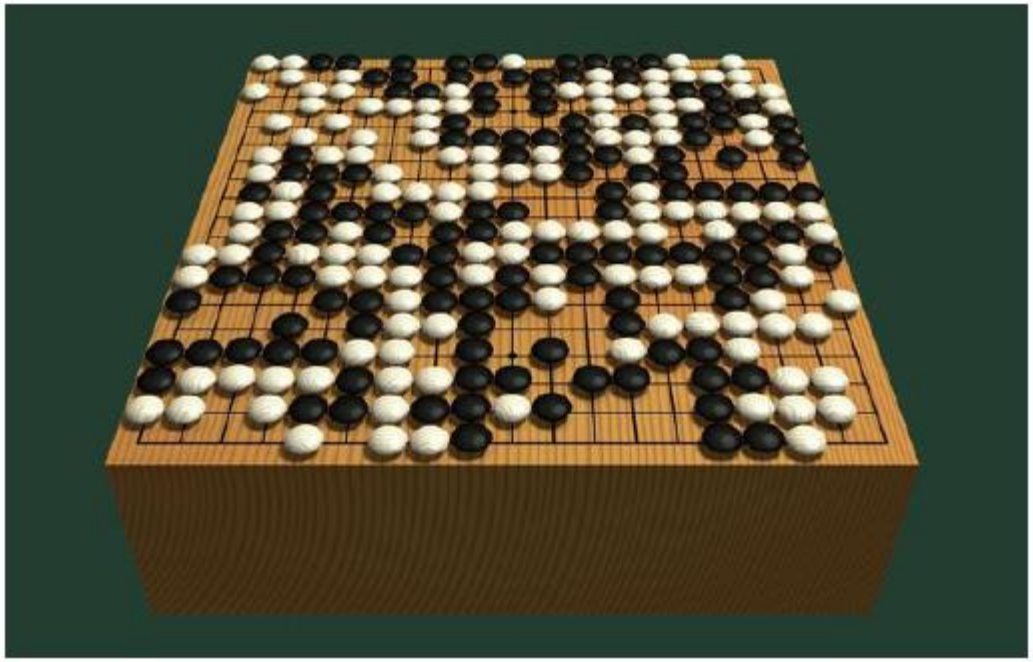
Uczenie przez wzmacnianie

Gra Breakout na komputer Atari



Czas nauki: 240 minut

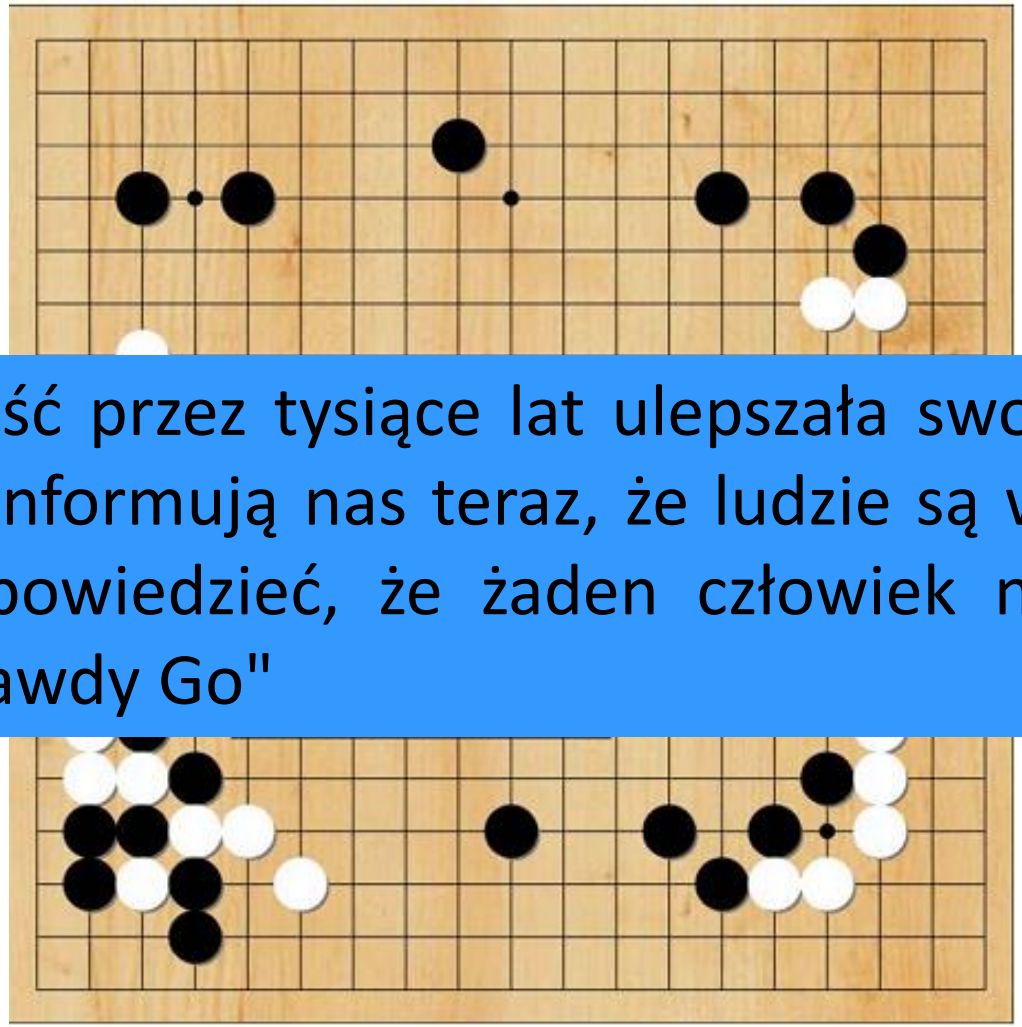
Uczenie przez wzmacnianie



Uczenie przez wzmacnianie

AlphaGO	Lee Se-dol
1202 CPUs, 176 GPUs,	1 Human Brain,
100+ Scientists.	1 Coffee.

