Zaawansowane Metody Inteligencji Obliczeniowej

Lab 1: Wprowadzenie

Michał Kempka Marek Wydmuch

4 marca 2021

Fundusze Europejskie Polska Cyfrowa | Rzeczpospolita Polska | Unia Europejska Europejski Fundusz Rozwoju Regionalnego

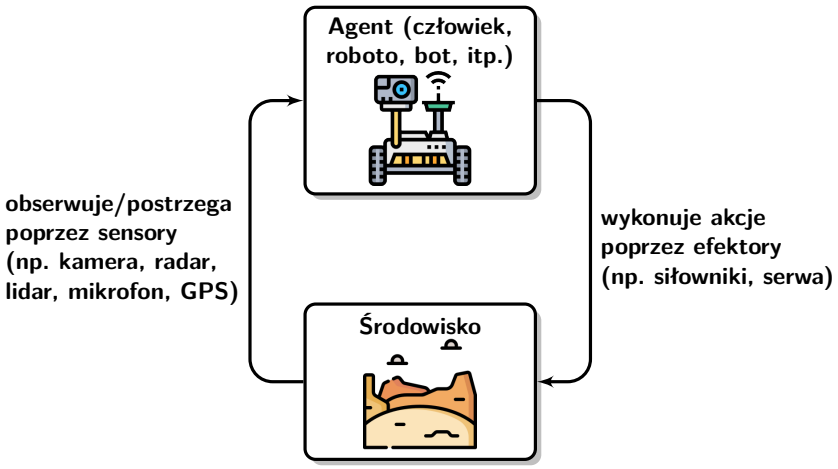
"Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)",
projekt finansowany ze środków Programu Operacyjnego Polska Cyfrowa POPC.03.02.00-00-0001/20

1 Wprowadzenie - agent i środowisko

Każde laboratoria będziemy rozpoczynać krótkim wprowadzeniem. Aczkolwiek generalnie wychodzimy z założenia, że studenci byli obecni na wykładach, dlatego o ile laboratoria nie będą dotyczyć zagadnień z poza wykładu, wprowadzenie będzie tylko krótkim przypomnieniem najważniejszych dla rozwiązania zadań zagadnień.

1.1 Agent i środowisko

Agent (od łac. agere, działać/czynić) to ktoś lub coś (np. człowiek, robot, bot, program itp.) co działa – **wykonuje akcje**. Przede wszystkim mamy tutaj na myśli działanie autonomiczne, działanie przez dłuższy czas przez który agent obserwuje swoje otoczenie (**środowisko**) za pomocą dostępnych sensorów i wykonuje adekwatne do zachodzących w nim zmian akcje za pomocą dostępnych efektorów by realizować dane cele. Rysunek 1 prezentuje ten ogólny model, którym będziemy zajmować się podczas większości zajęć.



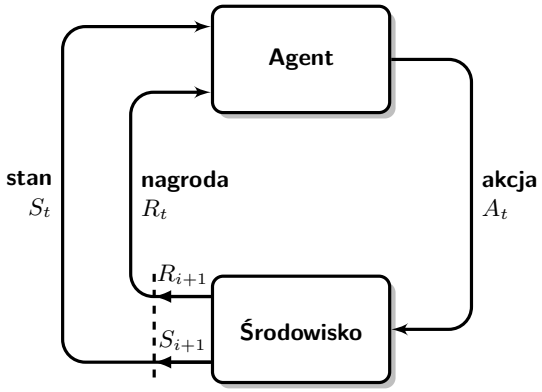
Rysunek 1: Schemat modelu agenta działającego w środowisku.

Funkcja agenta mapuje historię obserwacji na akcje. Funkcję agenta można stabularyzować na zasadzie zdefiniowania zasady: jeśli $\{S_1, S_2 \dots S_t\} = \mathcal{H}$ to wykonaj akcję a dla wszystkich możliwych $\mathcal{H}$. W praktyce nie jest to możliwe i funkcja agenta to abstrakcyjny matematyczny opis agenta, podczas gdy jego właściwą implementację nazywamy **programem agenta**.

