Sztuczna inteligencja. Markowowskie procesy decyzyjne i uczenie ze wzmocnieniem

Paweł Rychlikowski

Instytut Informatyki UWr

4 czerwca 2018

MDP. Przypomnienie

- Uogólnienie zadania przeszukiwania, w którym akcje są niedeterministyczne.
- Nagrody wypłacane są w sposób ciągły, co oznacza, że można rozważać nieskończone ciągi akcji (discount factor, $\gamma < 1$)
- Rozwiązaniem MDP jest polityka, czyli stwierdzenie, co mamy robić w danym stanie.

Algorytmy. Przypomnienie

- **Policy iteration**: wyznaczanie wartości polityki (ile średnio na niej zarobimy, startując w stanie s)
- **Value iteration**: wyznaczenie wartości stanów dla optymalnej polityki
  - Wyznacza też politykę: (w przybliżeniu) idziemy do stanu o najlepszej wartości.
  - Dokładniej: wykonujemy akcję, która daje największą wartość oczekiwaną rezultatu.

Gridworlds. Wulkany

Sprawdzamy działanie algorytmu Value Iteration w świecie wulkanów.

`http://web.stanford.edu/class/cs221/lectures/index.html#inclu`

Zadania

1) Jak wyznaczyć najkrótszą ścieżkę (dwa sposoby)?
2) Gdzie jest przejście między Searching Problem a MDP?
3) Kiedy warto odwiedzić wioskę?
4) Ile iteracji jest potrzebne?

Warunki zbieżności Value Itaration

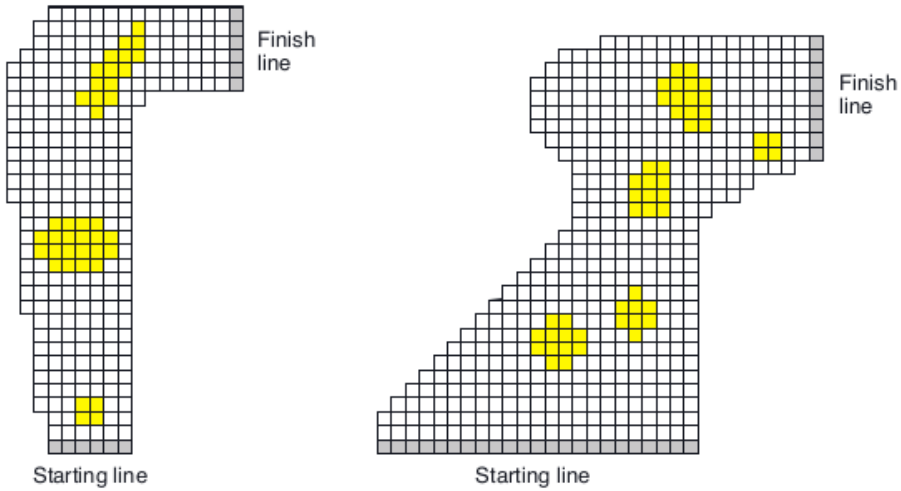
Mówiliśmy, że algorytm jest zbieżny, jeżeli zachodzi któryś z warunków

- $\gamma < 1$
- Graf MDP jest acykliczny

Uwaga

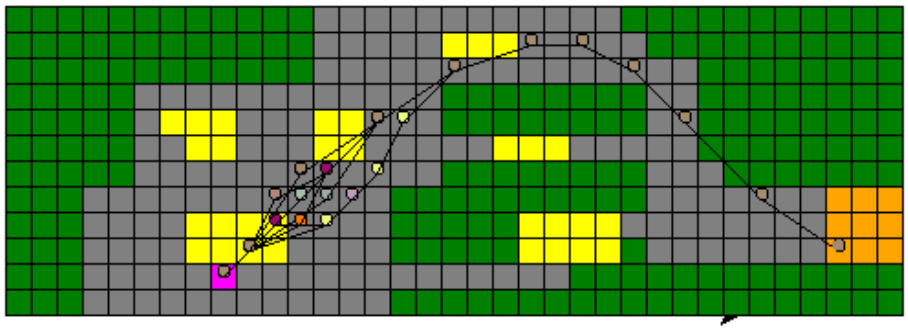
Zwróćmy uwagę na to ci się dzieje, jeżeli $\gamma = 1$ i mamy cykl.
Dla niezerowych nagród na krawędziach cyklu wartość oczekiwana może być nieokreślona

Przypomnienie. Wyścigi samochodzików.



