Wprowadzenie do Sztucznej Inteligencji (WSI)

Paweł Wawrzyński

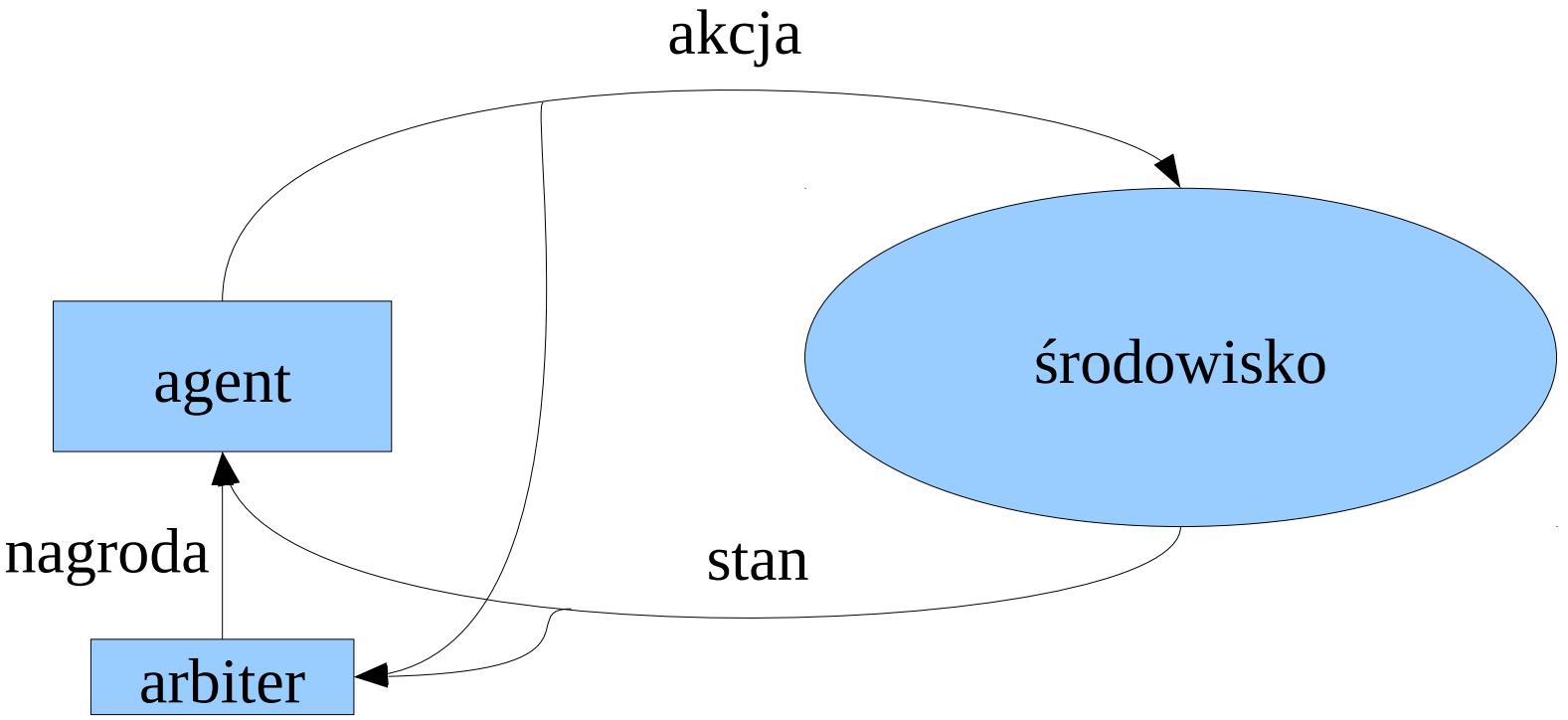
Uczenie maszynowe

Uczenie się ze wzmocnieniem

Plan na dziś

- Uczenie się ze wzmocnieniem
- Proces decyzyjny Markowa
- Programowanie dynamiczne
- Q-Learning

Sekwencyjne decyzje w warunkach niepewności



Proces Decyzyjny Markowa (PDM)

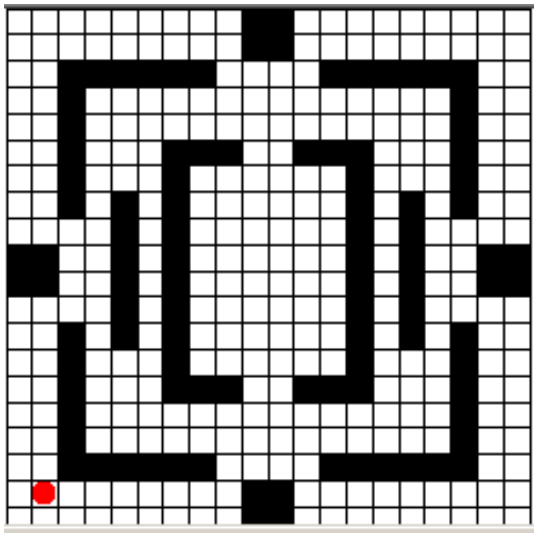
- agent + środowisko
- dyskretny czas $t = 1, 2, \ldots$
- stan x_t
- akcja u_t
- następny stan x_{t+1}
- nagroda/wzmocnienie r_t
- epizody
- cel: maksymalizacja przyszłych nagród

