Problemy Decyzyjne Markowa

na podstawie AIMA ch17 i slajdów S. Russel'a

Wojciech Jaśkowski

Instytut Informatyki,
Politechnika Poznańska

9 maja 2014

2014-05-09

Problemy Decyzyjne Markowa
na podstawie AIMA ch17 i slajdów S. Russel'a
Wojciech Jaśkowski
Instytut Informatyki,
Politechnika Poznańska
9 maja 2014

Sekwencyjne problemy decyzyjne

Sekwencyjny problem decyzyjny

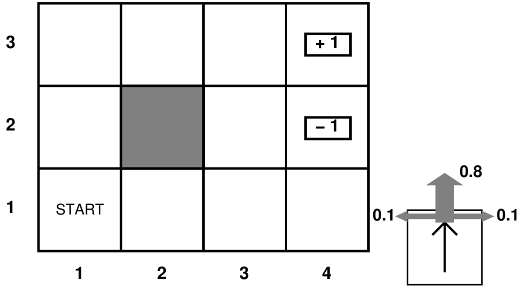
Cechy:

- ocena (użyteczność) agenta zależy od **sekwencji decyzji**, a nie od pojedynczej decyzji.
- **niepewność** (obserwacje, akcje)

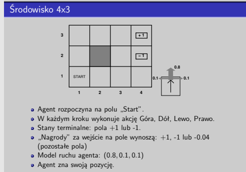
Problemy planowania i przeszukiwania — szczególny przypadek sekwencyjnych problemów decyzyjnych (bez niepewności)

Sekwencyjne problemy decyzyjne
Sekwencyjny problem decyzyjny
Cechy:
ocena (użyteczność) agenta zależy od sekwencji decyzji, a nie od pojedynczej decyzji.
niepewność (obserwacje, akcje)
Problemy planowania i przeszukiwania — szczególny przypadek sekwencyjnych problemów decyzyjnych (bez niepewności)

Środowisko 4x3

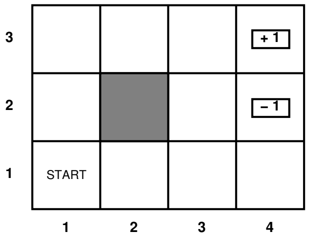


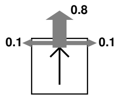
- Agent rozpoczyna na polu „Start”.
- W każdym kroku wykonuje akcję Góra, Dół, Lewo, Prawo.
- Stany terminalne: pola +1 lub -1.
- „Nagrody” za wejście na pole wynoszą: +1, -1 lub -0.04 (pozostałe pola)
- Model ruchu agenta: $(0.8, 0.1, 0.1)$
- Agent zna swoją pozycję.



1. Interakcja ze środowiskiem kończy się, gdy agent dotrze do stanów terminalnych
2. Czyli stan środowiska jest znany.

Cechy środowiska



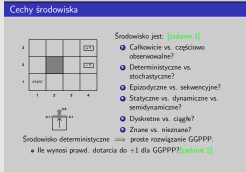


Środowisko jest: [zadanie 1]

1. Całkowicie vs. częściowo obserwowalne?
2. Deterministyczne vs. stochastyczne?
3. Epizodyczne vs. sekwencyjne?
4. Statyczne vs. dynamiczne vs. semidynamiczne?
5. Dyskretne vs. ciągłe?
6. Znane vs. nieznane?

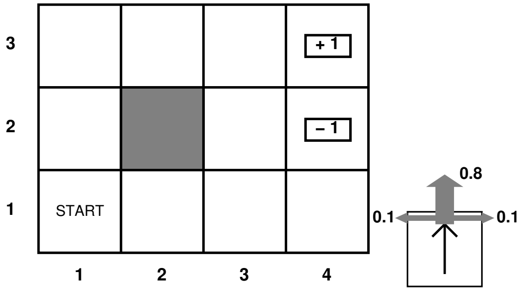
Środowisko deterministyczne $\Longrightarrow$ proste rozwiązanie GGPPP.

- Ile wynosi prawd. dotarcia do $+1$ dla GGPPP?[zadanie 2]


Cechy środowiska
Środowisko jest: [zadanie 1]
Całkowicie vs. częściowo obserwowalne?
Deterministyczne vs. stochastyczne?
Epizodyczne vs. sekwencyjne?
Statyczne vs. dynamiczne vs. semidynamiczne?
Dyskretne vs. ciągłe?
Znane vs. nieznane?
Środowisko deterministyczne ⟹ proste rozwiązanie GGPPP.
Ile wynosi prawd. dotarcia do +1 dla GGPPP?[zadanie 2]

1. Całkowicie
2. Stochastyczne
3. Sekwencyjne
4. Statyczne
5. Dyskretne
6. Znane
7. Prawd. dotarcia do $+1$ wynosi $0.8^5 + 0.1^4 \times 0.8 = 0.32776$ (dla strategii GGPPP).

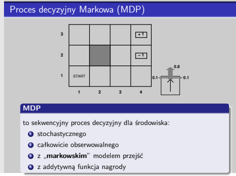
Proces decyzyjny Markowa (MDP)



MDP

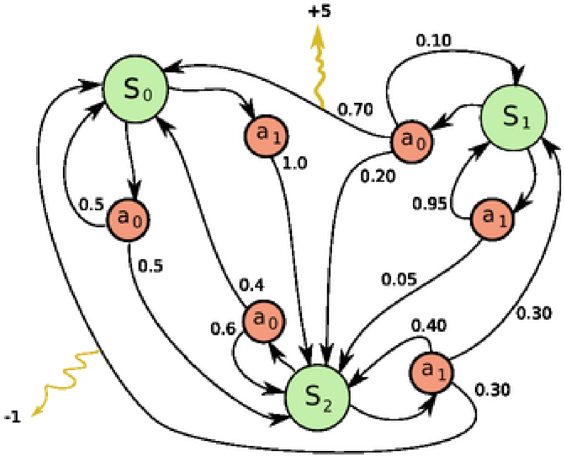
to sekwencyjny proces decyzyjny dla środowiska:

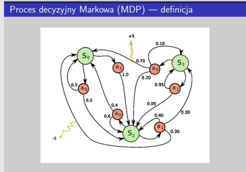
1. stochastycznego
2. całkowicie obserwowalnego
3. z „**markowskim**” modelem przejść
4. z addytywną funkcja nagrody



1. **MDP** = **Markov decision process**. W literaturze anglojęzycznej bardzo często używa się skrótu „MDP”, więc warto go pamiętać.

Proces decyzyjny Markowa (MDP) — definicja





Proces decyzyjny Markowa (MDP) — definicja

Elementy MDP:

- **Stany** $s \in S$, stan początkowy $s_0 \in S$ i stany terminalne
- **Akcje** $a \in A$, zbiór akcji ze stanu s, to $A(s)$.

Proces decyzyjny Markowa (MDP) — definicja
Elementy MDP:
- **Stany** $s \in S$, stan początkowy $s_0 \in S$ i stany terminalne
- **Akcje** $a \in A$, zbiór akcji ze stanu s, to $A(s)$.

1. Nazewnictwo: **model przejść** = **model tranzycji** = **model**
2. **Własność Markowa**: prawd. przejścia ze stanu s do stanu s' zależy tylko od stanu s a nie od historii poprzednich stanów. Przyszłe stany procesu są warunkowo niezależne od stanów przeszłych:
$$P(s^k|a, s^{k-1}) = P(s^k|a, s^{k-1}, s^{k-2}, \ldots, s^0).$$
Dzięki własności Markowa, aby podejmować racjonalne (optymalne) decyzje wystarczy aktualny stan (nie trzeba znać całej historii stanów, *vide* definicja racjonalności agenta).
3. Uogólniona nagroda jest za parę (s, a) czyli wykonanie danej akcji w danym stanie, a nie za samą obecność w danym stanie. A jak interpretować nagrodę $R(s, a, s')$?
4. Świat jest niedeterministyczny, ale całkowicie obserwowalny (częściowo jedynie w przypadku POMDP)
5. (PO)MPD jest rozszerzeniem (ukrytych) łańcuchów Markowa o możliwość decyzji (poprzez akcje) i nagrody (motywacja).

Proces decyzyjny Markowa (MDP) — definicja

Elementy MDP:

- **Stany** $s \in S$, stan początkowy $s_0 \in S$ i stany terminalne
- **Akcje** $a \in A$, zbiór akcji ze stanu s, to $A(s)$.
- **Model (przejść)** $T(s,a,s') \equiv P(s'|s,a)$ — prawd., że akcja a w stanie s prowadzi do stanu s'.
  - **własność Markowa**

Proces decyzyjny Markowa (MDP) — definicja
Elementy MDP:
Stany $s \in S$, stan początkowy $s_0 \in S$ i stany terminalne
Akcje $a \in A$, zbiór akcji ze stanu s, to $A(s)$.
Model (przejść) $T(s,a,s') \equiv P(s'|s,a)$ — prawd., że akcja a w stanie s prowadzi do stanu s'.
własność Markowa

1. Nazewnictwo: **model przejść** = **model tranzycji** = **model**
2. **Własność Markowa**: prawd. przejścia ze stanu s do stanu s' zależy tylko od stanu s a nie od historii poprzednich stanów. Przyszłe stany procesu są warunkowo niezależne od stanów przeszłych:
$$P(s^k|a,s^{k-1}) = P(s^k|a,s^{k-1},s^{k-2},\ldots,s^0).$$
Dzięki własności Markowa, aby podejmować racjonalne (optymalne) decyzje wystarczy aktualny stan (nie trzeba znać całej historii stanów, *vide* definicja racjonalności agenta).
3. Uogólniona nagroda jest za parę (s,a) czyli wykonanie danej akcji w danym stanie, a nie za samą obecność w danym stanie. A jak interpretować nagrodę $R(s,a,s')$?
4. Świat jest niedeterministyczny, ale całkowicie obserwowalny (częściowo jedynie w przypadku POMDP)
5. (PO)MPD jest rozszerzeniem (ukrytych) łańcuchów Markowa o możliwość decyzji (poprzez akcje) i nagrody (motywacja).

