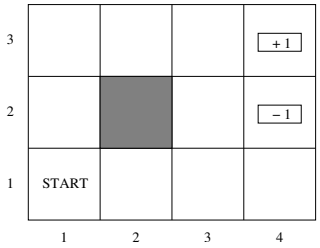
Sekwencyjne problemy decyzyjne

W **sekwencyjnych problemach decyzyjnych** użyteczność działań agenta nie zależy od pojedynczej decyzji, wyrażonej stanem, do którego ta decyzja doprowadziłaby agenta, ale raczej od całej sekwencji jego akcji.

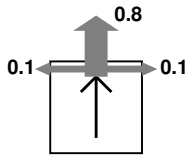
PRZYKŁAD: agent znajduje się w polu START, i może poruszać się we wszystkich kierunkach między kratkami. Jego działalność kończy się gdy osiągnie jedno z pól (4,2) lub (4,3), z wynikiem zaznaczonym w tych polach.



Gdyby zagadnienie było w pełni deterministyczne — i tym samym wiedza agenta o jego położeniu pełna — to problem sprowadzałby się do planowania działań. Na przykład, dla powyższego zagadnienia przykładowego dobrym rozwiązaniem byłby następujący plan działań: U-U-R-R-R. Ale równie dobry byłby plan: R-R-U-U-R. Jeśli w dodatku pojedyncze akcje nic nie kosztują (czyli liczy się tylko stan końcowy), to równie dobry jest nawet plan: R-R-R-L-L-L-U-U-R-R-R, i wiele innych.

Niepewność efektów działań agenta

Jednak po uwzględnieniu niepewności, wynik działań agenta jest zgodny z jego intencją tylko z pewnym prawdopodobieństwem. Na przykład, możemy przyjąć, że akcja U (Up) przenosi agenta na pożądaną pozycję „w górę" z prawdopodobieństwem 0.8, natomiast z prawdopodobieństwem 0.1 wykonuje ruch w lewo, i podobnie w prawo. Pewne jest tylko, że agent nie pójdzie w kierunku przeciwnym do zamierzonego. Aby uprościć analizę przyjmijmy dodatkowo, że obecność ścian nie zmienia tego rozkładu prawdopodobieństwa, a tylko spowoduje niewykonanie żadnego ruchu, gdyby „wypadło" ruszyć się w ścianę.

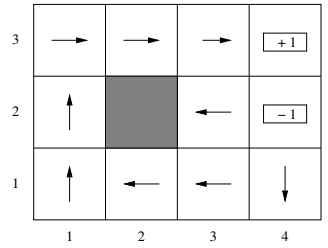


W tej sytuacji możemy obliczać wartości oczekiwane sekwencji ruchów agenta. Ogólnie agent nie może mieć pewności, że po wykonaniu dowolnej z powyższych sekwencji znajdzie się w pożądanym stanie terminalnym.

Polityka agenta

W odróżnieniu od algorytmów planowania działań, agent powinien wypracować swoją strategię nie w postaci konkretnej sekwencji działań, lecz w postaci **polityki**, czyli schematu wyznaczającego akcje, które powinny być podjęte dla każdego konkretnego stanu, gdyby agent w nim się znalazł.

Można określić optymalną politykę dla zagadnienia przykładowego. Zauważmy, że w punkcie (3,2) polityka nakazuje agentowi próbować ruchu w lewo, co pozornie nie ma sensu, ale pozwala agentowi ustrzec się przed wylądowaniem w niepożądanym stanie (4,2). Podobna sytuacja jest w stanie (4,1).

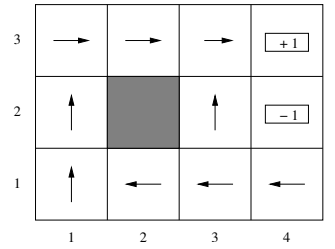


Taka polityka wynika oczywiście z domyślnego przyjęcia zerowego kosztu ruchów. Gdyby wynik agenta zależał nie tylko od stanu końcowego ale również od liczby wykonanych ruchów, wtedy nie opłacałoby mu się stosować tak konserwatywnej polityki.