Uczenie przez wzmacnianie



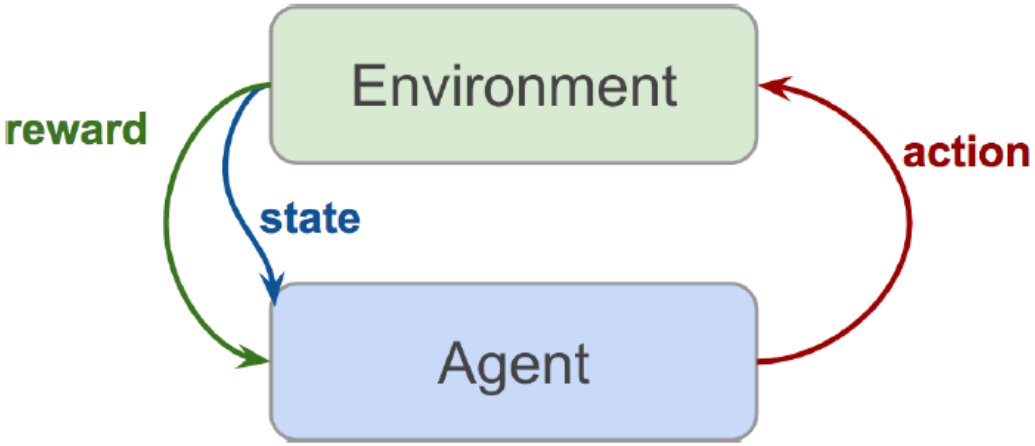
"Gdy ludzkość przez tysiące lat ulepszała swoją taktykę, komputery informują nas teraz, że ludzie są w błędzie... Chciałbym powiedzieć, że żaden człowiek nie dotknął krawędzi prawdy Go"

Uczenie przez wzmacnianie

Artificial General Intelligence

Idea

Celem RL nie jest aproksymowanie pewnego nieznanego odwzorowania przez generalizację na podstawie zbioru przykładów trenujących.



Systemowi uczącemu się ze wzmocnieniem nie są dostarczane żadne przykłady trenujące, a jedynie wartościująca informacja trenująca (nagroda), oceniające jego dotychczasową skuteczność.

Idea

U podstaw uczenia się ze wzmocnieniem leżą dynamiczne interakcje ucznia ze środowiskiem, w którym działa, realizując swoje zadanie.

Interakcje te odbywają się dyskretnych (na ogół) krokach czasu i polegają na obserwowaniu przez ucznia kolejnych stanów środowiska oraz wykonywaniu wybranych zgodnie z jego obecną strategią decyzyjną akcji.

Po wykonaniu akcji uczeń otrzymuje nagrody, które stanowią pewną miarę oceny jakości jego działania. Wykonanie akcji może również powodować zmianę stanu środowiska.

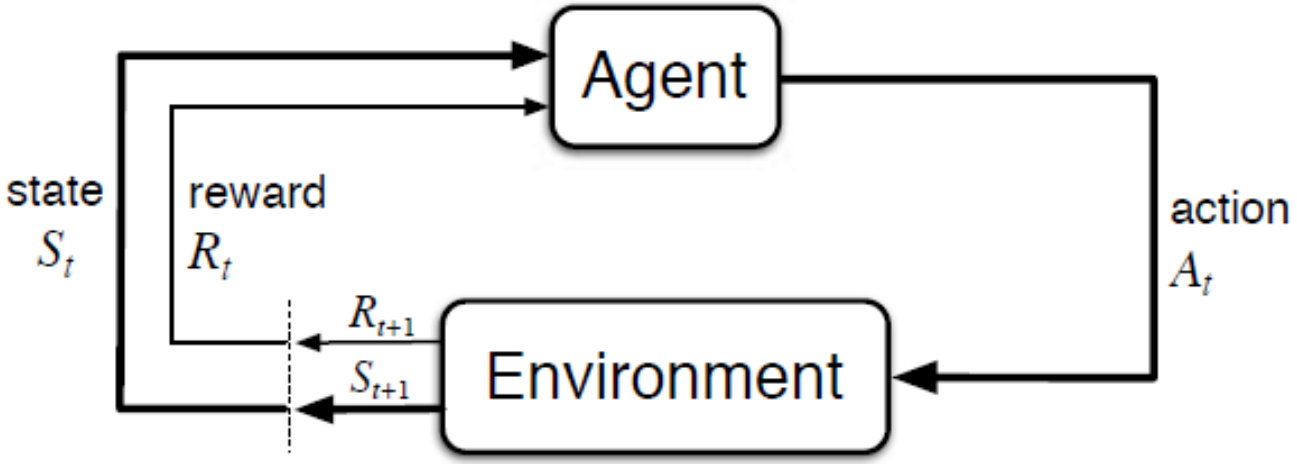
Idea

W każdym kroku czasu:

- obserwuj aktualny stan środowiska
- wybierz akcję do wykonania w stanie
- wykonaj akcję
- obserwuj wzmocnienie (nagrodę) i następny stan
- ucz się na podstawie doświadczenia

Procesy decyzyjne Markowa

Modelem matematycznym uczenia się ze wzmocnieniem (a właściwie modelem środowiska) jest proces decyzyjny Markowa (Markov decision process, MDP).