- Prędkość dyskretna, akcja to zmiana prędkości, olej wprowadza losowość

Wynik algorytmu Value Iteration



Zwróćmy uwagę, że bez żadnych dodatkowych obliczeń można umieszczać w innych miejscach punkt startowy.

Jeszcze o autach i oleju

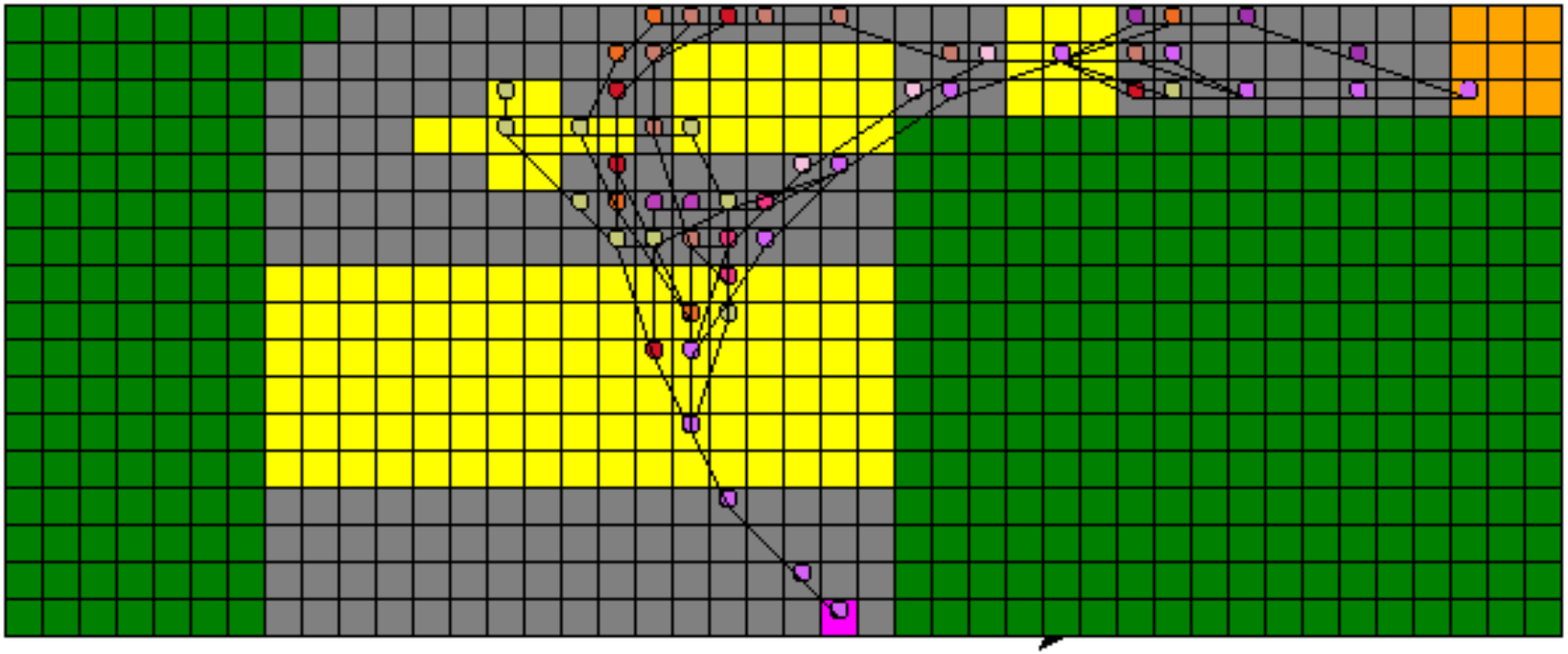
- Fajnie jest dojechać na metę. (+100)
- Ale jeszcze fajniej nie dać się zabić. (-100?)