PDM formalnie

- przestrzeń stanów $x \in \mathcal{X}$
- przestrzeń akcji $u \in \mathcal{U}$
- rozkład przejścia stanów $P_x(x_{t+1}|x_t, u_t)$
- nagrody $r_t = r(u_t, x_{t+1})$
- stany terminalne epizodów $\mathcal{X}^*$
- rozkład stanów początkowych epizodu P_0
- środowisko: $P_x,\ r,\ \mathcal{X}^*$ i P_0

Przykład: problem dyskretny

- skończona przestrzeń stanów $x \in \mathcal{X}$
- skończona przestrzeń akcji $u \in \mathcal{U}$
- środowisko $P_x,\ r,\ \mathcal{X}^*$ i P_0
- cel: maksymalizacja sumy przyszłych nagród
- ilustracja: labirynt
  - nagroda -1 za każdą chwilę pobytu w labiryncie



Przykład: problem ciągły

- Sterowanie urządzeniem o nieznanej dynamice
  - stan środowiska: stan urządzenia + stan docelowy
  - akcje: sterowanie
  - nagroda: minus odchyłka

Przykład: problem dyskretno-ciągły

- Bot w grze komputerowej
  - stan środowiska: orientacja bota w świecie gry + obserwowana przez niego sytuacja w grze
  - akcje: sterowanie
  - nagroda: przeżycie, zbieranie fantów, wyrządzanie jak najbardziej perfidnej szkody ludzkiemu graczowi...

Ogólne podejścia

- Znany model środowiska
  - strategia deliberatywna, np. MIN-MAX
  - strategia reaktywna << programowanie dynamiczne
- Nieznany model środowiska
  - skonstruować model środowiska i jw.
  - *nauczyć się* strategii reaktywnej
    przy użyciu „prób i błędów”
    czyli: uczenie się ze wzmocnieniem

Polityka i cel

- polityka π → na podstawie stanów
  → akcje lub ich prawdopodobieństwa

- funkcja wartości $V^{\pi}(x) = E\left(\sum_{i=0}^{\infty} \gamma^i r_{t+i} \middle| x_t = x; \pi\right)$

- dyskonto: $\gamma \in (0, 1)$
- cel: taka polityka, która maksymalizuje funkcję wartości dla każdego stanu

Narzędzia analizy

- stany $x \in \mathcal{X}$ akcje $u \in \mathcal{U}$
- środowisko $P_x,\ r,\ \mathcal{X}^*$ i P_0 polityka π
- Funkcja wartości

$$V^{\pi}(x) = E\left(\sum_{i=0}^{\infty} \gamma^i r_{t+i} \middle| x_t = x; \pi\right)$$

- Funkcja wartości-akcji

$$Q^{\pi}(x,u) = E\left(\sum_{i=0}^{\infty} \gamma^i r_{t+i} \middle| x_t = x, u_t = u; \pi\right)$$

Ilustracja: labirynt

$$V^{\pi}(x) = E\left(\sum_{i=0}^{\infty} \gamma^i r_{t+i} \middle| x_t = x; \pi\right)$$

$$Q^{\pi}(x,u) = E\left(\sum_{i=0}^{\infty} \gamma^i r_{t+i} \middle| x_t = x, u_t = u; \pi\right)$$

nagroda = -1, dyskonto = 1

Polityka

Wy	<	<	<
>	>	>	^
>	>	>	^

V, Q

	-2 -1 **-1** -3 -7	-3 -2 **-2** -4 -6	-4 -3 **-3** -4 -5
-1 -8 **-7** -7 -9	-2 -8 **-6** -6 -8	-3 -7 **-5** -5 -7	-4 -6 **-4** -5 -6
-8 -9 **-8** -8 -10	-7 -9 **-7** -7 -9	-6 -8 **-6** -6 -8	-5 -7 **-5** -6 -7

Polityka wyindukowana

Wy	<	<	<
^	^	^	^
^	^	^	^

Narzędzia analizy 2

$$V^{\pi}(x) = E\left(\sum_{i=0}^{\infty} \gamma^i r_{t+i} \middle| x_t = x; \pi\right)$$

$$Q^{\pi}(x,u) = E\left(\sum_{i=0}^{\infty} \gamma^i r_{t+i} \middle| x_t = x, u_t = u; \pi\right)$$