Procesy decyzyjne Markowa

Agent oddziałuje ze środowiskiem w dyskretnych krokach czasowych:

$$t = 0, 1, 2, 3, \ldots$$

W każdej chwili czasu t agent otrzymuje pewną reprezentację stanu środowiska $S_t \in S$, i na tej podstawie podejmuje akcję $A_t \in A(s)$.

W następnym kroku $t+1$ agent otrzymuje nagrodę $R_{t+1} \in R \subset \mathbb{R}$ i przechodzi do nowego stanu $S_{t+1} \in S$.

Oddziaływanie agenta i środowiska możemy opisać za pomocą trajektorii:

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, S_3, A_3, \ldots$$

Procesy decyzyjne Markowa

W skończonym MDP zbiory S, A, R są skończone.

W tym przypadku zmienne losowe R_t i S_t mają dobrze zdefiniowane dyskretne rozkłady prawdopodobieństwa zależne tylko od poprzedniego stanu S_{t-1} i działania A_{t-1}.

Oznacza to, że dla poszczególnych wartości tych zmiennych losowych, $s' \in S$ i $r \in R$, istnieje prawdopodobieństwo wystąpienia tych wartości w czasie t, przy założeniu pewnych wartości wcześniejszych (w czasie $t-1$) stanu i działania ($s \in S$ i $a \in A(s)$):

$$p(s', r \mid s, a) \doteq \Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\},$$

dla wszystkich $s, s' \in S, r \in R$ i $a \in A(s)$.

Procesy decyzyjne Markowa

Funkcja:

$$p : \mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$$

p definiuje dynamikę MDP.

Funkcja p określa rozkład prawdopodobieństwa dla każdego wyboru s i a czyli:

$$\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \,|\, s, a) = 1,$$

dla każdego $s \in S$ i $a \in A(s)$.

Procesy decyzyjne Markowa

Własność Markowa:

Stan zawiera całą informację o wszystkich aspektach oddziaływania agent-środowisko z przeszłości, które mają wpływ na przyszłość.

Własność Markowa oznacza, że prawdopodobieństwo każdej możliwej wartości dla S_t i R_t zależy tylko od bezpośrednio poprzedzającego stanu i działania (S_{t-1} i A_{t-1}) i nie zależy od jeszcze wcześniejszych stanów i działań.

W czasie zajęć zakładamy, że własność Markowa zachodzi.

Procesy decyzyjne Markowa

Z funkcji dynamiki p można obliczyć wszystko, co ktoś mógłby chcieć wiedzieć o środowisku.

Prawdopodobieństwa przejścia stanu (state-transition probabilities):

$$p(s'|s,a) \doteq \Pr\{S_t = s' \mid S_{t-1} = s, A_{t-1} = a\} = \sum_{r \in \mathcal{R}} p(s', r|s, a)$$

Wartość oczekiwana nagrody dla pary stan-akcja:

$$r(s,a) \doteq \mathbb{E}[R_t \mid S_{t-1} = s, A_{t-1} = a] = \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r|s, a)$$

Procesy decyzyjne Markowa

Ramy MDP są abstrakcyjne i elastyczne i mogą być stosowane do wielu różnych problemów, na wiele różnych sposobów.

- Kroki czasowe nie muszą odnosić się do ustalonych interwałów czasie - mogą odnosić się do dowolnych kolejnych etapów podejmowania decyzji i działania.

- Akcje mogą być związane z operacjami niskiego poziomu, takimi jak napięcia przykładane do silników ramienia robota lub operacjami na wysokim poziomie, takimi zjedzenie (lub nie) obiadu czy wybór studiów.

  Akcja może też sprowadzać się to wyboru tego o czym agent pomyśli albo na czym skupi swoją uwagę.

Procesy decyzyjne Markowa

- Podobnie stany mogą przyjmować różnorodne formy. Mogą być całkowicie określone przez pomiary niskiego poziomu, takie jak bezpośrednie odczyty z czujników, ale mogą być bardziej szczegółowe i abstrakcyjne np. mogą to być symboliczne opisy przedmiotów w pomieszczeniu.

Przyjmujemy zasadę, że wszystko co nie może być zmienione dowolnie przez agenta jest częścią środowiska.

Granica między agentem i środowiskiem reprezentuje granicę kontroli kontrola agenta, a nie jego wiedzy.

W szczególności agent nie jest w stanie wpływać na wysokość nagrody.

Procesy decyzyjne Markowa

Przykład

Przypuśćmy, że uczenie ze wzmocnieniem jest stosowane w celu określenia temperatury i tempa mieszania dla bioreaktora (kadź zawierająca składniki odżywcze i bakterie stosowane do produkcji użytecznych chemikaliów).

Akcje: docelowe temperatury i docelowe szybkości mieszania (wektor)

Stany: odczyty parametrów, być może przefiltrowane i opóźnione, oraz dane wejściowe reprezentujące składniki w kadzi i docelowy związek chemiczny. Stany są wektorami.