Uwaga

Pamiętamy, że monotoniczna zmiana funkcji wypłaty:

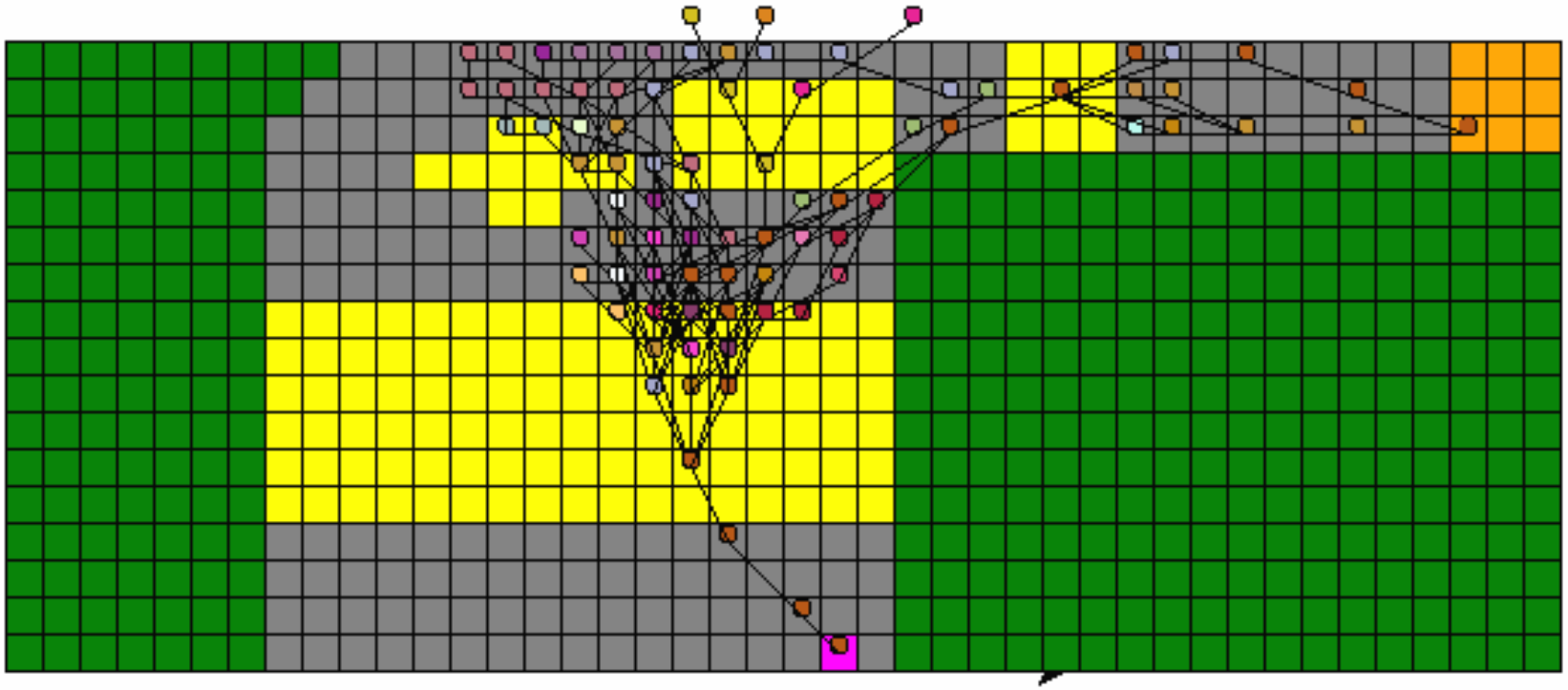
1. nie zmienia wartości MiniMax-owej gry,
2. może zmienić ExpectMinMax