Proces decyzyjny Markowa (MDP) — definicja

Elementy MDP:

- **Stany** $s \in S$, stan początkowy $s_0 \in S$ i stany terminalne
- **Akcje** $a \in A$, zbiór akcji ze stanu s, to $A(s)$.
- **Model (przejść)** $T(s, a, s') \equiv P(s'|s, a)$ — prawd., że akcja a w stanie s prowadzi do stanu s'.
  - **własność Markowa**
- **Funkcja nagrody** (ang. reward) $R(s)$
  - np. $R(s) = \begin{cases} -0,04 & \text{dla stanów nieterminalnych (kara)} \\ \pm 1 & \text{dla stanów terminalnych} \end{cases}$
    - Można też uogólnić nagrodę do $R(s, a)$ lub $R(s, a, s')$, ale nie zmienia to podstawowych cech problemu.

Proces decyzyjny Markowa (MDP) — definicja
Elementy MDP:
Stany $s \in S$, stan początkowy $s_0 \in S$ i stany terminalne
Akcje $a \in A$, zbiór akcji ze stanu s, to $A(s)$.
Model (przejść) $T(s, a, s') \equiv P(s'|s, a)$ — prawd., że akcja a w stanie s prowadzi do stanu s'.
własność Markowa
Funkcja nagrody (ang. reward) $R(s)$
np. $R(s) = \begin{cases} -0,04 & \text{dla stanów nieterminalnych (kara)} \\ \pm 1 & \text{dla stanów terminalnych} \end{cases}$
Można też uogólnić nagrodę do $R(s, a)$ lub $R(s, a, s')$, ale nie zmienia to podstawowych cech problemu.

1. Nazewnictwo: **model przejść** = **model tranzycji** = **model**
2. **Własność Markowa**: prawd. przejścia ze stanu s do stanu s' zależy tylko od stanu s a nie od historii poprzednich stanów. Przyszłe stany procesu są warunkowo niezależne od stanów przeszłych:
$$P(s^k|a, s^{k-1}) = P(s^k|a, s^{k-1}, s^{k-2}, \ldots, s^0).$$
Dzięki własności Markowa, aby podejmować racjonalne (optymalne) decyzje wystarczy aktualny stan (nie trzeba znać całej historii stanów, *vide* definicja racjonalności agenta).
3. Uogólniona nagroda jest za parę (s, a) czyli wykonanie danej akcji w danym stanie, a nie za samą obecność w danym stanie. A jak interpretować nagrodę $R(s, a, s')$?
4. Świat jest niedeterministyczny, ale całkowicie obserwowalny (częściowo jedynie w przypadku POMDP)
5. (PO)MPD jest rozszerzeniem (ukrytych) łańcuchów Markowa o możliwość decyzji (poprzez akcje) i nagrody (motywacja).

Proces decyzyjny Markowa (MDP) — definicja

Elementy MDP:

- **Stany** $s \in S$, stan początkowy $s_0 \in S$ i stany terminalne
- **Akcje** $a \in A$, zbiór akcji ze stanu s, to $A(s)$.
- **Model (przejść)** $T(s,a,s') \equiv P(s'|s,a)$ — prawd., że akcja a w stanie s prowadzi do stanu s'.
  - **własność Markowa**
- **Funkcja nagrody** (ang. reward) $R(s)$
  - np. $R(s) = \begin{cases} -0,04 & \text{dla stanów nieterminalnych (kara)} \\ \pm 1 & \text{dla stanów terminalnych} \end{cases}$
    - Można też uogólnić nagrodę do $R(s,a)$ lub $R(s,a,s')$, ale nie zmienia to podstawowych cech problemu.
- **Użyteczność** agenta jest funkcją funkcji nagrody (np. suma nagród w czasie życia).

Proces decyzyjny Markowa (MDP) — definicja
Elementy MDP:
Stany $s \in S$, stan początkowy $s_0 \in S$ i stany terminalne
Akcje $a \in A$, zbiór akcji ze stanu s, to $A(s)$.
Model (przejść) $T(s,a,s') \equiv P(s'|s,a)$ — prawd., że akcja a w stanie s prowadzi do stanu s'.
własność Markowa
Funkcja nagrody (ang. reward) $R(s)$
np. $R(s) = \begin{cases} -0,04 & \text{dla stanów nieterminalnych (kara)} \\ \pm 1 & \text{dla stanów terminalnych} \end{cases}$
Można też uogólnić nagrodę do $R(s,a)$ lub $R(s,a,s')$, ale nie zmienia to podstawowych cech problemu.
Użyteczność agenta jest funkcją funkcji nagrody (np. suma nagród w czasie życia).

1. Nazewnictwo: **model przejść** = **model tranzycji** = **model**
2. **Własność Markowa**: prawd. przejścia ze stanu s do stanu s' zależy tylko od stanu s a nie od historii poprzednich stanów. Przyszłe stany procesu są warunkowo niezależne od stanów przeszłych:
   $$P(s^k|a, s^{k-1}) = P(s^k|a, s^{k-1}, s^{k-2}, \ldots, s^0).$$
   Dzięki własności Markowa, aby podejmować racjonalne (optymalne) decyzje wystarczy aktualny stan (nie trzeba znać całej historii stanów, *vide* definicja racjonalności agenta).
3. Uogólniona nagroda jest za parę (s,a) czyli wykonanie danej akcji w danym stanie, a nie za samą obecność w danym stanie. A jak interpretować nagrodę $R(s,a,s')$?
4. Świat jest niedeterministyczny, ale całkowicie obserwowalny (częściowo jedynie w przypadku POMDP)
5. (PO)MPD jest rozszerzeniem (ukrytych) łańcuchów Markowa o możliwość decyzji (poprzez akcje) i nagrody (motywacja).

Rozwiązywanie MDP

Cel dla:

- problemów przeszukiwania: znalezienie optymalnej sekwencji akcji.
- MDP: znalezienie **optymalnej polityki** $\pi(s)$

Rozwiązywanie MDP
Cel dla:
problemów przeszukiwania: znalezienie optymalnej sekwencji akcji.
MDP: znalezienie optymalnej polityki $\pi(s)$

1. Nie może być stała, ponieważ nie wiadomo, które stany zostaną odwiedzone (środowisko jest niedeterministyczne). Niezależnie gdzie agent się znajdzie, zawsze dokładnie wie co robić.
2. **Polityka** (ang. policy) = **strategia**
3. **optymalna polityka** = **racjonalna polityka** (czyli agent ma być racjonalny!)
4. Polityka *explicite* reprezentuje funkcję agenta i w ten sposób opisuje prostego agenta odruchowego na podstawie obliczonej użyteczności przez agenta z f. użyteczności.
5. Na rysunku: Optymalna polityka, gdy kara $R(s)$ wynosi $-0,04$. Zwróćmy uwagę na konserwatywny wybór (3,1).

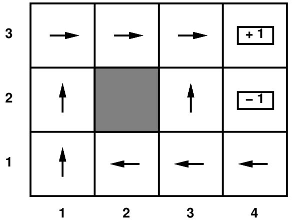
Rozwiązywanie MDP

Cel dla:

- problemów przeszukiwania: znalezienie optymalnej sekwencji akcji.
- MDP: znalezienie **optymalnej polityki** $\pi(s)$

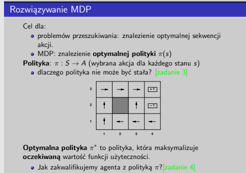
Polityka: $\pi : S \rightarrow A$ (wybrana akcja dla każdego stanu s)

- dlaczego polityka nie może być stała? [zadanie 3]



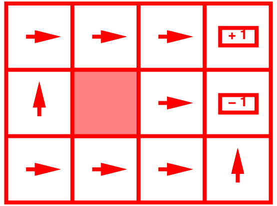
Optymalna polityka π^* to polityka, która maksymalizuje **oczekiwaną** wartość funkcji użyteczności.

- Jak zakwalifikujemy agenta z polityką π?[zadanie 4]

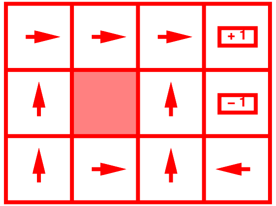


1. Nie może być stała, ponieważ nie wiadomo, które stany zostaną odwiedzone (środowisko jest niedeterministyczne). Niezależnie gdzie agent się znajdzie, zawsze dokładnie wie co robić.
2. **Polityka** (ang. policy) = **strategia**
3. **optymalna polityka** = **racjonalna polityka** (czyli agent ma być racjonalny!)
4. Polityka *explicite* reprezentuje funkcję agenta i w ten sposób opisuje prostego agenta odruchowego na podstawie obliczonej użyteczności przez agenta z f. użyteczności.
5. Na rysunku: Optymalna polityka, gdy kara $R(s)$ wynosi $-0,04$. Zwróćmy uwagę na konserwatywny wybór (3,1).

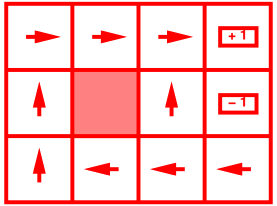
Ryzyko kary vs. nagroda



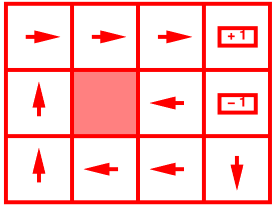
r = [−∞ : −1.6284]



r = [−0.4278 : −0.0850]

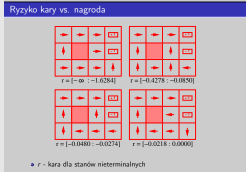


r = [−0.0480 : −0.0274]



r = [−0.0218 : 0.0000]

- r - kara dla stanów nieterminalnych



1. Polityka zmienia się w zależności od r. W tym przypadku manipulujemy tylko karą za ruch. Jeśli kara jest duża, należy jak najszybciej dostać się do stanu terminalnego niezależnie od tego, czy jest to pole +1 czy -1 (życie w tym świecie jest bardzo bolesne, więc **w tym świecie** samobójstwo jest jak najbardziej racjonalną opcją). Jeśli r jest małe (da się żyć), robimy wszystko, żeby tylko nie trafić na -1.