Uwzględnienie kosztów ruchu

Przyjęcie niezerowego kosztu ruchu, pomniejsza wynik uzyskany w stanach końcowych o sumaryczny koszt wszystkich ruchów. Oczywiście wpływa to na kształt optymalnej polityki agenta.

Na przykład, diagram przedstawia optymalną politykę uwzględniającą koszty ruchu w wysokości $1/25$ jednostki. Zauważmy, że w stanach (4,1) i (3,2) polityka dyktuje teraz ruch bezpośrednio w kierunku stanu (4,3), pomimo ryzyka. Jednak w punktach (2,1) i (3,1) nadal zalecany jest ruch okrężny.



Formalnie, koszty ruchów wprowadza się w postaci funkcji **nagrody** dla stanów $R(s) = -0.04$, w tym przypadku nagrody o wartości ujemnej, czyli kary. Suma nagród dla sekwencji stanów wyznacza użyteczność tej sekwencji.

Problemy decyzyjne Markowa

Obliczanie polityki w postaci kompletnego odwzorowania stanów do zbioru akcji nazywane jest **problemem decyzyjnym Markowa (MDP)** jeśli prawdopodobieństwa przejść wynikające z podejmowanych akcji zależą tylko od bieżącego stanu, a nie np. od historii. Mówimy wtedy, że problem posiada **własność Markowa**.

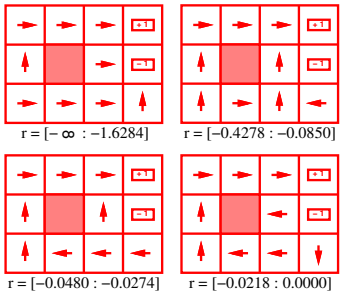
Formalnie, problem decyzyjny Markowa jest określony przez:

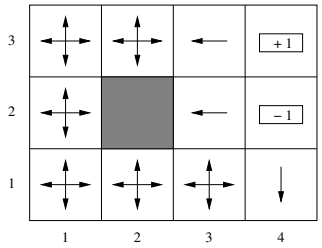
- zbiór stanów ze stanem początkowym s_0
- zbiór akcji $\textsc{Actions}(s)$ możliwych w stanie s
- model przejść $P(s'|s,a)$
- funkcję nagrody $R(s)$ (możliwe również: $R(s,a), R(s,a,s')$)

Rozwiązaniem MDP jest polityka $\pi(s)$ przyporządkowująca każdemu stanowi ruch. Zauważmy, że w warunkach niepewności, każde podjęcie działania przez agenta zgodne z pewną polityką może skończyć się inną ścieżką działań, i innym wynikiem. **Optymalną polityką** $\pi^*(s)$ nazywamy politykę osiągającą najwyższą oczekiwaną użyteczność.

Wpływ funkcji nagrody na politykę agenta

Zmienianie wartości nagrody dla stanów powoduje zmianę optymalnej polityki dla zagadnienia. Przy bardzo dużych negatywnych nagrodach (wysokich karach) zalecane jest jak najszybsze podążanie do stanu końcowego, obojętnie którego. Przy zbliżaniu się nagrody do zera powraca pierwotna „rozrzutna" polityka.





W przypadku dodatnich wartości nagrody agentowi przestaje się opłacać w ogóle zmierzać w kierunku rozwiązania. Działanie przynosi zyski, więc należy działać, a nie kończyć, zatem agent unika stanów terminalnych.

Problem horyzontu

W problemach MDP stany nie posiadają użyteczności, z wyjątkiem stanów końcowych. Możemy jednak mówić o użyteczności sekwencji (historii) stanów $U_h([s_0, s_1, ..., s_n])$, jeśli odpowiada ona zastosowanej sekwencji akcji, i prowadzi do stanu końcowego. Jest ona wtedy równa uzyskanemu wynikowi końcowemu.