Rozwiązanie podstawowe

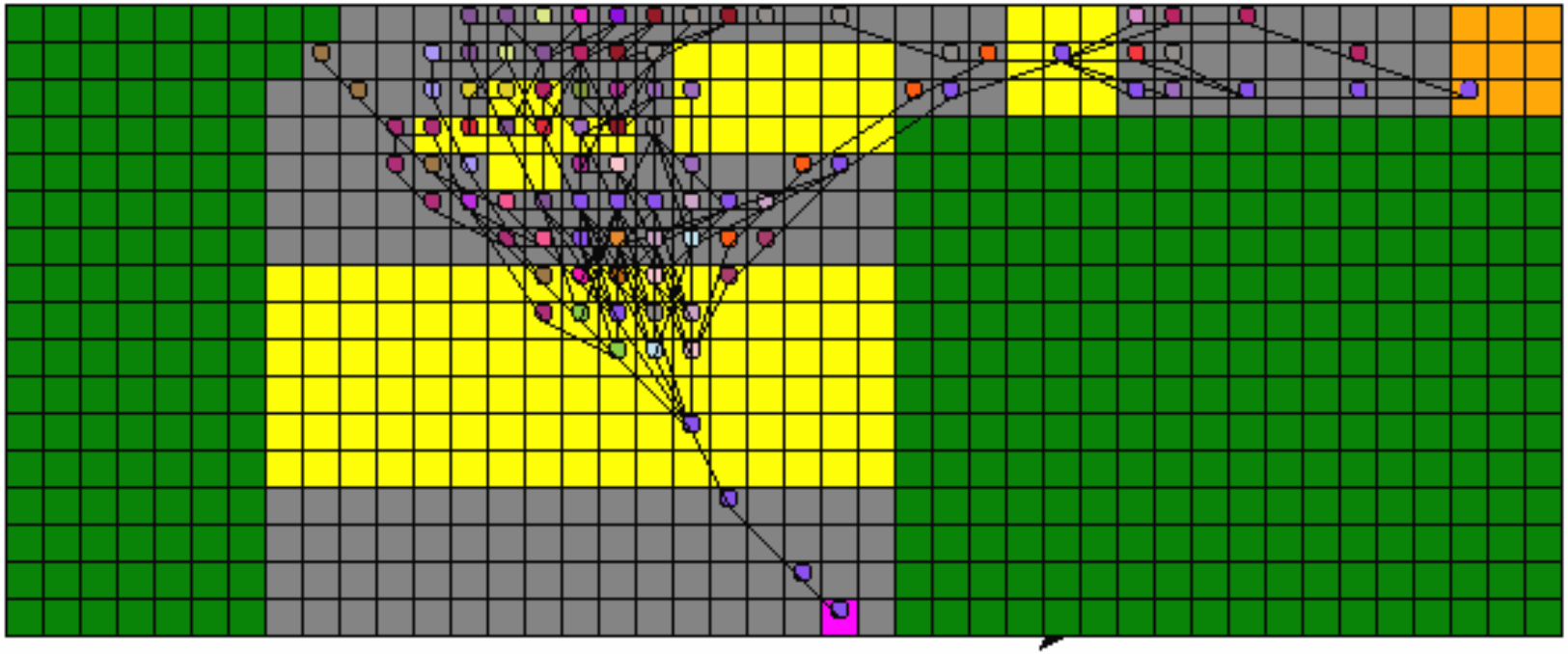


Pytanie: Czego spodziewamy się, jeżeli zamienimy karę na wypadek na 10000?