Ryzyko kary vs. nagroda c.d.

- Jak wygląda optymalna polityka dla $r > 0$ [zadanie 5]

Ryzyko kary vs. nagroda c.d.
- Jak wygląda optymalna polityka dla $r > 0$ [zadanie 5]

1. Życie w tym świecie jest bardzo przyjemne, więc trzeba robić wszystko, byleby żyć. Ucieczka od pól terminalnych. Pozostałe pola, obojętnie. Interesujące: w polu [4,3] należy iść w dół (ryzyko wpadnięcia na -1).

Ryzyko kary vs. nagroda c.d.

- Jak wygląda optymalna polityka dla $r > 0$ [zadanie 5]
- Kompromis pomiędzy **ryzykiem kary** a **szansą na nagrodę** jest cechą charakterystyczną MDP.
  - Nie występuje w problemach deterministycznych.
  - Dlatego MDP rozważane są w wielu dziedzinach:
    - AI
    - badania operacyjne
    - ekonomia
    - teoria sterowania

Ryzyko kary vs. nagroda c.d.
Jak wygląda optymalna polityka dla $r > 0$ [zadanie 5]
Kompromis pomiędzy **ryzykiem kary** a **szansą na nagrodę** jest cechą charakterystyczną MDP.
Nie występuje w problemach deterministycznych.
Dlatego MDP rozważane są w wielu dziedzinach:
AI
badania operacyjne
ekonomia
teoria sterowania

1. Życie w tym świecie jest bardzo przyjemne, więc trzeba robić wszystko, byleby żyć. Ucieczka od pól terminalnych. Pozostałe pola, obojętnie. Interesujące: w polu [4,3] należy iść w dół (ryzyko wpadnięcia na -1).

Stacjonarność preferencji sekwencji stanów

- W MDP **ocenie podlega sekwencja stanów** $\rightarrow$ musimy zrozumieć preferencje pomiędzy sekwencjami stanów.
- Naturalne założenie:
  preferencje względem sekwencji stanów są ***stacjonarne***, czyli:

$$[s_1, s_2, \ldots] \succ [s_1', s_2', \ldots] \implies [s_0, s_1, s_2, \ldots] \succ [s_0, s_1', s_2', \ldots]$$

Stacjonarność preferencji sekwencji stanów

- W MDP **ocenie podlega sekwencja stanów** $\rightarrow$ musimy zrozumieć preferencje pomiędzy sekwencjami stanów.
- Naturalne założenie:
  preferencje względem sekwencji stanów są ***stacjonarne***, czyli:

$[s_1, s_2, \ldots] \succ [s_1', s_2', \ldots] \implies [s_0, s_1, s_2, \ldots] \succ [s_0, s_1', s_2', \ldots]$

1. Dotychczas zakładaliśmy, że użyteczność jest sumą nagród za przejścia do stanów. Teraz zajmiemy się tym głębiej tym zagadnieniem i rozważymy inne możliwe definicje funkcji użyteczności.
2. Sekwencja stanów = historia obserwacji z pierwszego wykładu (jest pełna obserwowalność).
3. Rozrysować.

Użyteczność sekwencji stanów

Twierdzenie

Przy założeniu **stacjonarności** sekwencji stanów istnieją tylko dwie możliwości, aby przyporządkować użyteczności do sekwencji stanów:

- **Addytywna funkcja użyteczności**

$$U([s_0, s_1, s_2, \ldots]) = R(s_0) + R(s_1) + R(s_2) + \ldots$$

Użyteczność sekwencji stanów
Twierdzenie
Przy założeniu stacjonarności sekwencji stanów istnieją tylko dwie możliwości, aby przyporządkować użyteczności do sekwencji stanów:
Addytywna funkcja użyteczności
$U([s_0, s_1, s_2, \ldots]) = R(s_0) + R(s_1) + R(s_2) + \ldots$

1. ang. discounted utility function
2. współczynnik dyskontowy (ang. discount factor)
3. a, bo $U(a) = 2$, $U(b) = 0.9^4 \times 3 = 1.96$.
4. Zdyskontowana f. użyteczności ma uzasadnienie w życiu, w ekonomii, w ludzkich zachowaniach. Współczynnik dyskontowy decyduje, czy ważniejsze są nagrody w odległej przyszłości, czy dzisiaj. Czy lepiej dzisiaj zjeść lody, czy zbierać na maszynę do robienie lodów? Czy oszczędzać na telewizor, czy kupić go na raty?

Użyteczność sekwencji stanów

Twierdzenie

Przy założeniu **stacjonarności** sekwencji stanów istnieją tylko dwie możliwości, aby przyporządkować użyteczności do sekwencji stanów:

- **Addytywna funkcja użyteczności**

$$U([s_0, s_1, s_2, \dots]) = R(s_0) + R(s_1) + R(s_2) + \dots$$

- **Zdyskontowana funkcja użyteczności**:

$$U([s_0, s_1, s_2, \dots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \dots,$$

gdzie $\gamma \in (0, 1)$ jest współczynnikiem dyskontowym.

Która sekwencja nagród $a = [2, 0, 0, 0, 0]$ czy $b = [0, 0, 0, 0, 3]$ jest preferowana przy $\gamma = 0.9$? [zadanie 6]

Użyteczność sekwencji stanów

Twierdzenie
Przy założeniu **stacjonarności** sekwencji stanów istnieją tylko dwie możliwości, aby przyporządkować użyteczności do sekwencji stanów:
- **Addytywna funkcja użyteczności**
$U([s_0, s_1, s_2, \dots]) = R(s_0) + R(s_1) + R(s_2) + \dots$
- **Zdyskontowana funkcja użyteczności**:
$U([s_0, s_1, s_2, \dots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \dots,$
gdzie $\gamma \in (0, 1)$ jest współczynnikiem dyskontowym.

Która sekwencja nagród $a = [2, 0, 0, 0, 0]$ czy $b = [0, 0, 0, 0, 3]$ jest preferowana przy $\gamma = 0.9$? [zadanie 6]

1. ang. discounted utility function
2. współczynnik dyskontowy (ang. discount factor)
3. a, bo $U(a) = 2$, $U(b) = 0.9^4 \times 3 = 1.96$.
4. Zdyskontowana f. użyteczności ma uzasadnienie w życiu, w ekonomii, w ludzkich zachowaniach. Współczynnik dyskontowy decyduje, czy ważniejsze są nagrody w odległej przyszłości, czy dzisiaj. Czy lepiej dzisiaj zjeść lody, czy zbierać na maszynę do robienie lodów? Czy oszczędzać na telewizor, czy kupić go na raty?

Użyteczności — problem nieskończoności

Addytywna użyteczność może być nieskończona. **Nieskończona przyszłość** jest problemem, bo

jak porównać dwie sekwencje stanów, gdy użyteczności obu wynoszą $+\infty$?

Użyteczności — problem nieskończoności

Addytywna użyteczność może być nieskończona. **Nieskończona przyszłość** jest problemem, bo

jak porównać dwie sekwencje stanów, gdy użyteczności obu wynoszą $+\infty$?

1. Agent może krążyć pomiędzy dwoma nieterminalnymi stanami w nieskończoność zdobywając dodatnie nagrody.

Użyteczności — problem nieskończoności

Możliwe rozwiązania:

1. **Skończony horyzont:** koniec po liczbie kroków T
   - wtedy polityka π jest niestacjonarna: tzn. zależy od pozostałego do końca czasu.
   - przykład: pole [3,1] i T=3 vs. T=100.

Użyteczności — problem nieskończoności
Możliwe rozwiązania:
1. Skończony horyzont: koniec po liczbie kroków T
- wtedy polityka π jest niestacjonarna: tzn. zależy od pozostałego do końca czasu.
- przykład: pole [3,1] i T=3 vs. T=100.

1. Za „horyzontem", nagrody wynoszą już prawie 0.
2. Jeśli mamy stany absorbujące to oczekiwana użyteczność każdej polityki jest skończona.
3. Średnia nagroda: tak też można i to ma sens, ale tym się tutaj nie zajmujemy
4. Zwykle wybiera się opcję ze współczynnikiem dyskontowym.

Użyteczności — problem nieskończoności

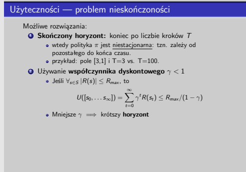
Możliwe rozwiązania:

1. **Skończony horyzont:** koniec po liczbie kroków T
   - wtedy polityka π jest niestacjonarna: tzn. zależy od pozostałego do końca czasu.
   - przykład: pole [3,1] i T=3 vs. T=100.
2. Używanie **współczynnika dyskontowego** $\gamma < 1$
   - Jeśli $\forall_{s \in S} |R(s)| \leq R_{max}$, to

$$U([s_0, \ldots s_\infty]) = \sum_{t=0}^{\infty} \gamma^t R(s_t) \leq R_{max}/(1-\gamma)$$

   - Mniejsze $\gamma \implies$ krótszy **horyzont**

2014-05-09



1. Za „horyzontem”, nagrody wynoszą już prawie 0.
2. Jeśli mamy stany absorbujące to oczekiwana użyteczność każdej polityki jest skończona.
3. Średnia nagroda: tak też można i to ma sens, ale tym się tutaj nie zajmujemy
4. Zwykle wybiera się opcję ze współczynnikiem dyskontowym.

Użyteczności — problem nieskończoności

Możliwe rozwiązania:

1. **Skończony horyzont:** koniec po liczbie kroków T
   - wtedy polityka π jest niestacjonarna: tzn. zależy od pozostałego do końca czasu.
   - przykład: pole [3,1] i T=3 vs. T=100.
2. Używanie **współczynnika dyskontowego** $\gamma < 1$
   - Jeśli $\forall_{s \in S} |R(s)| \leq R_{max}$, to

$$U([s_0, \ldots s_\infty]) = \sum_{t=0}^{\infty} \gamma^t R(s_t) \leq R_{max}/(1-\gamma)$$

   - Mniejsze $\gamma \implies$ krótszy **horyzont**
3. **Stan(y) absorbujące** $\implies$ z prawd. 1 agent w końcu zakończy działanie dla każdej polityki π
   - polityka, która zawsze prowadzi do stanu terminalnego, to **właściwa polityka**
   - możemy używać $\gamma = 1$

1. Za „horyzontem”, nagrody wynoszą już prawie 0.
2. Jeśli mamy stany absorbujące to oczekiwana użyteczność każdej polityki jest skończona.
3. Średnia nagroda: tak też można i to ma sens, ale tym się tutaj nie zajmujemy
4. Zwykle wybiera się opcję ze współczynnikiem dyskontowym.