Poprzednio zdefiniowaliśmy optymalną politykę na podstawie oczekiwanej użyteczności sekwencji stanów. Jednak wyznaczenie optymalnej polityki zależy od istotnej kwestii: czy mamy do dyspozycji nieskończony **horyzont** czasowy, czy też horyzont ograniczony do jakiejś skończonej liczby kroków? W tym drugim przypadku konkretna wartość horyzontu może wpływać na kształt polityki optymalnej. W takich przypadkach mówimy, że optymalna polityka jest **niestacjonarna**. Dla problemów z nieskończonym horyzontem polityka optymalna jest stacjonarna.

Obliczanie optymalnej polityki przy skończonych horyzontach jest trudniejsze, i na razie będziemy rozważali zagadnienia z horyzontem nieskończonym.

Dyskontowanie

Jak pokazuje rozważany wcześniej przykład, nieskończone sekwencje akcji mogą się zdarzać, a nawet mogą stanowić optymalną politykę agenta. Rozważanie nieskończonych, albo choćby bardzo długich, sekwencji jest czasami konieczne, np. gdy zagadnienie nie posiada stanów terminalnych, albo gdy agent może ich nie osiągnąć. Jednak takie obliczenia są kłopotliwe, ponieważ sumy nagród osiągają wtedy nieskończone wartości, które trudno jest porównywać.

Jako jedno z rozwiązań tego problemu stosuje się technikę zwaną **dyskontowaniem** (*discounting*) polegającą na efektywnym zmniejszeniu wkładu przyszłych nagród do użyteczności za pomocą współczynnika $0 < \gamma < 1$. Użyteczność sekwencji stanów H definiujemy jako $U(H) = \Sigma_i \gamma^i R_i$, czyli:

$$U_h([s_0, s_1, ..., s_n]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + ... + \gamma^n R(s_n)$$

Dla $\gamma < 1$ i $R \leq R_{max}$ tak zefiniowane użyteczności są zawsze skończone.

Technika dyskontowania ma swoje intuicyjne uzasadnienie w wielu dziedzinach życia. Odzwierciedla ona mniejsze znaczenie nagród w odległej przyszłości. Podobnie, w ekonomii stosuje się dyskontowanie w ocenie wartości inwestycji.

Polityki właściwe i uśrednianie

W przypadku nieskończonych sekwencji ruchów istnieją jeszcze inne możliwe podejścia poza dyskontowaniem. Na przykład, jako użyteczność sekwencji można przyjąć **średnią nagrodę** obliczaną na jeden krok.

Z kolei, jeśli zagadnienie posiada stany terminalne, to możliwe jest wyznaczenie polityki, która gwarantuje doprowadzenie agenta do jednego z tych stanów. Wtedy rozważanie użyteczności sekwencji nieskończonych nie wchodzi w grę. Polityki gwarantujące doprowadzenia agenta do któregoś ze stanów terminalnych nazywamy politykami **właściwymi**.

Własności użyteczności sekwencji stanów

Funkcję użyteczności sekwencji stanów nazywamy **separowalną** jeśli:

$$U([s_0, s_1, ..., s_n]) = f(s_0, U([s_1, ..., s_n]))$$

Zauważmy, że dla naszego przykładowego zagadnienia 4×3 funkcja użyteczności jest separowalna, ponieważ można ją obliczać z wzoru:

$$U_h([s_0, s_1, ..., s_n]) = R(s_0) + R(s_1) + \cdots + R(s_n)$$

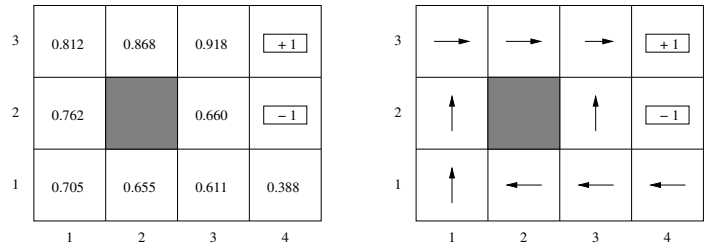
Mówimy, że funkcja użyteczności sekwencji stanów jest **addytywna**, gdy posiada następującą własność:

$$U_h([s_0, s_1, ..., s_n]) = R(s_0) + U_h([s_1, ..., s_n])$$

Okazuje się, że w wielu zagadnieniach praktycznych funkcje użyteczności są addytywne. Na przykład, rozważając funkcje kosztu w zagadnieniach przeszukiwania, domyślnie zakładaliśmy, że są one addytywne. Addytywność oznaczała tam, że poniesione koszty po prostu się sumują.

Obliczanie optymalnej polityki — użyteczności stanów

Do wyznaczania polityki optymalnej przydałyby się użyteczności stanów (np. takie jak na diagramie po lewej, jednak nie pytajmy na razie skąd się wzięły). Moglibyśmy wtedy posłużyć się zasadą MEU (maksymalnej oczekiwanej użyteczności), i dla każdego stanu wyznaczyć ruch, który maksymalizuje oczekiwaną użyteczność.



Jednak w zagadnieniach MDP stany jako takie nie mają obiektywnych użyteczności! „Użyteczność" stanu zależy od polityki agenta, od tego co zamierza on w danym stanie zrobić. Jednocześnie polityka agenta zależy od „użyteczności" stanów.

Użyteczność stanów można więc wprowadzić na podstawie polityki.

Użyteczności stanów

Użyteczność stanu ze względu na daną politykę można zdefiniować jako wartość oczekiwaną nagród uzyskanych przez działanie zaczynające się w tym stanie:

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

Przez S_t oznaczamy tu zmienną losową oznaczającą stan w jakim agent znajdzie się w kroku t po wystartowaniu ze stanu s i realizowaniu polityki π.

Okazuje się, że pomimo iż teoretycznie polityka optymalna $\pi^* = \underset{\pi}{\operatorname{argmax}}\, U^\pi(s)$ zależy od wyboru stanu początkowego, to dla procesów decyzyjnych posiadających własność Markowa, dla nieskończonych sekwencji i przy zastosowaniu dyskontowania, nie ma tej zależności. Polityka optymalna wyznaczająca drogę agenta jest taka sama niezależnie od punktu startowego.

Jako użyteczność stanów $U(s)$ będziemy więc przyjmować tak właśnie określoną użyteczność tego stanu ze względu na politykę optymalną $U^{\pi^*}(s)$.

Programowanie dynamiczne

Optymalną politykę π^* jako funkcję określoną na zbiorze stanów można związać z funkcją użyteczności stanów (jeszcze nieznaną):

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

gdzie $P(s'|s,a)$ jest prawdopodobieństwem, że agent osiągnie stan s' jeśli znajdzie się w stanie s i zastosuje akcję a.

Ponieważ użyteczność stanu chcemy określić jako wartość oczekiwaną dyskontowanej sumy nagród sekwencji stanów, zatem można ją związać z użytecznościami stanów sąsiednich następującym równaniem (Bellman 1957):

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

Dla n stanów uzyskujemy wtedy n równań — niestety nieliniowych ze względu na obecność operatora $\max$ — z n niewiadomymi. Rozwiązywanie tego równania nazywane jest **programowaniem dynamicznym**.

n-krokowe problemy decyzyjne

Gdyby w jakimś zagadnieniu stany końcowe były osiągane ze znanymi użytecznościami po dokładnie n krokach, wtedy możma z równania Bellmana najpierw wyznaczyć użyteczności stanów w kroku $n-1$, potem w kroku $n-2$, itd., aż do stanu początkowego. Zagadnienie tego typu nazywane jest n-krokowym problemem decyzyjnym, i znalezienie jego rozwiązania jest stosunkowo proste.