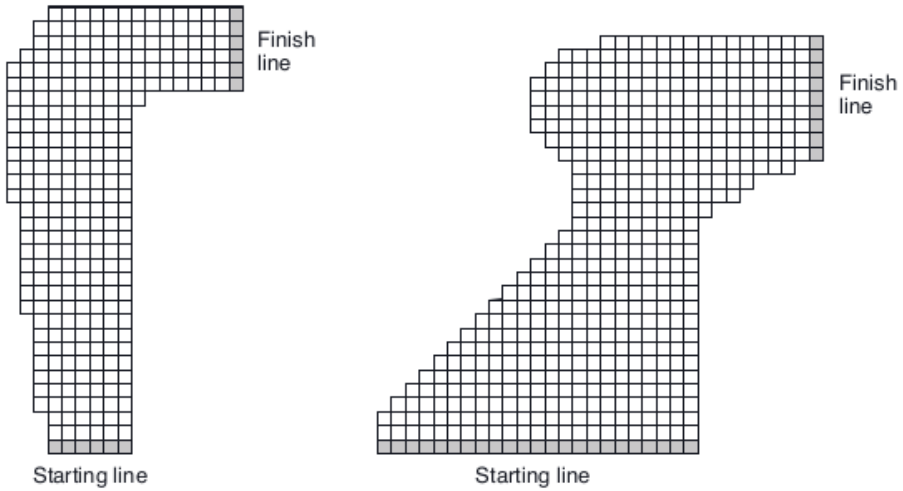
Kara=100



Kara=10000



Przykład. Wyścigi samochodzików. Float



- Prędkość autka jest wektorem $(v\cos(d), v\sin(d))$,
- Możemy zmieniać d (skręcać), oraz v (przyśpieszać, hamować)
- Celem jest meta.
- W pełni deterministyczny świat, ale bardzo duża liczba stanów, zawierających liczby `float`)

Autka float. Rozwiązanie

- Możemu stworzyć **stan abstrakcyjny** i opisać mechanikę świata dla takich stanów
- Oczywiście będzie ona niedeterministyczna, bo nigdy nie będziemy wiedzieć, czy zmiana w świecie float-ów przenosi się na zmianę w świecie int-ów.

Uwaga

Możemy myśleć o tym, że modelujemy błędy pomiarowe (int zamiast float) za pomocą losowości.

Bezmodelowe uczenie ze wzmocnieniem

Przypomnienie

Mówiliśmy o metodach Monte Carlo, w których przeprowadzamy eksperymenty (losowe przebiegi), żeby estymować (nieznane) parametry MDP.

- Nowy cel: od razu liczyć $Q(s, a)$, nie przejmując się tworzeniem modelu.
- Zaczniemy od obliczenia $Q_\pi(s, a)$

Bezmodelowe Monte Carlo

Definicja

$Q_\pi(s, a)$ to oczekiwana sumaryczna nagroda, jaką otrzymamy wykonując w stanie s akcję a, a następnie postępując zgodnie z polityką π

- Użyteczność (dla konkretnego przebiegu): $u_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots$
- $\hat{Q}_\pi(s, a) =$ średnie u_t, gdzie $s_{t-1} = s$, $a_t = a$

Bezmodelowe Monte Carlo – inne sformułowanie

- Zamiast liczyć średnią z całości, można myśleć o uaktualnianiu średniej wraz z pojawieniem się kolejnej informacji.
- Niech: $\eta = \frac{1}{1+\mathrm{cnt}(s,a)}$
- $\hat{Q}_\pi(s,a) \leftarrow (1-\eta)\hat{Q}_\pi(s,a) + \eta u$ (gdzie u jest użytecznością zaobserwowaną w konkretnym przebiegu)

Sprawdźmy, czy to się zgadza.

Bezmodelowe Monte Carlo – inne sformułowanie (2)

$$\hat{Q}_\pi(s,a) \leftarrow (1-\eta)\hat{Q}_\pi(s,a) + \eta u$$

- u jest zaobserwowaną użytecznością
- $\hat{Q}_\pi(s,a)$ jest naszą predykcją.

Reguła ta minimalizuje odległość między predykcją a obserwacją.

Bootstraping

Uwaga

W informatyce często, rozwiązując jakieś zadanie, korzystamy z niedoskonałego (tymczasowego) rozwiązania, żeby rozwiązać zadanie lepiej.