Użyteczności — problem nieskończoności

Możliwe rozwiązania:

1. **Skończony horyzont:** koniec po liczbie kroków T
   - wtedy polityka π jest <u>niestacjonarna</u>: tzn. zależy od pozostałego do końca czasu.
   - przykład: pole [3,1] i T=3 vs. T=100.
2. Używanie **współczynnika dyskontowego** $\gamma < 1$
   - Jeśli $\forall_{s \in S} |R(s)| \leq R_{max}$, to
   $$U([s_0, \ldots s_\infty]) = \sum_{t=0}^{\infty} \gamma^t R(s_t) \leq R_{max}/(1-\gamma)$$
   - Mniejsze $\gamma \implies$ krótszy **horyzont**
3. **Stan(y) absorbujące** $\implies$ z prawd. 1 agent w końcu zakończy działanie dla każdej polityki π
   - polityka, która zawsze prowadzi do stanu terminalnego, to **właściwa polityka**
   - możemy używać $\gamma = 1$
4. **Średnia nagroda** — maksym. średniej wypłaty na krok

Użyteczności — problem nieskończoności
Możliwe rozwiązania:
1. Skończony horyzont: koniec po liczbie kroków T
- wtedy polityka π jest niestacjonarna: tzn. zależy od pozostałego do końca czasu.
- przykład: pole [3,1] i T=3 vs. T=100.
2. Używanie współczynnika dyskontowego $\gamma < 1$
- Jeśli $\forall_{s \in S} |R(s)| \leq R_{max}$, to
$$U([s_0, \ldots s_\infty]) = \sum_{t=0}^{\infty} \gamma^t R(s_t) \leq R_{max}/(1-\gamma)$$
- Mniejsze $\gamma \implies$ krótszy horyzont
3. Stan(y) absorbujące $\implies$ z prawd. 1 agent w końcu zakończy działanie dla każdej polityki π
- polityka, która zawsze prowadzi do stanu terminalnego, to właściwa polityka
- możemy używać $\gamma = 1$
4. Średnia nagroda — maksym. średniej wypłaty na krok

1. Za „horyzontem”, nagrody wynoszą już prawie 0.
2. Jeśli mamy stany absorbujące to oczekiwana użyteczność każdej polityki jest skończona.
3. Średnia nagroda: tak też można i to ma sens, ale tym się tutaj nie zajmujemy
4. Zwykle wybiera się opcję ze współczynnikiem dyskontowym.

Optymalna polityka i oczekiwana użyteczność

Jak porównywać polityki?

Porównywanie polityk → porównywanie **oczekiwanych wartości użyteczności** sekwencji stanów.

Niech agent realizuje **politykę** π zaczynając od **stanu** s. Wtedy:

- S_t — zmienna losowa, oznaczająca stan osiągnięty w momencie t (czyli $S_0 = s$)

Optymalna polityka i oczekiwana użyteczność
Jak porównywać polityki?
Porównywanie polityk → porównywanie **oczekiwanych wartości użyteczności** sekwencji stanów.
Niech agent realizuje **politykę** π zaczynając od **stanu** s. Wtedy:
- S_t — zmienna losowa, oznaczająca stan osiągnięty w momencie t (czyli $S_0 = s$)

1. które wynikają z realizowania danej polityki.
2. Tak. Optymalna polityka dla MDP **nie zależy od stanu początkowego**. Dlatego po prostu piszemy π^*. Ale to jest tylko prawdą jeśli używamy zdyskontowanej użyteczności z nieskończonym horyzontem.
   Intuicyjnie: Jeśli π^*_a jest optymalna przy rozpoczęciu ze stanu a, a π^*_b przy rozpoczęciu ze stanu b, to jeśli obie dotrą do stanu c, obie muszą mieć tę samą użyteczność ze stanu c.
   Dlatego będziemy po prostu pisali $\pi^* = \pi^*_a = \pi^*_b$.

Optymalna polityka i oczekiwana użyteczność

Jak porównywać polityki?

Porównywanie polityk $\rightarrow$ porównywanie **oczekiwanych wartości użyteczności** sekwencji stanów.

Niech agent realizuje **politykę** π zaczynając od **stanu** s. Wtedy:

- S_t — zmienna losowa, oznaczająca stan osiągnięty w momencie t (czyli $S_0 = s$)
- **Oczekiwana użyteczność** przy stosowanie polityki π od stanu s:

$$U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(S_t)\right]$$

Optymalna polityka i oczekiwana użyteczność
Jak porównywać polityki?
Porównywanie polityk → porównywanie oczekiwanych wartości użyteczności sekwencji stanów.
Niech agent realizuje politykę π zaczynając od stanu s. Wtedy:
S_t — zmienna losowa, oznaczająca stan osiągnięty w momencie t (czyli S_0 = s)
Oczekiwana użyteczność przy stosowanie polityki π od stanu s:
U^π(s) = E[Σ_{t=0}^∞ γ^t R(S_t)]

1. które wynikają z realizowania danej polityki.
2. Tak. Optymalna polityka dla MDP **nie zależy od stanu początkowego**. Dlatego po prostu piszemy π^*. Ale to jest tylko prawdą jeśli używamy zdyskontowanej użyteczności z nieskończonym horyzontem.
   Intuicyjnie: Jeśli π^*_a jest optymalna przy rozpoczęciu ze stanu a, a π^*_b przy rozpoczęciu ze stanu b, to jeśli obie dotrą do stanu c, obie muszą mieć tę samą użyteczność ze stanu c.
   Dlatego będziemy po prostu pisali $\pi^* = \pi^*_a = \pi^*_b$.

Optymalna polityka i oczekiwana użyteczność

Jak porównywać polityki?

Porównywanie polityk $\rightarrow$ porównywanie **oczekiwanych wartości użyteczności** sekwencji stanów.

Niech agent realizuje **politykę** π zaczynając od **stanu** s. Wtedy:

- S_t — zmienna losowa, oznaczająca stan osiągnięty w momencie t (czyli $S_0 = s$)
- **Oczekiwana użyteczność** przy stosowanie polityki π od stanu s:

$$U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(S_t)\right]$$

- Optymalna polityka ze stanu to

$$\pi_s^* = \mathrm{argmax}_{\pi} \mathrm{U}^{\pi}(\mathrm{s})$$

Optymalna polityka i oczekiwana użyteczność
Jak porównywać polityki?
Porównywanie polityk $\rightarrow$ porównywanie **oczekiwanych wartości użyteczności** sekwencji stanów.
Niech agent realizuje **politykę** π zaczynając od **stanu** s. Wtedy:
S_t — zmienna losowa, oznaczająca stan osiągnięty w momencie t (czyli $S_0 = s$)
Oczekiwana użyteczność przy stosowanie polityki π od stanu s:
$U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(S_t)\right]$
Optymalna polityka ze stanu to
$\pi_s^* = \mathrm{argmax}_{\pi} \mathrm{U}^{\pi}(\mathrm{s})$

1. które wynikają z realizowania danej polityki.
2. Tak. Optymalna polityka dla MDP **nie zależy od stanu początkowego**. Dlatego po prostu piszemy π^*. Ale to jest tylko prawdą jeśli używamy zdyskontowanej użyteczności z nieskończonym horyzontem.
   Intuicyjnie: Jeśli π_a^* jest optymalna przy rozpoczęciu ze stanu a, a π_b^* przy rozpoczęciu ze stanu b, to jeśli obie dotrą do stanu c, obie muszą mieć tę samą użyteczność ze stanu c.
   Dlatego będziemy po prostu pisali $\pi^* = \pi_a^* = \pi_b^*$.

Optymalna polityka i oczekiwana użyteczność

Jak porównywać polityki?

Porównywanie polityk → porównywanie **oczekiwanych wartości użyteczności** sekwencji stanów.

Niech agent realizuje **politykę** π zaczynając od **stanu** s. Wtedy:

- S_t — zmienna losowa, oznaczająca stan osiągnięty w momencie t (czyli $S_0 = s$)
- **Oczekiwana użyteczność** przy stosowanie polityki π od stanu s:

$$U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(S_t)\right]$$

- Optymalna polityka ze stanu to

$$\pi^*_s = \mathrm{argmax}_{\pi} \mathrm{U}^{\pi}(\mathrm{s})$$

- Zadanie: mamy dwa stany $a \neq b$. Czy $\pi^*_a = \pi^*_b$?[zadanie 7]

Optymalna polityka i oczekiwana użyteczność
Jak porównywać polityki?
Porównywanie polityk → porównywanie oczekiwanych wartości użyteczności sekwencji stanów.
Niech agent realizuje politykę π zaczynając od stanu s. Wtedy:
S_t — zmienna losowa, oznaczająca stan osiągnięty w momencie t (czyli $S_0 = s$)
Oczekiwana użyteczność przy stosowanie polityki π od stanu s:
$U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(S_t)\right]$
Optymalna polityka ze stanu to
$\pi^*_s = \mathrm{argmax}_{\pi} \mathrm{U}^{\pi}(\mathrm{s})$
Zadanie: mamy dwa stany $a \neq b$. Czy $\pi^*_a = \pi^*_b$?[zadanie 7]

1. które wynikają z realizowania danej polityki.
2. Tak. Optymalna polityka dla MDP **nie zależy od stanu początkowego**. Dlatego po prostu piszemy π^*. Ale to jest tylko prawdą jeśli używamy zdyskontowanej użyteczności z nieskończonym horyzontem.
   Intuicyjnie: Jeśli π^*_a jest optymalna przy rozpoczęciu ze stanu a, a π^*_b przy rozpoczęciu ze stanu b, to jeśli obie dotrą do stanu c, obie muszą mieć tę samą użyteczność ze stanu c.
   Dlatego będziemy po prostu pisali $\pi^* = \pi^*_a = \pi^*_b$.