Niestety, w większości zagadnień praktycznych nie możemy zakładać stałej, n-krokowej sekwencji kroków, np. ze względu na pojawianie się pętli.

Algorytm iteracji wartości

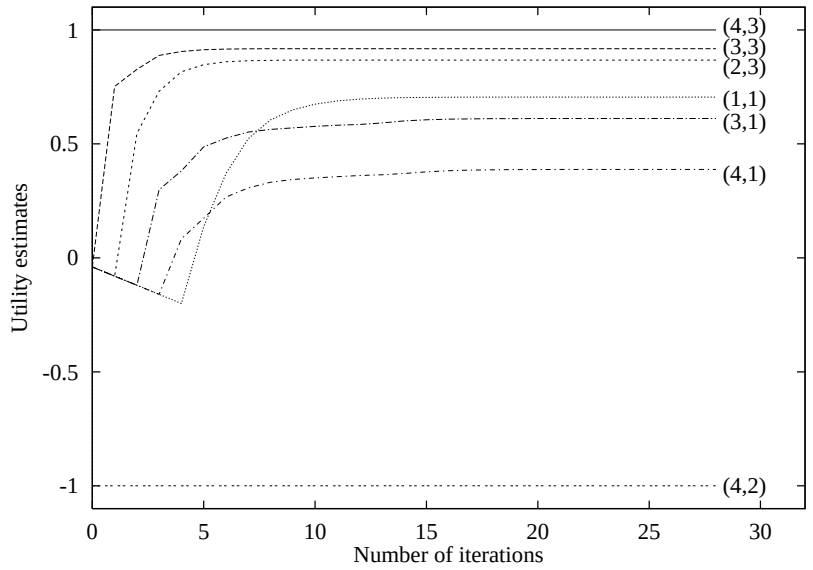
Dla zagadnień, których nie można przedstawić w postaci powyższego n-krokowego problemu decyzyjnego, można obliczyć przybliżone wartości użyteczności stanów w procesie iteracyjnym zwanym **iteracją wartości**:

$$U_{t+1}(s) = R(s) + \gamma \max_a \sum_{s'} P(s'|s,a)U_t(s')$$

W kroku $(t=0)$ przyjmujemy dowolne wartości użyteczności wszystkich stanów, i w kolejnych krokach algorytmu obliczamy kolejne ich przybliżenia.

Algorytm można zatrzymać porównując kolejne wartości użyteczności stanów, i oszacowując w ten sposób błąd. Optymalna polityka może być wyznaczona przez przybliżone wartości użyteczności, nawet przed ich zbiegnięciem się.

Algorytm iteracji wartości — przykład



Zbieżność algorytmu iteracji wartości

W rozważanym przykładzie procedura iteracji wartości przykładnie zbiegła się we wszystkich stanach. Pytanie jednak czy można liczyć, że tak będzie zawsze?

Okazuje się, że tak. Algorytm iteracji wartości zawsze doprowadzi do osiągnięcia ustalonych wartości użyteczności stanów, które są jedynym rozwiązaniem równania Bellmana. Można określić liczbę iteracji algorytmu niezbędną do osiągnięcia dowolnie określonego błędu ϵ, gdzie $R_{\max}$ jest górnym ograniczeniem wartości nagrody:

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

Zbieżność algorytmu iteracji wartości — uwagi

- W praktyce w algorytmie iteracji wartości można stosować kryterium stopu: $||U_{i+1} - U_i|| < \epsilon(1-\gamma)/\gamma$
- W praktyce optymalną politykę algorytm wyznacza istotnie wcześniej, niż wartości użyteczności ustabilizują się z małymi błędami.
- N rośnie w nieograniczony sposób, gdy γ zbliża sie do jedynki. Można przyspieszyć zbieżność zmniejszając γ, ale to oznacza skrócenie horyzontu agenta i zaniedbanie efektów długofalowych.
- Dla $\gamma = 1$ jeśli w zagadnieniu istnieją stany terminalne, można wyprowadzić podobne do powyższych kryteria zbieżności i błędów.