Przykład

Szukanie dobrych i złych słów (analizujemy wpisy na jakimś forum), na początku znamy kilka przykładowych dobrych i złych słów.

Będziemy używać Q (poprzedniej wartości) do obliczenia nowego Q

Bootstraping: SARSA

Obserwujemy ciąg akcji i nagród:

$$s_0, a_1, r_1, s_1, a_2, r_2, s_2, \ldots$$

- Uaktualnianie Monte Carlo:

$$\hat{Q}_\pi(s,a) \leftarrow (1-\eta)\hat{Q}_\pi(s,a) + \eta u$$

- SARSA (obserwujemy s, a, r, s', a'):

$$\hat{Q}_\pi(s,a) \leftarrow (1-\eta)\hat{Q}_\pi(s,a) + \eta(r + \gamma\hat{Q}_\pi(s',a'))$$

W algorytmie SARSA zamiast konkretnego (zaobserwowanego) u bierzemy zaobserwowaną jego i pierwszą część (r) i estymowaną resztę (zielony jest cel)

Uwaga

Nie musimy czekać do końca epizodu, żeby uaktualnić wartość Q!

SARSA(λ)

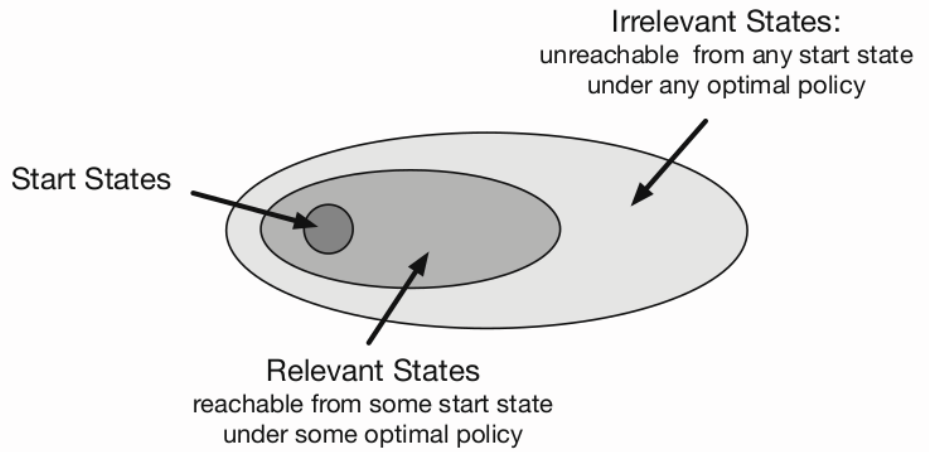
- Można balansować między SARSA a MC
- W skrócie:
  - MC uwzględnia **wszystkie** nagrody (od teraz do końca świata)
  - SARSA uwzględnia **jedną** nagrodę
- A jakbyśmy chcieli uwzględnić 2 nagrody?
- Wówczas cel byłby równy:

$$r_t + \gamma r_{t+1} + \gamma^2 \hat{Q}_\pi(s_{t+1}, a_{t+2})$$

Odpowiednik SARSA dla wartości

- Jak nie znamy modelu, to niespecjalnie użyteczne (same V nie wystarczają do wyboru akcji, jak nie wiemy, co ta akcja spowoduje)
- Ale może być użyteczne w grach (znana mechanika, duża przestrzeń)

Value iteration vs. SARSA i spółka



źródło: Sutton, Reinforcement Larning. An introduction

- VI liczy wartości dla stanów „nieoptymalnych”
- VI liczy wartości dla stanów nieosiągalnych (łatwo wymyślić dla autek taką kombinację prędkości i położenia, która jest bezużyteczna)

W momencie, gdy operujemy przebiegami, być może sensownymi, to koncentrujemy się na estymacji rzeczy użytecznych (a na pewno na osiągalnych!)