1.2 Racjonalność i uczenie ze wzmocnieniem

Podczas zajęć będzie się skupiać na zagadnieniu budowania inteligentnych agentów. Przyjmujemy tutaj pogląd, że inteligencja jest równoznaczna z racjonalnością/racjonalnym działaniem – czyli takim który ma na celu osiągnięcie jak najlepszego rezultatu lub w wypadku niepewności najlepszego oczekiwanego rezultatu.

Przez większość zajęć będziemy się skupiać na modelu **uczenia ze wzmocnieniem (ang. reinforcement learning (RL))** zaprezentowanym na Rysunku 2, gdzie w każdym kroku agent wykonuje **akcję (ang. action)**, obserwuje nowy **stan (ang. state)** środowiska i otrzymuje **nagrodę (ang. reward)**, która formalizuje cel agenta.



Rysunek 2: Schemat modelu uczenia ze wzmocnieniem.

W tym kontekście racjonalny agent **zawsze** wybiera akcję, która maksymalizuje oczekiwaną wartość sumy wszystkich otrzymanych nagród, biorąc pod uwagę aktualną **wiedzę** agenta.

- **Zawsze** – dla każdego możliwego stanu.
- **Wiedza** – a priori (np. model środowiska) + historia wszystkich dotychczas zaobserwowanych stanów.

Suma wszystkich otrzymanych nagród jest nazywana **całkowitą nagrodą (ang. total reward)** lub w bardziej ogólnie ujęciu **miarą jakości**.

1.3 Typy środowisk

Będziemy rozważać różne typy środowisk, kategoryzowane według następujących cech/kryteriów:

1.3.1 Obserwowalność

- **Całkowicie obserwowalne** – stan całego środowiska znany w każdym momencie (np. szachy), w konsekwencji agent nie musi pamiętać poprzednio zaobserwowanych stanów.
- **Częściowo obserwowalne** – stan zawierający tylko część informacji lub zaszumione/niedokładne informacje (np. poker).
- **Nieobserwowalne** – całkowity brak informacji, agent otrzymuje wyłącznie nagrodę (np. jednoręcy bandyci).

1.3.2 Liczność agentów

- **Jednoagentowe** – np. gra w *Breakout*.
- **Wieloagentowe** – mogą być kompetytywne jak i kooperacyjne lub mieszane (np. ruch drogowy).

1.3.3 Determinizm

- **Deterministyczne** – dana akcja w danym stanie będzie skutkować obserwacją zawsze tego samego, następnego stanu (np. balansowanie) .
- **Stochastyczne** – dana akcja w danym stanie może skutkować różnymi stanami i/lub nagrodami, zgodnie ze znanym lub nieznanym rozkładem prawdopodobieństwa (np. gra w ruletkę).

1.3.4 Statyczne i dynamiczne

- **Statyczne** – środowisko 'czeka' na akcję (np. szachy, gry turowe).
- **Dynamiczne** – środowisko zmienia się podczas gdy agent decyduje o kolejnej akcji (np. ruch drogowy, regulowanie temperatury).
- **Semidynamiczne** – środowisko się nie zmienia, ale czas podejmowania decyzji wpływa na otrzymaną nagrodę (np. teleturniej).

1.3.5 Ciągłość

- **Dyskretne** – akcje i stany są skończonymi zbiorami np. szachy. Warto zauważyć, że nagrodę uznaje się zwykle za niedyskretną i pomija w rozważaniach ciągłości środowisk.
- **Ciągłe** – akcje i/lub stany są zmiennymi ciągłymi (liczbami rzeczywiste) np. ruch drogowy.