Algorytm iteracji polityki

Ponieważ często optymalna polityka jest względnie nieczuła na konkretne wartości funkcji użyteczności, można ją obliczać innym procesem iteracyjnym, zwanym **iteracją polityki**. Polega ona na wyborze dowolnej polityki początkowej π_0, a następnie cyklicznym, naprzemiennym, obliczaniu kolejnych przybliżeń uaktualnionych użyteczności, zgodnie z poniższym wzorem:

$$U_{t+1}(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi_t(s)) U_{t+1}(s')$$

oraz uaktualnionej polityki, zgodnie z wzorem:

$$\pi_{t+1}(s) = \operatorname*{argmax}_a \sum_{s'} P(s'|s,a) U_t(s')$$

W powyższych wzorach $\pi_t(s)$ oznacza akcję wyznaczoną przez aktualną politykę π_t dla stanu s. Zauważmy, że pierwszy wzór generuje układ równań liniowych, które można rozwiązać dokładnie ze względu na U_{t+1} (są to dokładne wartości użyteczności dla aktualnej przybliżonej polityki) w czasie $O(n^3)$.

Algorytm iteracji polityki (cd.)

Algorytm iteracji polityki zatrzymuje się, gdy krok aktualizacji polityki nic już nie zmienia. Ponieważ dla skończonej przestrzeni istnieje skończona liczba polityk, zatem algorytm na pewno zatrzyma się.

Dla małych przestrzeni stanów (n w $O(n^3)$) powyższa procedura jest często najefektywniejsza. Jednak dla większych przestrzeni czynnik $O(n^3)$ powoduje znaczne spowolnienie procesu. Można wtedy stosować **zmodyfikowaną iterację polityki** polegającą na iteracyjnej aktualizacji wartości użyteczności — zamiast ich każdorazowego dokładnego wyznaczania — z wykorzystaniem uproszczonej aktualizacji Bellmana zgodnie z wzorem:

$$U_{t+1}(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi_t(s)) U_t(s')$$

W porównaniu z oryginalnym równaniem Bellmana pominięte tu zostało obliczanie optymalnej akcji, ponieważ tutaj akcje wyznacza aktualna polityka. Tym samym obliczenie to jest prostsze, i można wykonać kilka kroków takich aktualizacji przed kolejnym krokiem iteracji polityki (czyli aktualizacją polityki).

Brak pewnej informacji o stanie — POMDP

W ogólnym przypadku agent może nie być w stanie stwierdzić w jakim stanie znalazł się po wykonaniu akcji, a raczej może to stwierdzić z pewnym prawdopodobieństwem. Takie zagadnienia nazywamy **częściowo obserwowalnymi problemami decyzyjnymi Markowa (POMDP)**. W tych problemach agent musi obliczać oczekiwaną użyteczność swoich akcji biorąc pod uwagę różne możliwe ich wyniki, jak również różne możliwe nowe informacje (nadal niekompletne), które może uzyskać, w zależności od tego w jakim stanie się znajdzie.

Rozwiązanie problemu decyzyjnego można otrzymać obliczając rozkład prawdopodobieństwa po wszystkich możliwych stanach, w których agent może się potencjalnie znajdować, uwzględniając niepewną informację o otoczeniu jaką udało mu się zgromadzić. Jednak w ogólnym przypadku obliczenie to jest utrudnione ze względu na fakt, że podjęcie danej akcji spowoduje otrzymanie przez agenta jakichś nowych informacji, które mogą zmienić jego posiadaną wiedzę w sposób trudny do uwzględnienia. Praktycznie agent musi brać pod uwagę nowe informacje, jakie może otrzymać, na równi ze stanami, do których może trafić. Pojawia się tu ponownie kwestia wartości informacji rozważana wcześniej.