Nagroda to szybkość z jaką użyteczny związek chemiczny jest wytwarzany przez bioreaktor (liczba, a nie wektor).

Procesy decyzyjne Markowa

Przykład

Robot sprzątający puste puszki.

Stan opisuje poziom naładowania baterii:

$$S = \{\texttt{high}, \texttt{low}\}$$

Akcje: search, wait, recharge

$$A(\texttt{low}) = \{\texttt{search}, \texttt{wait}, \texttt{recharge}\}$$

$$A(\texttt{high}) = \{\texttt{search}, \texttt{wait}\}$$

Nagrody: każda zebrana puszka +1

$$r_{\texttt{search}} > r_{\texttt{wait}}, 0, -3$$

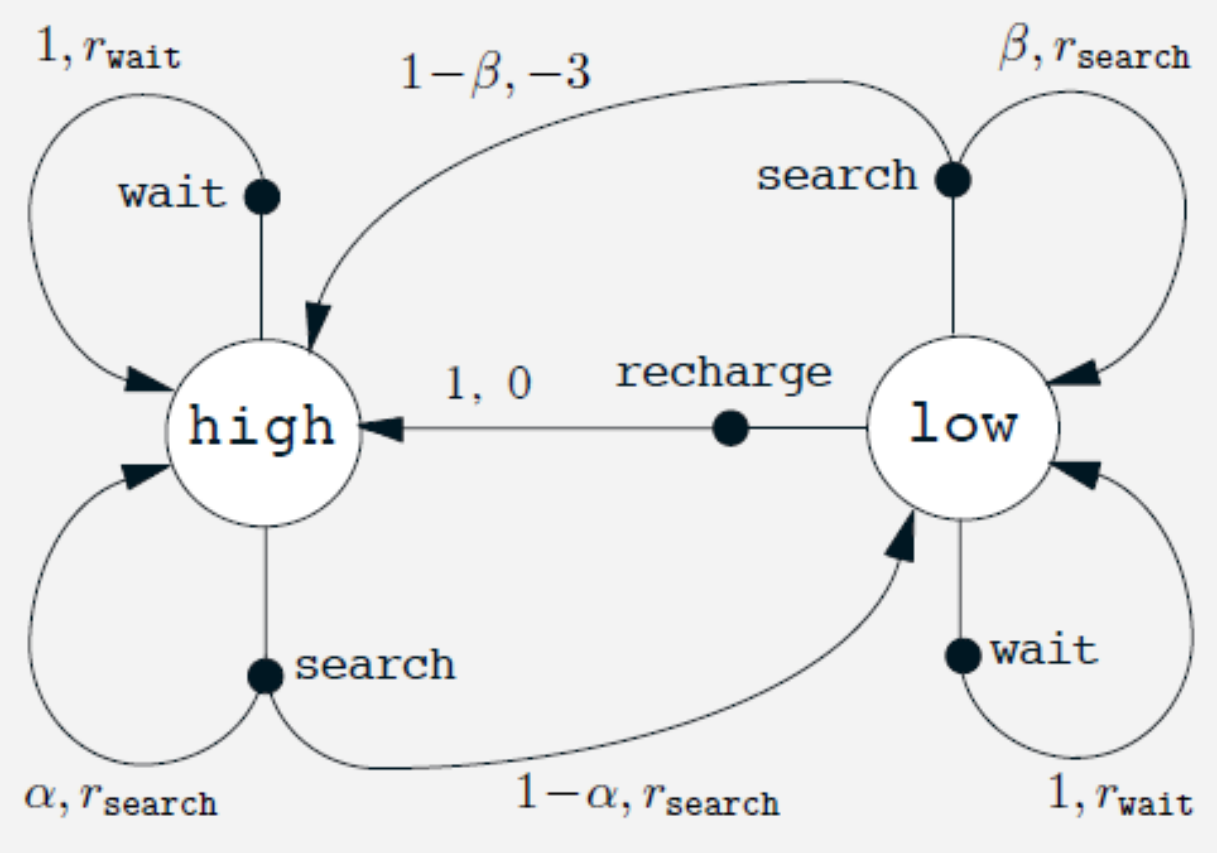
MDP - przykład

Robot sprzątający jako MDP:

s	a	s'	$p(s' \mid s, a)$	$r(s, a, s')$
high	search	high	α	$r_{\texttt{search}}$
high	search	low	$1 - \alpha$	$r_{\texttt{search}}$
low	search	high	$1 - \beta$	-3
low	search	low	β	$r_{\texttt{search}}$
high	wait	high	1	$r_{\texttt{wait}}$
high	wait	low	0	-
low	wait	high	0	-
low	wait	low	1	$r_{\texttt{wait}}$
low	recharge	high	1	0
low	recharge	low	0	-

MDP - przykład

Robot sprzątający – graf przejścia:



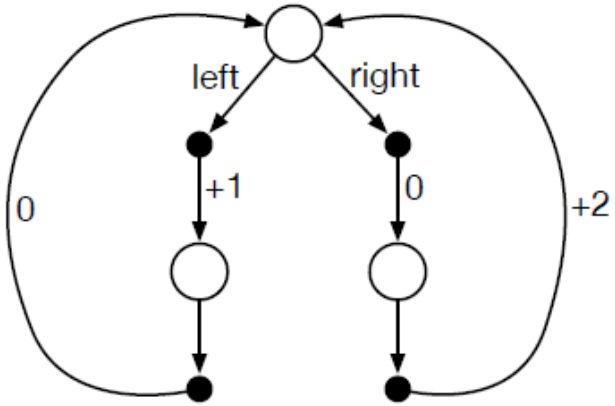
Cele i nagrody

Agent w każdej chwili czasu otrzymuje nagrodę $R_t \in \mathbb{R}$.