Użyteczność stanu

$U^{\pi^*}(s)$ jest (oczekiwaną, zdyskontowaną) **użytecznością optymalnej polityki** realizowanej od stanu s.

- Będziemy ją oznaczać $U(s)$.

Użyteczność stanu

$U^{\pi^*}(s)$ jest (oczekiwaną, zdyskontowaną) **użytecznością optymalnej polityki** realizowanej od stanu s.

- Będziemy ją oznaczać $U(s)$.

Użyteczność stanu

$U^{\pi^*}(s)$ jest (oczekiwaną, zdyskontowaną) **użytecznością optymalnej polityki** realizowanej od stanu s.

- Będziemy ją oznaczać $U(s)$.
- Czyli możemy ją interpretować jako **użyteczność stanu** s.

Użyteczność stanu
$U^{\pi^*}(s)$ jest (oczekiwaną, zdyskontowaną) **użytecznością optymalnej polityki** realizowanej od stanu s.
- Będziemy ją oznaczać $U(s)$.
- Czyli możemy ją interpretować jako **użyteczność stanu** s.

Użyteczność stanu

$U^{\pi^*}(s)$ jest (oczekiwaną, zdyskontowaną) **użytecznością optymalnej polityki** realizowanej od stanu s.

- Będziemy ją oznaczać $U(s)$.
- Czyli możemy ją interpretować jako **użyteczność stanu** s.

Użyteczność stanu

$U(s)$ jest oczekiwaną, zdyskontowaną sumą nagród uzyskanych <u>przy realizowaniu optymalnej polityki</u> od stanu s

Porównanie R i U:

- $R(s)$ — nagroda krótkoterminowa
- $U(s)$ — nagroda długoterminowa

Użyteczność stanu
$U^{\pi^*}(s)$ jest (oczekiwaną, zdyskontowaną) **użytecznością optymalnej polityki** realizowanej od stanu s.
- Będziemy ją oznaczać $U(s)$.
- Czyli możemy ją interpretować jako **użyteczność stanu** s.
Użyteczność stanu
$U(s)$ jest oczekiwaną, zdyskontowaną sumą nagród uzyskanych przy realizowaniu optymalnej polityki od stanu s
Porównanie R i U:
- $R(s)$ — nagroda krótkoterminowa
- $U(s)$ — nagroda długoterminowa

Użyteczność stanu

$U^{\pi^*}(s)$ jest (oczekiwaną, zdyskontowaną) **użytecznością optymalnej polityk**i realizowanej od stanu s.

- Będziemy ją oznaczać $U(s)$.
- Czyli możemy ją interpretować jako **użyteczność stanu** s.

Użyteczność stanu

$U(s)$ jest oczekiwaną, zdyskontowaną sumą nagród uzyskanych przy realizowaniu optymalnej polityki od stanu s

Porównanie R i U:

- $R(s)$ — nagroda krótkoterminowa
- $U(s)$ — nagroda długoterminowa

Użyteczność stanu

$U^{\pi^*}(s)$ jest (oczekiwaną, zdyskontowaną) **użytecznością optymalnej polityk**i realizowanej od stanu s.

- Będziemy ją oznaczać $U(s)$.
- Czyli możemy ją interpretować jako **użyteczność stanu** s.

Użyteczność stanu

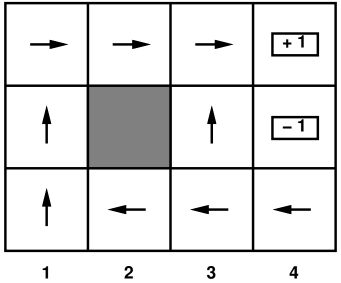
$U(s)$ jest oczekiwaną, zdyskontowaną sumą nagród uzyskanych przy realizowaniu optymalnej polityki od stanu s

Porównanie R i U:

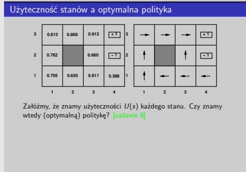
- $R(s)$ — nagroda krótkoterminowa
- $U(s)$ — nagroda długoterminowa

Użyteczność stanów a optymalna polityka

3	0.812	0.868	0.912	+1
2	0.762		0.660	−1
1	0.705	0.655	0.611	0.388
	1	2	3	4



Załóżmy, że znamy użyteczności $U(s)$ każdego stanu. Czy znamy wtedy (optymalną) politykę? [zadanie 8]



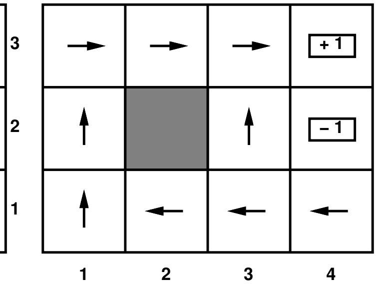
1. Uwaga: nie wybieramy akcji, które prowadzą po prostu do stanów o najwyższej użyteczności! Musimy wziąć pod uwagę także model tranzycji, czyli to jest niepoprawne:

$$\pi^*(s) = \operatorname{argmax}_{a \in A(s)} \sum_{s'} U(s')$$

2. Ważne: ten sposób okreslania polityki możemy zastosować nawet, gdy użyteczności stanów nie są „optymalne”. Wtedy „odczytana” w ten sposób polityka też nie będzie optymalna, w ogólności.

Użyteczność stanów a optymalna polityka

	1	2	3	4
3	0.812	0.868	0.912	+ 1
2	0.762		0.660	– 1
1	0.705	0.655	0.611	0.388

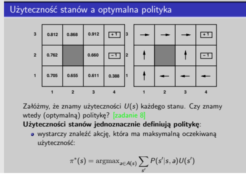


Załóżmy, że znamy użyteczności $U(s)$ każdego stanu. Czy znamy wtedy (optymalną) politykę? [zadanie 8]

Użyteczności stanów jednoznacznie definiują politykę:

- wystarczy znaleźć akcję, która ma maksymalną oczekiwaną użyteczność:

$$\pi^*(s) = \operatorname{argmax}_{a \in A(s)} \sum_{s'} P(s'|s,a)U(s')$$



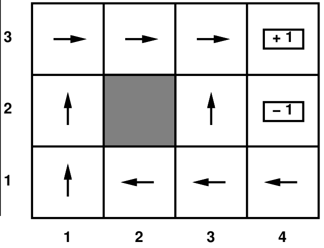
1. Uwaga: nie wybieramy akcji, które prowadzą po prostu do stanów o najwyższej użyteczności! Musimy wziąć pod uwagę także model tranzycji, czyli to jest niepoprawne:

$$\pi^*(s) = \operatorname{argmax}_{a \in A(s)} \sum_{s'} U(s')$$

2. Ważne: ten sposób okreslania polityki możemy zastosować nawet, gdy użyteczności stanów nie są „optymalne”. Wtedy „odczytana” w ten sposób polityka też nie będzie optymalna, w ogólności.

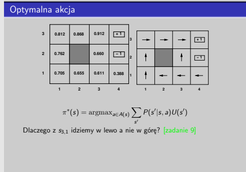
Optymalna akcja

3	0.812	0.868	0.912	+ 1
2	0.762		0.660	− 1
1	0.705	0.655	0.611	0.388
	1	2	3	4



$$\pi^*(s) = \text{argmax}_{a \in A(s)} \sum_{s'} P(s'|s,a)U(s')$$

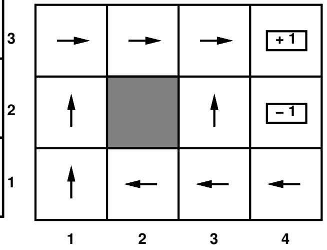
Dlaczego z $s_{3,1}$ idziemy w lewo a nie w górę? [zadanie 9]



1. Obserwacja szczegółowa: winne jest pole (4,1), bo U(4,1) jest male (0.388)

Optymalna akcja

3	0.812	0.868	0.912	+ 1
2	0.762		0.660	– 1
1	0.705	0.655	0.611	0.388
	1	2	3	4

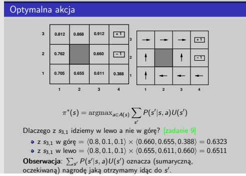


$$\pi^*(s) = \operatorname{argmax}_{a \in A(s)} \sum_{s'} P(s'|s,a)U(s')$$

Dlaczego z $s_{3,1}$ idziemy w lewo a nie w górę? [zadanie 9]

- z $s_{3,1}$ w górę $= \langle 0.8, 0.1, 0.1\rangle \times \langle 0.660, 0.655, 0.388\rangle = 0.6323$
- z $s_{3,1}$ w lewo $= \langle 0.8, 0.1, 0.1\rangle \times \langle 0.655, 0.611, 0.660\rangle = 0.6511$

Obserwacja: $\sum_{s'} P(s'|s,a)U(s')$ oznacza (sumaryczną, oczekiwaną) nagrodę jaką otrzymamy idąc do s'.