POMDP — formalizacja

Zagadnienie POMDP jest zdefiniowane przez następujące elementy:

- zbiór stanów, jednak bez stanu początkowego s_0,
- zbiór akcji $\textsc{Actions}(s)$ możliwych w stanie s,
- funkcję przejść: $P(s'|s,a)$ — rozkład prawdopodobieństw przejścia do stanu s' po wykonaniu akcji a w stanie s,
- funkcja nagrody: $R(s)$,
- model czujników: $P(e|s)$ — rozkład prawdopodobieństw uzyskania obserwacji e (*evidence*), częściowo błędnej, w stanie s,
- początkowy stan przekonań: b_0.

W zagadnieniach POMDP brak jest założenia o znajomości stanu początkowego. Zamiast tego, wprowadza się **stan przekonań** agenta $b(s)$ (*belief state*), który jest rozkładem prawdopodobieństw, że agent jest w pewnym stanie s. W chwili początkowej znamy jedynie początkowy stan przekonań b_0.

Zadaniem jest obliczenie polityki, generującej sekwencję ruchów o maksymalnej użyteczności. Oczywiście, w trakcie wykonywania tej sekwencji agent będzie zmieniał swój stan przekonań, tak ze względu na wykonywanie akcji, jak również na otrzymywane w ich wyniku obserwacje.

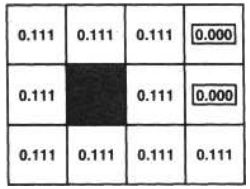
przykładowy stan przekonań b_0:

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	0
$\frac{1}{9}$	$\times$	$\frac{1}{9}$	0
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

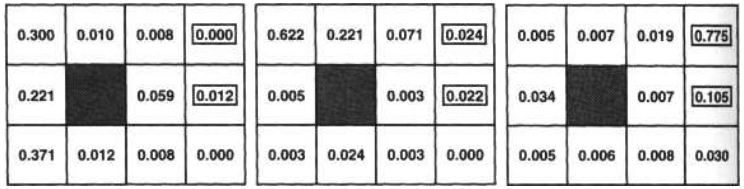
POMDP — przykład

Rozważmy ponownie przykład agenta w środowisku 4x3, jednak tym razem agent nie wie w jakim stanie początkowym się znajduje, i musi przyjąć równe prawdopodobieństwo $\frac{1}{9}$ znajdowania się w każdym z nieterminalnych stanów.

Jaka może być teraz optymalna polityka?



Rysunki poniżej przedstawiają kolejne rozkłady prawdopodobieństw położenia agenta po wykonaniu przez niego kolejno po pięciu ruchów: w lewo, w górę, i w prawo. Jest to polityka niezwykle ostrożna i konserwatywna, i przy tym rozrzutna. Jakkolwiek agent z prawdopodobieństwem 0.775 znajdzie się w „dobrym" stanie terminalnym, to oczekiwana użyteczność tej sekwencji wynosi tylko 0.08.



POMDP — obliczanie stanów przekonań agenta

W danym momencie agent, znajdując się w pewnym stanie przekonań b, który jest rozkładem prawdopodobieństwa znajdowania się w każdym z możliwych stanów s, musi wykonać jakąś akcję a. Po wykonaniu tej akcji agent otrzymuje obserwację e i na podstawie tej obserwacji, w połączeniu z poprzednim stanem przekonań i znajomością reguł rządzących światem (czyli rozkładu $P(s'|s,a)$) agent może obliczać nowy stan przekonań $b'(s')$ według wzoru:

$$b'(s') = \alpha P(e|s') \sum_s P(s'|s,a)b(s)$$

gdzie $P(e|s')$ określa prawdopodobieństwo otrzymania obserwacji e w stanie s', a α jest pomocniczą stałą normalizującą sumę stanów przekonań do 1.

Jest to operacja **filtracji** integrująca informacje otrzymywane z różnych źródeł.