Celem agenta jest uzyskanie jak największej całkowitej nagrody (suma nagród otrzymanych w kolejnych krokach czasowych).

Nie oznacza to maksymalizacji natychmiastowej nagrody, ale całkowitej, sumarycznej nagrody.

Przykład:



Cele i nagrody

Hipoteza nagrody:

Wszystko co rozumiemy przez cel agenta, może być pomyślane jako maksymalizacja oczekiwanej wartości łącznej sumy otrzymanej nagrody.

Sygnał nagrody jest naszym sposobem przekazywania agentowi tego co chcemy osiągnąć, a nie jak chcemy to osiągnąć.

Przykład Agent szachowy powinien być nagradzanym tylko za faktyczne wygrane, a nie za osiąganie celów cząstkowych, takich jak zabieranie ich części przeciwnika lub zdobycie kontroli nad środkiem planszy.

Cele i nagrody

Spróbujmy jakoś sformalizować fakt, że celem agenta jest zmaksymalizowanie łącznej nagrody, którą otrzymuje w dłuższej perspektywie?

Nagrody otrzymywane przez agenta po chwili t oznaczmy przez:

$$R_{t+1}, R_{t+2}, R_{t+3}, R_{t+4}, \ldots$$

Ogólnie rzecz biorąc, staramy się zmaksymalizować oczekiwany zwrot, gdzie zwrot (oznaczony G_t) jest zdefiniowany jako pewna określona funkcja powyższej sekwencji nagród.

Cele i nagrody

W najprostszym przypadku zwrot G_t jest sumą nagród:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \ldots + R_T$$

gdzie T jest ostatnim krokiem czasowym.

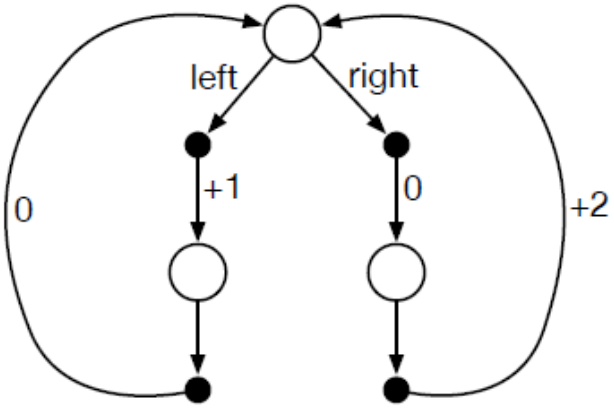
Taka definicja zwrotu ma sens w przypadku gdy istnieje (naturalny) ostatni krok czasu T tzn. wtedy gdy oddziaływanie agent-środowisko można w sposób naturalny podzielić na tzw. epizody.

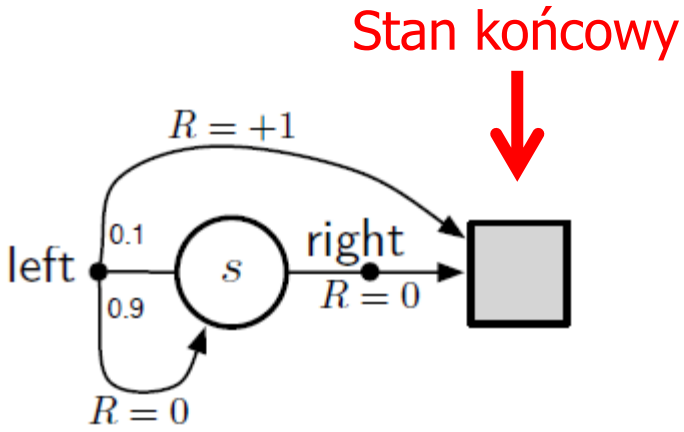
Każdy epizod kończy się w stanie końcowym, po którym następuję reset czyli powrót do pewnego stanu początkowego.

Cele i nagrody

Nawet jeśli odcinki kończą się na różne sposoby (na przykład zwycięstwem lub przegraną) rozpoczyna się następny epizod niezależnie od tego, jak zakończył się poprzedni.

Tak więc wszystkie epizody można uznać za zakończone w tym samym stanie końcowym, z różnymi nagrodami za różne wyniki.



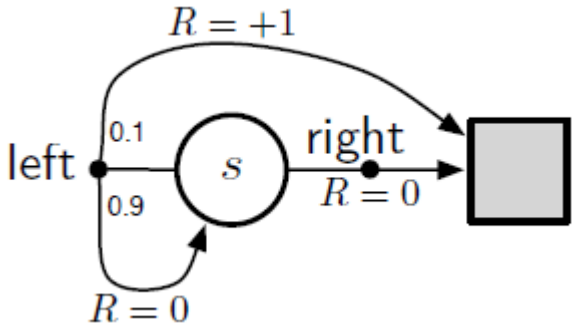


Cele i nagrody

Zadania, które można podzielić na epizody nazywamy epizodycznymi.