1. Obserwacja szczegółowa: winne jest pole (4,1), bo U(4,1) jest male (0.388)

Programowanie dynamiczne: równanie Bellmana

Równanie Bellman'a (1957)

oczekiwana suma wypłat $=$ aktualna wypłata
$+ \gamma \times$ oczekiwana suma wypłat po wybraniu najlepszej akcji:

$$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} U(s')P(s'|s,a)$$

Programowanie dynamiczne: równanie Bellmana
Równanie Bellman'a (1957)
oczekiwana suma wypłat $=$ aktualna wypłata
$+ \gamma \times$ oczekiwana suma wypłat po wybraniu najlepszej akcji:
$$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} U(s')P(s'|s,a)$$

1. Definicja użyteczności stanów prowadzi do prostej zależności pomiędzy użytecznościami sąsiadujących ze sobą stanów:
2. Nagroda $=$ nagroda krótkoterminowa (teraz) $+$ nagroda długoterminowa (w przyszłości)

Programowanie dynamiczne: równanie Bellmana

Równanie Bellman'a (1957)

oczekiwana suma wypłat $=$ aktualna wypłata
$+ \ \gamma \ \times$ oczekiwana suma wypłat po wybraniu najlepszej akcji:

$$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} U(s')P(s'|s,a)$$

Przykład:

$$\begin{aligned} U(1,1) = -0.04 + \gamma \max\{ & \\ & 0.8\times U(1,2) + 0.1\times U(2,1) + 0.1\times U(1,1), \\ & 0.9\times U(1,1) + 0.1\times U(1,2) \\ & 0.9\times U(1,1) + 0.1\times U(2,1), \\ & 0.8\times U(2,1) + 0.1\times U(1,2) + 0.1\times U(1,1)\} \end{aligned}$$

Programowanie dynamiczne: równanie Bellmana
Równanie Bellman'a (1957)
oczekiwana suma wypłat = aktualna wypłata
+ γ × oczekiwana suma wypłat po wybraniu najlepszej akcji:
$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} U(s')P(s'|s,a)$
Przykład:
$U(1,1) = -0.04 + \gamma \max\{ 0.8\times U(1,2) + 0.1\times U(2,1) + 0.1\times U(1,1), 0.9\times U(1,1) + 0.1\times U(1,2) 0.9\times U(1,1) + 0.1\times U(2,1), 0.8\times U(2,1) + 0.1\times U(1,2) + 0.1\times U(1,1)\}$

1. Definicja użyteczności stanów prowadzi do prostej zależności pomiędzy użytecznościami sąsiadujących ze sobą stanów:
2. Nagroda $=$ nagroda krótkoterminowa (teraz) $+$ nagroda długoterminowa (w przyszłości)

Układ równań Bellmana

Układ równań:

$$\begin{cases} U(s_0) = R(s_0) + \gamma \max_{a \in A(s_0)} \sum_{s'} U(s')P(s'|s_0, a) \\ U(s_1) = R(s_1) + \gamma \max_{a \in A(s_1)} \sum_{s'} U(s')P(s'|s_1, a) \\ \vdots \\ U(s_n) = R(s_n) + \gamma \max_{a \in A(s_n)} \sum_{s'} U(s')P(s'|s_n, a) \end{cases}$$

- 1 równanie na stan + n stanów $\implies$ n równań z n niewiadomymi.
- Równania są liniowe czy nieliniowe? [zadanie 10]

Układ równań Bellmana

Układ równań:

$$\begin{cases} U(s_0) = R(s_0) + \gamma \max_{a \in A(s_0)} \sum_{s'} U(s')P(s'|s_0, a) \\ U(s_1) = R(s_1) + \gamma \max_{a \in A(s_1)} \sum_{s'} U(s')P(s'|s_1, a) \\ \vdots \\ U(s_n) = R(s_n) + \gamma \max_{a \in A(s_n)} \sum_{s'} U(s')P(s'|s_n, a) \end{cases}$$

- 1 równanie na stan + n stanów $\implies$ n równań z n niewiadomymi.
- Równania są liniowe czy nieliniowe? [zadanie 10]

1. Nieliniowe (max). Nie tak łatwo jest rozwiązać $\implies$ podejście iteracyjne

Algorytm iteracji wartości

Algorytm iteracji wartości

1. Rozpocznij z losowymi wartościami użyteczności

Algorytm iteracji wartości
Algorytm iteracji wartości
1. Rozpocznij z losowymi wartościami użyteczności

1. Ang. Value Iteration

Algorytm iteracji wartości

Algorytm iteracji wartości

1. Rozpocznij z losowymi wartościami użyteczności
2. Uaktualnij użyteczności zgodne z układem równań Bellmana. Dla wszystkich stanów s:

$$U_{i+1}(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} U_i(s')P(s'|s,a)$$

- Uwaga: wykonujemy synchronicznie (kopia tablicy U).

Algorytm iteracji wartości
Algorytm iteracji wartości
1. Rozpocznij z losowymi wartościami użyteczności
2. Uaktualnij użyteczności zgodne z układem równań Bellmana. Dla wszystkich stanów s:
$U_{i+1}(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} U_i(s')P(s'|s,a)$
- Uwaga: wykonujemy synchronicznie (kopia tablicy U).

1. Ang. Value Iteration

Algorytm iteracji wartości

Algorytm iteracji wartości

1. Rozpocznij z losowymi wartościami użyteczności
2. Uaktualnij użyteczności zgodne z układem równań Bellmana. Dla wszystkich stanów s:

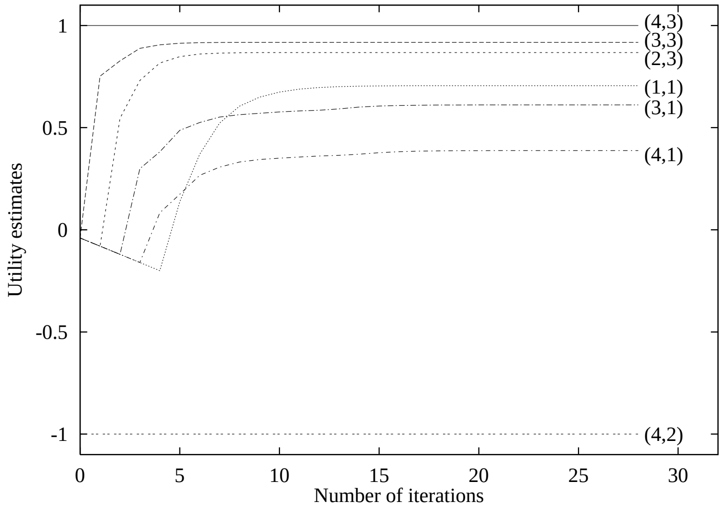
$$U_{i+1}(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} U_i(s') P(s'|s, a)$$

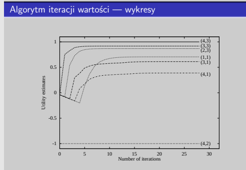
   - Uwaga: wykonujemy synchronicznie (kopia tablicy U).
3. Jeśli osiągnęliśmy równowagę (brak zmian), to mamy **globalne** optimum.

Algorytm iteracji wartości
Algorytm iteracji wartości
1. Rozpocznij z losowymi wartościami użyteczności
2. Uaktualnij użyteczności zgodne z układem równań Bellmana. Dla wszystkich stanów s:
$$U_{i+1}(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} U_i(s') P(s'|s, a)$$
- Uwaga: wykonujemy synchronicznie (kopia tablicy U).
3. Jeśli osiągnęliśmy równowagę (brak zmian), to mamy **globalne** optimum.

1. Ang. Value Iteration

Algorytm iteracji wartości — wykresy





1. Można o tym algorytmie myśleć jak o propagowaniu się informacji poprzez przestrzeń stanów za pomocą lokalnych uaktualnień.

Algorytm iteracji wartości — uwagi i zbieżność

- Algorytm jest zbieżny do globalnego optimum
- Liczba wymaganych iteracji wynosi:

$$N = \lceil \log(2R_{\max}/\epsilon(1-\gamma))/\log(1/\gamma) \rceil ,$$

gdzie:
  - ϵ jest maksymalnym błędem pomiędzy obliczoną użytecznością stanu a użytecznością rzeczywistą
  - $R_{\max}$ jest maksymalną wartością nagrody
- Stosowane kryterium stopu:

$$\|U_{i+1} - U_i\| < \epsilon(1-\gamma)/\gamma$$

- **Optymalna polityka jest dostępna zanim wartości użyteczności zbiegną do idealnych**.
- Powyższe działa, gdy $\gamma < 1$. Jeśli $\gamma = 1$, trzeba innego kryterium stopu.

Algorytm iteracji wartości — uwagi i zbieżność
- Algorytm jest zbieżny do globalnego optimum
- Liczba wymaganych iteracji wynosi:
$$N = \lceil \log(2R_{\max}/\epsilon(1-\gamma))/\log(1/\gamma) \rceil ,$$
gdzie:
  - ϵ jest maksymalnym błędem pomiędzy obliczoną użytecznością stanu a użytecznością rzeczywistą
  - $R_{\max}$ jest maksymalną wartością nagrody
- Stosowane kryterium stopu:
$$\|U_{i+1} - U_i\| < \epsilon(1-\gamma)/\gamma$$
- **Optymalna polityka jest dostępna zanim wartości użyteczności zbiegną do idealnych**.
- Powyższe działa, gdy $\gamma < 1$. Jeśli $\gamma = 1$, trzeba innego kryterium stopu.

Algorytm iteracji polityki

- **Obserwacja**: można otrzymać optymalną politykę nawet gdy wartości funkcji użyteczności są niedokładne.
- **Pomysł**: Szukaj optymalnej polityki oraz wartości funkcji użyteczności równocześnie.

Algorytm iteracji polityki
- **Obserwacja**: można otrzymać optymalną politykę nawet gdy wartości funkcji użyteczności są niedokładne.
- **Pomysł**: Szukaj optymalnej polityki oraz wartości funkcji użyteczności równocześnie.

1. (ang. Policy Iteration)
2. Czyli użyteczność polityki π.
3. Krok Ulepszanie polityki: To jest proste (i już to widzieliśmy).
4. Chociaż użyteczności wcale nie są poprawne.

Algorytm iteracji polityki

- **Obserwacja**: można otrzymać optymalną politykę nawet gdy wartości funkcji użyteczności są niedokładne.
- **Pomysł**: Szukaj optymalnej polityki oraz wartości funkcji użyteczności równocześnie.

Algorytm (Howard, 1960)

1. $\pi \leftarrow$ losowa polityka

Algorytm iteracji polityki
Obserwacja: można otrzymać optymalną politykę nawet gdy wartości funkcji użyteczności są niedokładne.
Pomysł: Szukaj optymalnej polityki oraz wartości funkcji użyteczności równocześnie.
Algorytm (Howard, 1960)
$\pi \leftarrow$ losowa polityka

1. (ang. Policy Iteration)
2. Czyli użyteczność polityki π.
3. Krok Ulepszanie polityki: To jest proste (i już to widzieliśmy).
4. Chociaż użyteczności wcale nie są poprawne.

Algorytm iteracji polityki

- **Obserwacja**: można otrzymać optymalną politykę nawet gdy wartości funkcji użyteczności są niedokładne.
- **Pomysł**: Szukaj optymalnej polityki oraz wartości funkcji użyteczności równocześnie.