Q-learning

Uwaga

SARSA estymuje $Q_\pi(s, a)$. Najbardziej naturalnym celem jest znajomość Q_{opt}.

- Algorytm umożliwiający bezpośrednie obliczanie Q_{opt} to właśnie **Q-learning**.
- Również radzimy sobie bez modelu.
- Mamy do spamiętania trochę więcej wartości: dla każdej akcji i każdego stanu.

Q-learning

Standardowy kształt reguły:

$$Q(s,a) \leftarrow (1-\eta)Q(s,a) + \eta \text{ cel}$$

Celem jest $r + \gamma V_{\text{opt}}(s')$
Natomiast:

$$V_{\text{opt}}(s') = \max_{a' \in \text{Actions(s')}} Q_{\text{opt}}(s', a'))$$

Algorytm **Q-learning**

Dla zaobserwowanych s, a, r, s':

$$Q(s,a) \leftarrow (1-\eta)Q(s,a) + \eta(r + \gamma \max_{a' \in \text{Actions(s')}} Q_{\text{opt}}(s', a')))$$

O wadze eksploracji

- Jeżeli chcemy zachowywać się optymalnie powinniśmy wiedzieć coś o każdej parze (s,a)
  - (wyobrażmy sobie katapultę w autkach)
- Istnieją dwie możliwości:
  1. Rzeczywiście mamy szansę (w granicy) wygenerować przebieg z każdą parą (s,a)
  2. Umiemy jakoś generalizować i wywnioskować coś na temat (s,a) korzystając z podobnego (s', a')

Slajd 33

Do przetestowania

- Pełna eksploatacja: używamy akcji o najwyższym $\hat{Q}_{\text{opt}}(s, a)$
- Pełna eksploracja: używamy polityki losowej

$$\pi(s) = \text{losowo wybrana } \mathit{Actions}(s)$$

- Próbu balansu

Inne spojrzenie na generalizację

Uwaga

Jednym z najprostszych algorytmów uczenia jest **k-NN** (czyli k najbliższych sąsiadów.

Algorytm

1. Mamy zdefiniowaną odległość pomiędzy obiektami (stanami)
2. Spamiętujemy wiele przykładów zawierających pary stan-akcja.
3. Dla nowego stanu znajdujemy **K** najbliższych mu stanów, i pozwalamy, by **zagłosowały** nad akcją.

k-NN a jazda samochodem

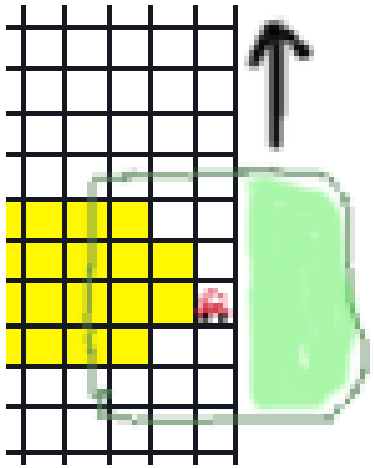
Uwaga

Algorytmy rozważały sytuację konkretnego toru, czyli kierowcy rajdowego, który korzysta z treningów w miejscu wyścigu.

A jakbyśmy chcieli dać prawo jazdy?

Spróbujmy rozważyć sposoby uogólnienia naszych autek.

Uogólnienie jazdy samochodem



Pomysł 1

Bierzemy mały fragment mapy (3×3, 5×5), z samochodem pośrodku. Ten fragment + prędkość to jest stan. **Pytanie:** Jaki problem?

Nie wiadomo, w którą stronę jechać!

Można dodać informację o odległości w grafie sąsiedztwa kratek między celem a danym punktem (wyjaśnienie na tablicy)

Znaki drogowe

Inne cechy:

- Kąt między osią auta a osią drogi [rysunek]
- Odległości pomiędzy autem i lewym i prawym poboczem
- Informacje o tym, co robi droga:
  - 200m prosto, potem skręt o 30 stopni w prawo.
  - Przy nim ograniczenie prędkości do 70.