Wiele środowisk łączy w sobie elementy ciągłe i dyskretne. Przykładem takiego środowiska może być jazda samochodem (np. stan diod, zmiana biegu), a część ciągła (np. prędkościomierz, prędkość skręcania kierownicą)

1.3.6 Model środowiska

Nie jest to cecha środowiska, lecz stan wiedzy agenta (lub projektanta algorytmu), lecz z praktycznego punktu widzenia możemy tę wiedzę uznać za cechę środowiska.

- **Znany model środowiska** – konsekwencje akcji lub ich rozkłady prawdopodobieństw (gdy środowisko jest stochastyczne) jest znany.
- **Nieznany model środowiska** – konsekwencje akcji nie są znane.

1.4 Rodzaje agentów

Analogicznie będziemy również rozważać różne rodzaje agentów:

- **Agent odruchowy** (ang. reflex agent) – odpowiada bezpośrednio na zaobserwowane stan, ignorując poprzednio zaobserwowane stany.
- **Agent z modelem/pamięcią** (ang. model-based agent) – utrzymuje wewnętrzny stan aktualizowany po każdej nowej obserwacji i podejmuje decyzje na jego podstawie. Inaczej mówiąc śledzi informacje na temat aspektów środowiska, które nie są zawsze obserwowalne.
- **Agent celowy** (ang. goal-based agent) – agent, który działa by osiągnąć jakiś cel. Zazwyczaj cel jest dyskretny (konkretny stan), a agent wykonuje jakiegoś rodzaju przeszukiwanie lub planowanie, w tym celu wymaga by model środowiska był znany.
- **Agent z funkcją jakości** (ang. utility-based agent) – agent, który działa by zmaksymalizować zadaną miarę jakości.

2 Zadania

2.1 Cechy środowisk

Określ cechy dla poniższych środowisk (niektóre z nich mogą być dyskusyjne, weź pod uwagę, że wiele środowisk w prawdziwym świecie może nie być technicznie stochastyczna ani częściowo obserwowalna, ale mogę być traktowane jako takie z powodu ich złożoności):

1. Eksploracja powierzchni Marsa w celu znalezienia śladów życia
2. Gra w piłkę nożną
3. Gra w bilard
4. Wykrywanie spamu

2.2 Racjonalność

Odpowiedź na poniższe pytania, odpowiedź uzasadnij:

1. Czy agent, który otrzymuje informację jedynie o części stanu środowiska może być w pełni racjonalny?
2. Czy jest możliwe by ten sam agent był w pełni racjonalny w dwóch różnych od siebie środowiskach?
3. Agent wybiera swoje akcje losowo, czy istnieje środowisko w którym jest racjonalny?
4. Czy perfekcyjnie racjonalny agent grający w pokera nigdy nie przegra?

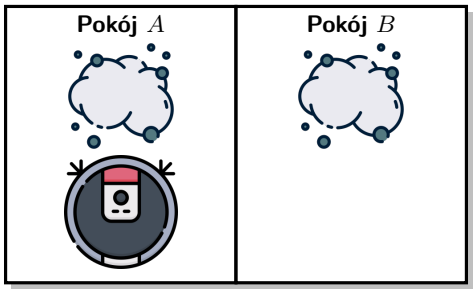
2.3 Horyzont czasowy a racjonalność

Rozważmy środowisko w którym, że miara jakości jest liczona tylko dla pierwszych T kroków, nagrody otrzymane po czasie T są ignorowane. Pokaż, że działanie racjonalnego agenta może zależeć nie tylko od stanu środowiska ale także od czasu, który upłynął.