W zadaniach epizodycznych czasami potrzebujemy rozróżnić zbiór wszystkich stanów nieterminalnych S od zbioru wszystkich stanów + stan końcowy - S^+.

Czas zakończenia T jest zmienną losową, która zazwyczaj różni się w różnych epizodach.



S, right, 0, S_T

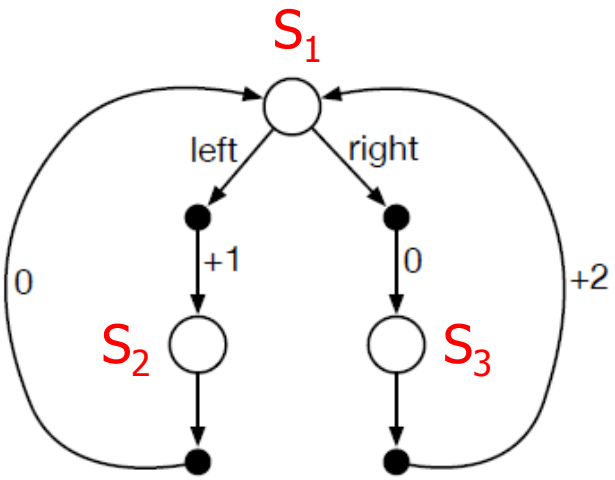
S, left, 0, S, left, 0, S, right, 0, S_T

S, left, 0, S, left, 1, S_T

Cele i nagrody

W wielu przypadkach interakcja agent-środowisko nie ulega naturalnemu podziałowi na możliwe do zidentyfikowania epizody, ale ciągnie się bez ograniczeń.

Zadania takie nazywamy ciągłymi (continuing tasks).



S_1, right, 0, S_3, A, 2, S_1 …

S_1, left, 1, S_2, A, 0, S_1 …

Cele i nagrody

W przypadku zadań ciągłych zastosowanie formuły:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \ldots + R_T$$

jest kłopotliwe bo $T=\infty$.

Przykład W przypadku agenta, który otrzymuje nagrodę +1 np. co drugi krok czasowy wartość $G_t = \infty$.

Dlatego w zwykle używamy definicji zwrotu, która jest nieco bardziej złożona koncepcyjnie, ale znacznie prostsza matematycznie.

Cele i nagrody

Zgodnie z tym podejściem, agent próbuje wybrać akcje, tak aby suma zdyskontowanych nagród, które otrzymuje w przyszłości była maksymalna.

W szczególności wybiera wartość A_t, tak aby zmaksymalizować oczekiwany zdyskontowany zwrot:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

gdzie $\gamma \in [0, 1]$ to tzw. współczynnik dyskontowania.

Cele i nagrody

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Warto zauważyć, że*:*

- Nagroda zdobyta w k-tym kroku jest warta γ^{k-1} razy jej wartość gdyby była zdobyta natychmiast.
- Dla $\gamma < 1$ wartość zwrotu jest ograniczona o ile ograniczony jest ciąg nagród $\{R_t\}$.

  Przykład Dla $\gamma < 1$ i wszystkich nagród $R_t = 1$ otrzymujemy:

$$G_t = \sum_{k=0}^{\infty} \gamma^k = \frac{1}{1-\gamma}$$

Cele i nagrody

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Warto zauważyć, że:

- Dla $\gamma = 1$ agent maksymalizuje tylko natychmiastową nagrodę.
- Powyższa formuła działa dla wszystkich kroków czasu $t<T$, nawet jeśli zakończenie nastąpi w kroku $t+1$, jeżeli przyjmiemy $G_T=0$. To często ułatwia obliczanie zwrotów z ciągów nagród.

Cele i nagrody

Łatwo zauważyć, że:

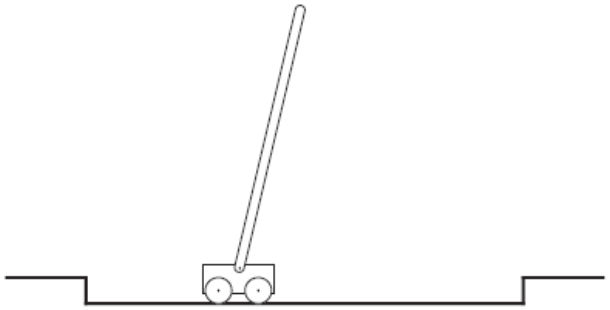
$$\begin{aligned} G_t &\doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \\ &= R_{t+1} + \gamma\left(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + \cdots\right) \\ &= R_{t+1} + \gamma G_{t+1} \end{aligned}$$

Czyli do obliczenia zwrotu w kroku t wystarczy nagroda i zwrot w chwili $t+1$.

Cele i nagrody

Przykład (pole balancing)

Poruszamy wózkiem w prawo i lewo, tak aby jak najdłużej utrzymać drążek w przedziale pewnych kątów.



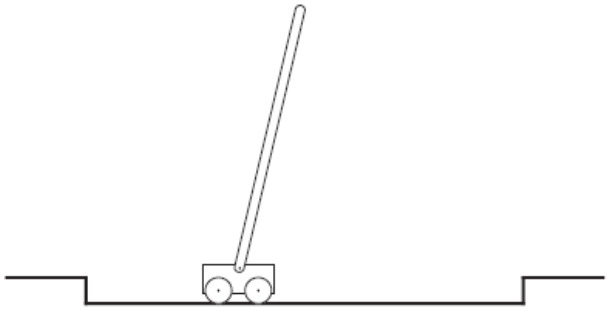
Zadanie to może być w oczywisty sposób traktowane jako epizodyczne, gdzie epizody to wielokrotne próby utrzymania drążka w pionie.