- Istotne własności

$$V^{\pi}(x) = E\left(Q^{\pi}(x_t, u_t) \middle| x_t = x; \pi\right)$$

$$\begin{aligned} Q^{\pi}(x,u) &= E\left(r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots \middle| x_t = x, u_t = u; \pi\right) \\ &= E\left(r_t + \gamma(r_{t+1} + \gamma r_{t+2} + \dots) \middle| x_t = x, u_t = u; \pi\right) \\ &= E\left(r_t + \gamma V^{\pi}(x_{t+1}) \middle| x_t = x, u_t = u\right) \end{aligned}$$

Narzędzia analizy 3

- Funkcja $Q : \mathcal{X} \times \mathcal{U} \mapsto \Re$
  indukuje politykę π jeśli

$$\arg\max_u Q(x_t, u) \rightarrow u_t$$

- Polityka π' indukowana przez Q^π
  jest niegorsza niż π

$$(\forall x \in \mathcal{X}) V^{\pi'}(x) \geq V^\pi(x)$$

- Jeśli π nie jest optymalna to

$$(\exists x \in \mathcal{X}) V^{\pi'}(x) > V^\pi(x)$$

Iteracja polityki

- Optymalna polityka, funkcja wartości i funkcja wartości-akcji

$$\pi^* \qquad V^* \qquad Q^*$$

- Q^* indukuje π^*

Programowanie dynamiczne

- Mamy model środowiska
- Możemy użyć programowania dynamicznego

- Algorytm *Iteracji polityki*:

0. Określ dowolną politykę początkowa π.
1. Wyznacz funkcję Q^π dla polityki π.
2. Wyznacz politykę indukowaną przez Q^π i przypisz ją do π.
3. Jeśli w ostatnim kroku polityka zmieniła się, wróć do 1.

Idea algorytmu Q-Learning

- Utrzymywana jest funkcja Q która ma zbiegać do Q^*
- Krok algorytmu = krok agenta w środowisku
- W każdym kroku $Q(x_t, u_t)$ jest poprawiana na podstawie zdobytego doświadczenia

Algorytm

0: Zainicjalizuj: $t \leftarrow 1$, Q.
1: Wylosuj u_t na podstawie Q i x_t.
2: Wykonaj u_t, zarejestruj x_{t+1} i r_t.
3: Przypisz

$$Q(x_t, u_t) \leftarrow Q(x_t, u_t) + \beta_t \left(r_t + \gamma \max_u Q(x_{t+1}, u) - Q(x_t, u_t) \right).$$

przy czym $Q(x_{t+1}, \cdot) \equiv 0$, jeśli po akcji u_t skończył się epizod.
4: Przypisz $t \leftarrow t + 1$ i wróć do Punktu 1.

Punkt 1 algorytmu

0: Zainicjalizuj: $t \leftarrow 1$, Q.
1: Wylosuj u_t na podstawie Q i x_t.
2: Wykonaj u_t, zarejestruj x_{t+1} i r_t.
3: Przypisz

$$Q(x_t, u_t) \leftarrow Q(x_t, u_t) + \beta_t \left(r_t + \gamma \max_u Q(x_{t+1}, u) - Q(x_t, u_t) \right).$$

przy czym $Q(x_{t+1}, \cdot) \equiv 0$, jeśli po akcji u_t skończył się epizod.
4: Przypisz $t \leftarrow t + 1$ i wróć do Punktu 1.

- Strategia ϵ- zachłanna

$$P(u_t = u|x_t) = \epsilon \frac{1}{|\mathcal{U}|} + (1 - \epsilon) \frac{[u \in \arg\max_u Q(x_t, u)]}{|\arg\max_u Q(x_t, u)|}$$

- Strategia Bolzmannowska

$$P(u_t = u|x_t) = \frac{\exp(Q(x_t, u)/T)}{\sum_{u' \in \mathcal{U}} \exp(Q(x_t, u')/T)}$$

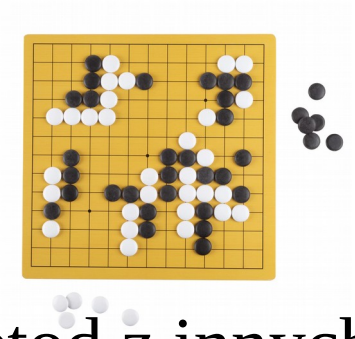
- Eksploracja vs. eksploatacja

Warunki zbieżności z prawdopodobieństwem 1

- Skończona przestrzeń stanów
- Skończona przestrzeń akcji
- Ograniczone nagrody
- Każda akcja w każdym stanie jest zostaje wykonana nieskończenie wiele razy

Uczenie się ze wzmocnieniem: co jeszcze?

- Algorytmy działające gdy przestrzenie stanów i akcji są ciągłe
- Algorytmy optymalizujący parametryczną politykę decyzyjną: Aktor-Krytyk
- Algorytmy, które nie zakładają, że stan jest obserwowalny
- Zastosowania
  - w grach, np. Go



  - wspomaganie metod z innych obszarów