Algorytm (Howard, 1960)

1. $\pi \leftarrow$ losowa polityka
2. Powtarzaj dopóki π się zmienia:

Algorytm iteracji polityki
Obserwacja: można otrzymać optymalną politykę nawet gdy wartości funkcji użyteczności są niedokładne.
Pomysł: Szukaj optymalnej polityki oraz wartości funkcji użyteczności równocześnie.
Algorytm (Howard, 1960)
π ← losowa polityka
Powtarzaj dopóki π się zmienia:

1. (ang. Policy Iteration)
2. Czyli użyteczność polityki π.
3. Krok Ulepszanie polityki: To jest proste (i już to widzieliśmy).
4. Chociaż użyteczności wcale nie są poprawne.

Algorytm iteracji polityki

- **Obserwacja**: można otrzymać optymalną politykę nawet gdy wartości funkcji użyteczności są niedokładne.
- **Pomysł**: Szukaj optymalnej polityki oraz wartości funkcji użyteczności równocześnie.

Algorytm (Howard, 1960)

1. $\pi \leftarrow$ losowa polityka
2. Powtarzaj dopóki π się zmienia:
    1. (**Ocena polityki**) Oblicz użyteczność $U^{\pi}(s)$ dla wszystkich stanów $s \in S$.

Algorytm iteracji polityki
- Obserwacja: można otrzymać optymalną politykę nawet gdy wartości funkcji użyteczności są niedokładne.
- Pomysł: Szukaj optymalnej polityki oraz wartości funkcji użyteczności równocześnie.
Algorytm (Howard, 1960)
1. $\pi \leftarrow$ losowa polityka
2. Powtarzaj dopóki π się zmienia:
    1. (Ocena polityki) Oblicz użyteczność $U^{\pi}(s)$ dla wszystkich stanów $s \in S$.

1. (ang. Policy Iteration)
2. Czyli użyteczność polityki π.
3. Krok Ulepszanie polityki: To jest proste (i już to widzieliśmy).
4. Chociaż użyteczności wcale nie są poprawne.

Algorytm iteracji polityki

- **Obserwacja**: można otrzymać optymalną politykę nawet gdy wartości funkcji użyteczności są niedokładne.
- **Pomysł**: Szukaj optymalnej polityki oraz wartości funkcji użyteczności równocześnie.

Algorytm (Howard, 1960)

1. $\pi \leftarrow$ losowa polityka
2. Powtarzaj dopóki π się zmienia:
   1. (**Ocena polityki**) Oblicz użyteczność $U^{\pi}(s)$ dla wszystkich stanów $s \in S$.
   2. (**Ulepszenie polityki**) Oblicz nową politykę π zakładając, że $U^{\pi}(s) = U(s)$ (użyteczności π są poprawne):

$$\pi(s) = \operatorname{argmax}_{a \in A(s)} \sum_{s'} P(s'|s,a)U(s')$$

Algorytm iteracji polityki
Obserwacja: można otrzymać optymalną politykę nawet gdy wartości funkcji użyteczności są niedokładne.
Pomysł: Szukaj optymalnej polityki oraz wartości funkcji użyteczności równocześnie.
Algorytm (Howard, 1960)
$\pi \leftarrow$ losowa polityka
Powtarzaj dopóki π się zmienia:
(Ocena polityki) Oblicz użyteczność $U^{\pi}(s)$ dla wszystkich stanów $s \in S$.
(Ulepszenie polityki) Oblicz nową politykę π zakładając, że $U^{\pi}(s) = U(s)$ (użyteczności π są poprawne):
$\pi(s) = \operatorname{argmax}_{a \in A(s)} \sum_{s'} P(s'|s,a)U(s')$

1. (ang. Policy Iteration)
2. Czyli użyteczność polityki π.
3. Krok Ulepszanie polityki: To jest proste (i już to widzieliśmy).
4. Chociaż użyteczności wcale nie są poprawne.

Algorytm iteracji polityki — ocena polityki

Aby obliczyć użyteczności polityki π wystarczy dla wszystkich stanów policzyć:

$$U^{\pi}(s) = R(s) + \gamma \sum_{s'} U^{\pi}(s') P(s'|s, \pi(s))$$

Algorytm iteracji polityki — ocena polityki

Aby obliczyć użyteczności polityki π wystarczy dla wszystkich stanów policzyć:

$$U^{\pi}(s) = R(s) + \gamma \sum_{s'} U^{\pi}(s') P(s'|s, \pi(s))$$

1. Nie ma już argmax'a, ponieważ liczymy użyteczność dla znanej polityki.
2. $O(n^3)$, a nawet w $O(n^{2.3})$

Algorytm iteracji polityki — ocena polityki

Aby obliczyć użyteczności polityki π wystarczy dla wszystkich stanów policzyć:

$$U^{\pi}(s) = R(s) + \gamma \sum_{s'} U^{\pi}(s')P(s'|s, \pi(s))$$

Tzn, mamy układ n liniowych równań i n niewiadomych. Można go rozwiązać w czasie: [zadanie 11]

- $O(n)$?
- $O(n^2)$?
- $O(n^3)$?

Algorytm iteracji polityki — ocena polityki

Aby obliczyć użyteczności polityki π wystarczy dla wszystkich stanów policzyć:

$$U^{\pi}(s) = R(s) + \gamma \sum_{s'} U^{\pi}(s')P(s'|s, \pi(s))$$

Tzn, mamy układ n liniowych równań i n niewiadomych. Można go rozwiązać w czasie: [zadanie 11]

- $O(n)$?
- $O(n^2)$?
- $O(n^3)$?

1. Nie ma już argmax'a, ponieważ liczymy użyteczność dla znanej polityki.
2. $O(n^3)$, a nawet w $O(n^{2.3})$

Zmodyfikowany algorytm iteracji polityki

Algorytm iteracji polityki:

- bardzo szybko zbiega do optimum, ale
- każdy jego krok jest kosztowny obliczeniowo, gdy stanów jest dużo:
  - $O(n^3)$ zaczyna być bolesne

Zmodyfikowany algorytm iteracji polityki
Algorytm iteracji polityki:
- bardzo szybko zbiega do optimum, ale
- każdy jego krok jest kosztowny obliczeniowo, gdy stanów jest dużo:
  - $O(n^3)$ zaczyna być bolesne

Zmodyfikowany algorytm iteracji polityki

Algorytm iteracji polityki:

- bardzo szybko zbiega do optimum, ale
- każdy jego krok jest kosztowny obliczeniowo, gdy stanów jest dużo:
  - $O(n^3)$ zaczyna być bolesne

Pomysł: Obliczmy więc iteracyjnie przybliżoną wartość $U(s)$

- (Przybliżoną) użyteczność $U(s)$ obliczamy wykonując k kroków algorytmu iteracji wartości (ze stałą polityką π) rozpoczynając od ostatnio znanego $U(s)$, czyli:

$$U_{i+1}(s) \leftarrow R(s) + \gamma \sum_{s'} P(s'|s, \pi_i(s)) U_i(s')$$

Zmodyfikowany algorytm iteracji polityki
Algorytm iteracji polityki:
- bardzo szybko zbiega do optimum, ale
- każdy jego krok jest kosztowny obliczeniowo, gdy stanów jest dużo:
  - $O(n^3)$ zaczyna być bolesne
Pomysł: Obliczmy więc iteracyjnie przybliżoną wartość $U(s)$
- (Przybliżoną) użyteczność $U(s)$ obliczamy wykonując k kroków algorytmu iteracji wartości (ze stałą polityką π) rozpoczynając od ostatnio znanego $U(s)$, czyli:
$$U_{i+1}(s) \leftarrow R(s) + \gamma \sum_{s'} P(s'|s, \pi_i(s)) U_i(s')$$

Zmodyfikowany algorytm iteracji polityki

Algorytm iteracji polityki:

- bardzo szybko zbiega do optimum, ale
- każdy jego krok jest kosztowny obliczeniowo, gdy stanów jest dużo:
  - $O(n^3)$ zaczyna być bolesne

Pomysł: Obliczmy więc iteracyjnie przybliżoną wartość $U(s)$

- (Przybliżoną) użyteczność $U(s)$ obliczamy wykonując k kroków algorytmu iteracji wartości (ze stałą polityką π) rozpoczynając od ostatnio znanego $U(s)$, czyli:

$$U_{i+1}(s) \leftarrow R(s) + \gamma \sum_{s'} P(s'|s, \pi_i(s)) U_i(s')$$

- Zwykle szybciej zbiega znacznie szybciej niż „czysty" algorytm iteracji wartości lub iteracji polityki.

Zmodyfikowany algorytm iteracji polityki
Algorytm iteracji polityki:
- bardzo szybko zbiega do optimum, ale
- każdy jego krok jest kosztowny obliczeniowo, gdy stanów jest dużo:
  - $O(n^3)$ zaczyna być bolesne
Pomysł: Obliczmy więc iteracyjnie przybliżoną wartość $U(s)$
- (Przybliżoną) użyteczność $U(s)$ obliczamy wykonując k kroków algorytmu iteracji wartości (ze stałą polityką π) rozpoczynając od ostatnio znanego $U(s)$, czyli:
$$U_{i+1}(s) \leftarrow R(s) + \gamma \sum_{s'} P(s'|s, \pi_i(s)) U_i(s')$$
- Zwykle szybciej zbiega znacznie szybciej niż „czysty" algorytm iteracji wartości lub iteracji polityki.

Rozszerzenia

Algorytmy synchronicznie aktualizowały użyteczności poszczególnych stanów.
W praktyce nie jest to konieczne. Można:

- Wybrać jakikolwiek podzbiór stanów i
- zaaplikować do niego którąkolwiek aktualizację (**ulepszenie polityki** lub **iterację wartości**)
- $\Longrightarrow$ algorytm **Asynchronicznej Iteracji Polityki**
  - Pod pewnymi warunkami dot. początkowych użyteczności i początkowej polityki jest zbieżny
  - Umożliwia dobranie heurystyki wyboru stanów do aktualizacji, np. algorytm, który koncentruje się na ocenie użyteczności stanów, które z dużym prawd. mają szansę być odwiedzone.