Nagrodą w tym przypadku może być +1 za każdy krok, w którym kąt wychylenia drążka pozostał w pewnych granicach. Zwrot za każdym razem byłby liczbą kroków do momentu niepowodzenia.

Cele i nagrody

Przykład (pole balancing)

Możemy też zadanie to traktować jako ciągłe ze zdyskontowaniem.



W tym przypadku nagrodą będzie -1 przy każdym niepowodzeniu i 0 w pozostałych sytuacjach.

Zwrot w tym przypadku będzie równy $-\gamma^{K}$, gdzie K jest liczbą kroków przed niepowodzeniem.

Cele i nagrody

Podsumujmy informacje o zwrotach (celach).

Zadania epizodyczne:

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T$$

Zadania nieprzerwane:

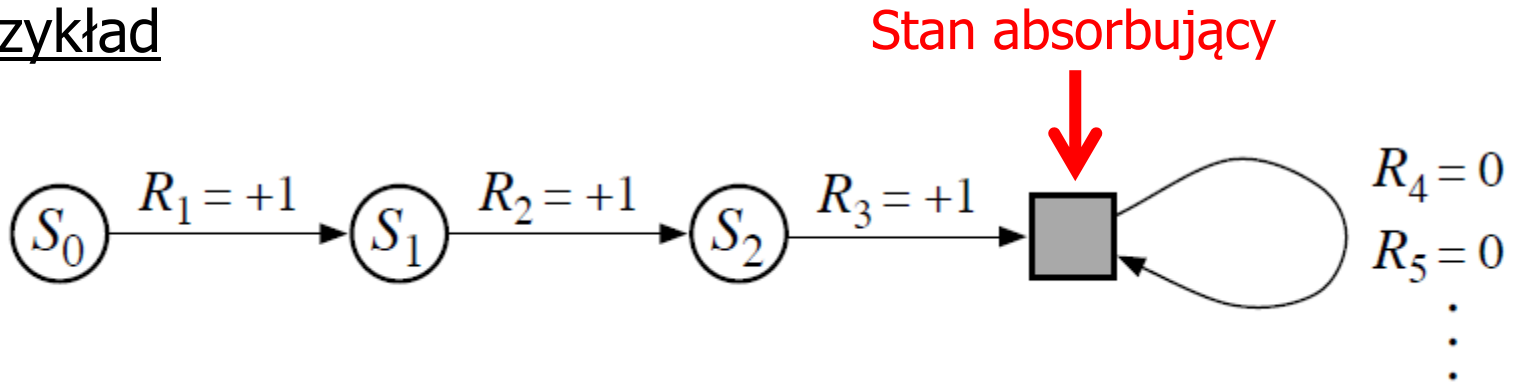
$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Okazuje się, że możliwa jest unifikacja na poziomie opisu matematycznego zadań epizodycznych i ciągłych.

Cele i nagrody

Unifikacja polega na wprowadzeniu specjalnego stanu tzw. absorbującego, który przechodzi tylko na siebie i który generuje tylko nagrody o wartości zero.

Przykład



Rozpoczynając od stanu S_t agent otrzymuje nagrody:

1, 1, 1, 0, 0, 0,…

Cele i nagrody

Możemy zatem zdefiniować zwrot jako:

$$G_t \doteq \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$$

przy czym dopuszczamy sytuację w której $G_t = \infty$ lub $\gamma = 1$ (ale nie jednocześnie).

Strategie i funkcje wartości

Niemal wszystkie algorytmy uczenia przez wzmocnienie pozwalają szacować funkcję wartości – funkcję stanów (lub stanów stan-akcja), która określa „jak dobre” jest to, że agent jest w danym stanie (lub jak dobrze jest wykonać daną czynność w danym stanie).

Pojęcie "jak dobre" jest tutaj zdefiniowane pod kątem przyszłych nagród, których można się spodziewać lub, bardziej precyzyjnie, za pomocą wartości oczekiwanej zwrotu.

Oczywiście nagrody, jakich może oczekiwać agent w przyszłości zależą od tego, jakie działania podejmie.

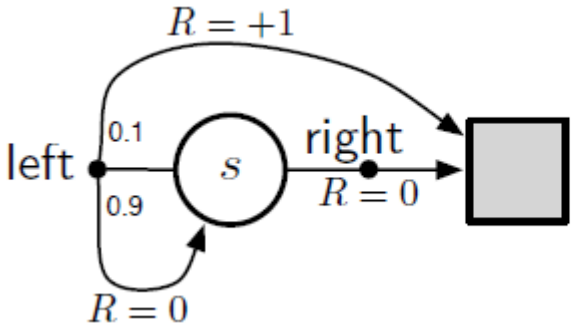
Strategie i funkcje wartości

Funkcje wartości są zatem zdefiniowane w odniesieniu do konkretnych sposobów działania, zwanych politykami lub strategiami.

Formalnie strategia jest odwzorowaniem ze stanów na prawdopodobieństwa wyboru każdej możliwej akcji.

Jeśli agent stosuje strategię π w kroku t wówczas $\pi(a|s)$ jest prawdopodobieństwem tego, że $A_t = a$ jeśli $S_t = s$.