2.4 Robot odkurzacz

Rozważmy proste środowisko odkurzacza robota, którego schemat widoczny jest na Rysunku 3:

- Środowisko składa się z dwóch pokoi A i B, w każdym pokoju może być kurz.
- Agent (robot odkurzacz) znajduje się w jednym z pokoi, otrzymuje stan mówiący mu o tym, w którym pokoju się znajduje (pozycja) i czy jest w nim brudno (zabrudzenie), np. (A, *brudno*) albo (B, *czysto*). Może się on przemieścić do pokoju w *lewe* lub *prawo*, *sprzątać* kurz lub nie robić *nic*.
- Środowisko kończy się po 100 krokach.
- Agent jest nagradzany 10 punktami za każdy czysty pokoju na koniec (po 100 krokach).
- Posprzątany pokój pozostaje do końca czysty.
- Początkowa lokalizacja agenta oraz obecność kurzu w pokojach A i B nie są znane a priori.



Rysunek 3: Świat odkurzacza robota z dwoma pokojami

1. Pokaż, że następująca funkcja agenta odruchowego jest racjonalna:
   - Jeśli $S_t = (A,$ *brudno*$)$ lub $S_t = (B,$ *brudno*$)$, to *sprzątać*.
   - Jeśli $S_t = (A,$ *czysto*$)$, to *prawo*.
   - Jeśli $S_t = (B,$ *czysto*$)$, to *lewo*.
2. Rozważmy zmodyfikowaną wersję środowiska, w której każdy ruch do sąsiedniego pokoju kosztuje agenta 1 punkt.
   (a) Czy agent opisany w punkcie 1 jest nadal racjonalny?
   (b) Co w wypadku agenta z pamięcią? Jaka powinna być funkcja takiego agenta?
   (c) Co jeśli agent będzie w stanie obserwować obecność kurzu w obu pokojach, czy wtedy agent odruchowy może być racjonalny?
3. Rozważmy niedeterministyczną (stochastyczną) wersję środowiska z Zadania 4.4.2 w której robot odkurzacz jest dodatkowo wadliwy:
   - w 50% przypadków akcja *sprzątać* się nie udaje i pozostawia kurz w pokoju nawet jeśli ten był czysty,

- w 25% przypadków czujniki kurzu w pokoju podaje nieprawidłową informację.

Co powinien uwzględniać racjonalny agent w tym wypadku?

4. Rozważmy niedeterministyczną wersję środowiska z Zadania 4.4.2, w której w oprócz robota odkurzacza w pokojach znajdują się koty z kłaczącym futerkiem:
   - W każdym kroku, pokój ma 10% szansy by znowu znowu się ubrudzić.
   - Agent jest nagradzany 2 punktami za każdym razem kiedy wyczyści brudny pokój.

   Co powinien uwzględniać racjonalny agent w tym wypadku?

Problem 4.4.3 jak i 4.4.4 jest (częściowo obserwowalnym) procesem decyzyjnym Markowa (ang. POMDP), o którym będziemy się dokładnie zajmować niedługo. Takie problemy są w ogólności trudne, ale niektóre szczególne przypadki jak ten mogą być poddane dokładnej analizie i rozwiązane optymalnie.

2.5 Zadanie dodatkowe

Zadanie do poeksperymentowania w domu, nie będzie ocenianie, ale doświadczanie z pracy nad nim przyda się w przyszłości.

Zaimplementuj symulację środowiska robota odkurzacza omawianego w Zadaniu 4.4. Zastanów się, jak zaprojektować symulację tak by była najbardziej modularna, tak by charakterystyki środowiska można było jak najłatwiej zmieniać, a z symulacji mogły korzystać różne programy agentów. Zaimplementuj odruchowego agenta i przetestuj go w różnych wariantach środowiska.

Alternatywnie przejrzyj się już gotowej implementacji klasy środowiska odkurzacza i agentów. Przykład jego zastosowania możecie znaleźć w tym notebooku.

Fundusze Europejskie Polska Cyfrowa | Rzeczpospolita Polska | Unia Europejska Europejski Fundusz Rozwoju Regionalnego

"Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)",
projekt finansowany ze środków Programu Operacyjnego Polska Cyfrowa POPC.03.02.00-00-0001/20