2014-05-09

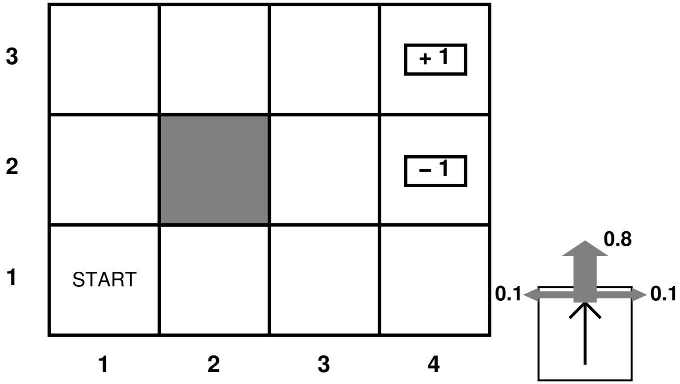
Problemy Decyzyjne Markowa
└ Rozwiązywanie MDP
└ Rozszerzenia

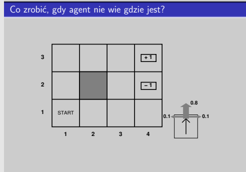
Rozszerzenia

Algorytmy synchronicznie aktualizowały użyteczności poszczególnych stanów.
W praktyce nie jest to konieczne. Można:

- Wybrać jakikolwiek podzbiór stanów i
- zaaplikować do niego którąkolwiek aktualizację (**ulepszenie polityki** lub **iterację wartości**)
- $\Longrightarrow$ algorytm **Asynchronicznej Iteracji Polityki**
  - Pod pewnymi warunkami dot. początkowych użyteczności i początkowej polityki jest zbieżny
  - Umożliwia dobranie heurystyki wyboru stanów do aktualizacji, np. algorytm, który koncentruje się na ocenie użyteczności stanów, które z dużym prawd. mają szansę być odwiedzone.

Co zrobić, gdy agent nie wie gdzie jest?





Rzecz o częściowej obserwowalności

- Agent nie zna aktualnego stanu środowiska („nie wie gdzie jest") $\implies$ nie ma sensu mówić o polityce $\pi(s)$!
- Musi zbierać informacje i wnioskować na temat możliwych stanów środowiska (**stan przekonań**, czyli rozkład prawd. nad stanami) $\implies$ **estymacja stanu** (filtrowanie).

Rzecz o częściowej obserwowalności

- Agent nie zna aktualnego stanu środowiska („nie wie gdzie jest") $\implies$ nie ma sensu mówić o polityce $\pi(s)$!
- Musi zbierać informacje i wnioskować na temat możliwych stanów środowiska (**stan przekonań**, czyli rozkład prawd. nad stanami) $\implies$ **estymacja stanu** (filtrowanie).

POMDP

POMDP:

- **zbiór stanów** S,
- **zbiór akcji** A,
- **model przejść** $P(s'|s,a)$,
- **funkcja nagrody** $R(s)$,

POMDP

POMDP:
- zbiór stanów S,
- zbiór akcji A,
- model przejść $P(s'|s,a)$,
- funkcja nagrody $R(s)$,

1. Model sensoryczny w ogólności może przybrać formę $P(e|s,a,s')$

POMDP

POMDP:

- **zbiór stanów** S,
- **zbiór akcji** A,
- **model przejść** $P(s'|s, a)$,
- **funkcja nagrody** $R(s)$,
- **model sensoryczny** $P(e|s)$ czyli prawd., że agent notuje obserwację e będąc w stanie s,

POMDP

POMDP:
- zbiór stanów S,
- zbiór akcji A,
- model przejść $P(s'|s, a)$,
- funkcja nagrody $R(s)$,
- model sensoryczny $P(e|s)$ czyli prawd., że agent notuje obserwację e będąc w stanie s,

1. Model sensoryczny w ogólności może przybrać formę $P(e|s, a, s')$

POMDP

POMDP:

- **zbiór stanów** S,
- **zbiór akcji** A,
- **model przejść** $P(s'|s, a)$,
- **funkcja nagrody** $R(s)$,
- **model sensoryczny** $P(e|s)$ czyli prawd., że agent notuje obserwację e będąc w stanie s,
- **początkowy stan przekonań** b_0 (nieznany jest stan początkowy s_0).

POMDP

POMDP:
- zbiór stanów S,
- zbiór akcji A,
- model przejść $P(s'|s, a)$,
- funkcja nagrody $R(s)$,
- model sensoryczny $P(e|s)$ czyli prawd., że agent notuje obserwację e będąc w stanie s,
- początkowy stan przekonań b_0 (nieznany jest stan początkowy s_0).

1. Model sensoryczny w ogólności może przybrać formę $P(e|s, a, s')$

Stany przekonań

Aktualizacja stanu przekonań — **problem filtrowania** (estymacji stanu systemu). Aktualizacja stanów przekonań:

$$b'(s') = \alpha P(e|s') \sum_{s} P(s'|s,a) b(s),$$

gdzie:

- $b(s)$ oznacza prawd., że agent jest w stanie s wg stanu przekonań b,
- e jest poczynioną obserwacją.
- α jest współczynnikiem normalizującym $\left(\sum_{s \in S} b(s) = 1\right)$,

Stany przekonań

Aktualizacja stanu przekonań — **problem filtrowania** (estymacji stanu systemu). Aktualizacja stanów przekonań:

$$b'(s') = \alpha P(e|s') \sum_{s} P(s'|s,a) b(s),$$

gdzie:

- $b(s)$ oznacza prawd., że agent jest w stanie s wg stanu przekonań b,
- e jest poczynioną obserwacją.
- α jest współczynnikiem normalizującym $\left(\sum_{s \in S} b(s) = 1\right)$,

Stany przekonań

Twierdzenie (Astrom, 1965)

Optymalna polityka w POMDP jest funkcją $\pi : B \to A$, gdzie B jest zbiorem stanów przekonań). **Optymalna polityka nie zależy od aktualnego stanu, w którym jest agent.**

Stany przekonań

Twierdzenie (Astrom, 1965)

Optymalna polityka w POMDP jest funkcją $\pi : B \to A$, gdzie B jest zbiorem stanów przekonań). **Optymalna polityka nie zależy od aktualnego stanu, w którym jest agent.**

Stany przekonań

Twierdzenie (Astrom, 1965)

Optymalna polityka w POMDP jest funkcją $\pi : B \to A$, gdzie B jest zbiorem stanów przekonań). **Optymalna polityka nie zależy od aktualnego stanu, w którym jest agent.**

Jak zachowuje się agent realizujący politykę π?

Powtarza:

1. Wykonaj akcję $a = \pi(b)$, gdzie b jest aktualnym stanem przekonań agenta
2. Otrzymaj obserwację środowiska e
3. Zaktualizuj swój stan przekonań obliczając b' (filtrowanie)

Stany przekonań
Twierdzenie (Astrom, 1965)
Optymalna polityka w POMDP jest funkcją $\pi : B \to A$, gdzie B jest zbiorem stanów przekonań). **Optymalna polityka nie zależy od aktualnego stanu, w którym jest agent.**
Jak zachowuje się agent realizujący politykę π?
Powtarza:
1. Wykonaj akcję $a = \pi(b)$, gdzie b jest aktualnym stanem przekonań agenta
2. Otrzymaj obserwację środowiska e
3. Zaktualizuj swój stan przekonań obliczając b' (filtrowanie)

POMDP → MDP

Wniosek

Można przekształcić POMDP w MDP operujący w przestrzeni stanów przekonań, gdzie

- $P(b'|a, b)$ jest prawd., że nowym stanem przekonań będzie b' pod warunkiem, że aktualny stan przekonań to b i agent wykonuje akcję a.

$P(b'|a, b)$ można łatwo wyprowadzić.

POMDP → MDP
Wniosek
Można przekształcić POMDP w MDP operujący w przestrzeni stanów przekonań, gdzie
- $P(b'|a, b)$ jest prawd., że nowym stanem przekonań będzie b' pod warunkiem, że aktualny stan przekonań to b i agent wykonuje akcję a.
$P(b'|a, b)$ można łatwo wyprowadzić.

POMDP $\to$ MDP

Ale, zauważmy, że przestrzeń stanów przekonań B:

- jest przestrzenią ciągłą (rozkłady prawd.),
- ma bardzo wiele wymiarów.
  - Jeśli mamy n stanów $\implies$ b jest n-wymiarowym wektorem liczb rzeczywistych

POMDP → MDP
Ale, zauważmy, że przestrzeń stanów przekonań B:
jest przestrzenią ciągłą (rozkłady prawd.),
ma bardzo wiele wymiarów.
Jeśli mamy n stanów ⟹ b jest n-wymiarowym wektorem liczb rzeczywistych

POMDP $\rightarrow$ MDP

Ale, zauważmy, że przestrzeń stanów przekonań B:

- jest przestrzenią ciągłą (rozkłady prawd.),
- ma bardzo wiele wymiarów.
  - Jeśli mamy n stanów $\implies$ b jest n-wymiarowym wektorem liczb rzeczywistych

Istnieje **algorytm iteracji wartości** dla POMDP (1970), ale jest zbyt wolny nawet dla 4×3

- rozwiązywanie POMDP jest bardzo trudne obliczeniowo (PSPACE-trudne).

Istnieją algorytmy przybliżone oparte na **dynamicznych sieciach baysowskich**.

POMDP $\rightarrow$ MDP

Ale, zauważmy, że przestrzeń stanów przekonań B:

- jest przestrzenią ciągłą (rozkłady prawd.),
- ma bardzo wiele wymiarów.
  - Jeśli mamy n stanów $\implies$ b jest n-wymiarowym wektorem liczb rzeczywistych

Istnieje **algorytm iteracji wartości** dla POMDP (1970), ale jest zbyt wolny nawet dla 4×3

- rozwiązywanie POMDP jest bardzo trudne obliczeniowo (PSPACE-trudne).

Istnieją algorytmy przybliżone oparte na **dynamicznych sieciach baysowskich**.