Przykład



$$\pi(\text{left}|s) = 1$$

$$b(\text{left}|s) = \frac{1}{2}$$

Strategie i funkcje wartości

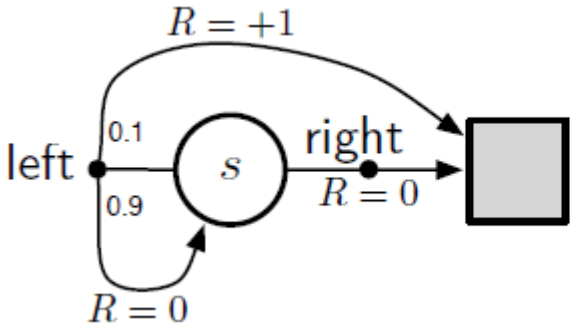
Strategie dzielimy na dwie kategorie: deterministyczne i stochastyczne.

Strategia jest deterministyczna jeżeli w dowolnym stanie s wybór pewnej akcji a ma prawdopodobieństwo równe 1. Wówczas zapisujemy to następująco:

$$\pi(s) = a$$

Strategia deterministyczna

Przykład



$$\pi(\text{left}|s) = 1$$

$$b(\text{left}|s) = \frac{1}{2}$$

Strategia stochastyczna

Strategie i funkcje wartości

Funkcja wartości stanu s w ramach strategii π oznaczona $v(s)$, jest oczekiwanym zwrotem w sytuacji gdy zaczynamy od stanu s i potem stosujemy strategię π.

Formalnie funkcję wartości stanu definiujemy tak:

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right]$$

dla każdego $s \in S$.

$E_\pi[\cdot]$ jest wartością oczekiwaną zmiennej losowej przy założeniu, że agent stosuje strategię π i t jest dowolnym krokiem czasu.

Strategie i funkcje wartości

Podobnie definiujemy funkcję wartości akcji:

$$q_\pi(s,a) \doteq \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s, A_t = a\right]$$

dla każdego $s \in S$ i $a \in A(s)$.

Wartości funkcji $v(s)$ i q(s,a) można oszacować na podstawie doświadczenia (→ metody Monte-Carlo).

Dla nas ważna jest pewna własność rekurencyjna funkcji wartości.

Strategie i funkcje wartości

W przypadku każdej strategii π i dowolnego stanu s następująca zależność zachodzi między wartością stanu s i wartością jego możliwego następcy s'.

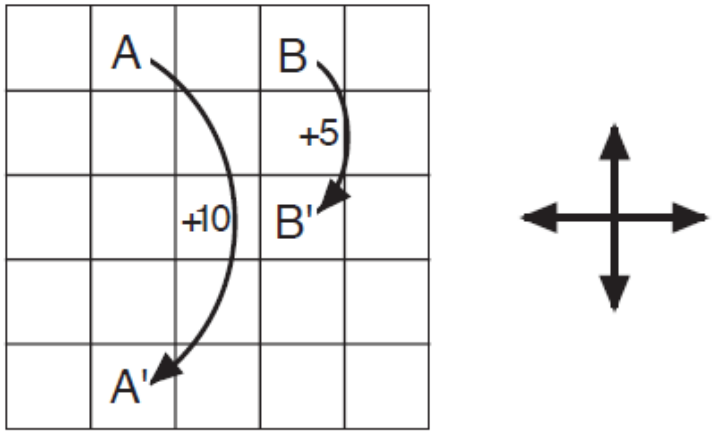
$$
\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r \,|\, s, a) \Big[r + \gamma \mathbb{E}_\pi[G_{t+1} | S_{t+1} = s'] \Big] \\
&= \sum_a \pi(a|s) \sum_{s', r} p(s', r \,|\, s, a) \Big[r + \gamma v_\pi(s') \Big]
\end{aligned}
$$

dla każdego $s \in S$.

Jest to tzw. równanie Bellmana dla funkcji stanu $v(s)$.

Strategie i funkcje wartości

Przykład (Gridworld)



Stany środowiska to 25 pól.

Na każdym polu możliwe są akcje: **`north,south,east, west`**

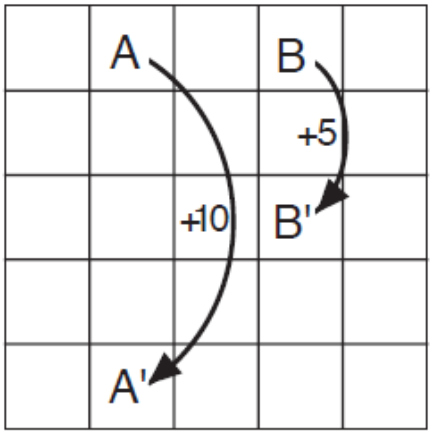
Akcja przenosząca agenta poza obszar kwadratu nie zmienia jego położenia i związana jest z nagrodą -1.

Pozostałe akcje dają nagrodę 0 z wyjątkiem (dowolnych) akcji na polach A i B – dowolna akcja przenosi A→A′, B→B′.

Strategie i funkcje wartości

Przykład (Gridworld)

Załóżmy, że agent wybiera wszystkie cztery akcje z jednakowym prawdopodobieństwem we wszystkich stanach.



Dla wartości $\gamma = 0.9$ funkcji stanu $v(s)$ wygląda następująco:

3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

Funkcja $v(s)$ jest rozwiązaniem układu równań z poprzedniego slajdu.

Koniec części 1