Rozwiązywanie problemów POMDP

Kluczem do rozwiązania POMPD jest zrozumienie, że wybór optymalnej akcji zależy tylko od aktualnego stanu przekonań agenta. Ponieważ agent nie zna swojego stanu (i tak naprawdę nigdy go nie pozna), zatem jego optymalna polityka musi być odwzorowaniem $\pi^*(b)$ stanów przekonań na akcje. Jednak wartości b są wektorami prawdopodobieństw, czyli liczb zmiennoprzecinkowych, zatem taka polityka jest złożoną funkcją, której reprezentacja może nie być łatwa do wyrażenia.

Cykl roboczy agenta POMDP, zakładając, że ma już obliczoną kompletną politykę optymalną $\pi^*(b)$, jest więc następujący:

1. Dla bieżącego stanu przekonań b, wykonaj akcję $\pi^*(b)$.
2. Odbierz obserwację e.
3. Przejdź do stanu przekonań $b'(s')$, i potwórz cykl.

Przestrzeń stanów przekonań

W problemach POMDP agent porusza się w niedeterministycznej przestrzeni przekonań. Ponieważ model MDP uwzględnia niedeterministyczne ruchy agenta i pozwala rozwiązywać takie zagadnienia, problemy POMDP można przekształcać na równoważne problemy MDP określone w przestrzeni przekonań. W tej przestrzeni operujemy na rozkładzie prawdopodobieństw osiągnięcia przez agenta zbioru przekonań b' gdy obecnie posiada on zbiór przekonań b i wykona akcję a. Dla zagadnienia o n stanach, b są n-elementowymi wektorami o wartościach rzeczywistych.

Zauważmy, że przestrzeń stanów przekonań, do której zaprowadziło nas rozważanie zagadnień POMDP, jest przestrzenią ciągłą, w odróżnieniu od oryginalnego zagadnienia. Ponadto typowo jest to przestrzeń wielowymiarowa. Na przykład, dla świata 4×3 z poprzedniego przykładu, będzie to przestrzeń 11-wymiarowa.

Przedstawione wcześniej algorytmy iteracji wartości i iteracji polityki nie nadają się do rozwiązywania takich zagadnień. Ich rozwiązywanie jest ogólnie bardzo trudne obliczeniowo (PSPACE-trudne).

Konwersja POMDP na MDP

W problemach POMDP dodatkowym elementem niedeterminizmu są obserwacje. Agent nie wie, jaką obserwację otrzyma po wykonaniu akcji a w stanie przekonań b. Może ją jednak oszacować:

$$\begin{aligned} P(e|a,b) &= \sum_{s'} P(e|a,s',b)P(s'|a,b) \\ &= \sum_{s'} P(e|s')P(s'|a,b) \\ &= \sum_{s'} P(e|s') \sum_s P(s'|s,a)b(s) \end{aligned}$$

Teraz możemy teraz wyznaczyć funkcję przejść dla stanów przekonań:

$$\begin{aligned} P(b'|b,a) &= P(b'|a,b) = \sum_e P(b'|e,a,b)P(e|a,b) \\ &= \sum_e P(b'|e,a,b) \sum_{s'} P(e|s') \sum_s P(s'|s,a)b(s) \end{aligned}$$

gdzie

$$P(b'|e,a,b) = \begin{cases} 1 & \text{gdy } b'(s') = \alpha P(e|s') \Sigma_s P(s'|s,a)b(s) \\ 0 & \text{w przeciwnym wypadku} \end{cases}$$

Jedyną pozostałą niewiadomą jest wartość nagrody ρ procesu Markowa otrzymanego z POMDP, którą również można wyznaczyć na podstawie wartości nagród R oryginalnego POMDP, w zależności od jej postaci:

$$\begin{aligned} \rho(b) &= \sum_s b(s)R(s) \\ \rho(b,a) &= \sum_s b(s) \sum_{s'} P(s'|s,a)R(s,a,s') \end{aligned}$$

Powyżej zdefiniowane elementy składają się na całkowicie obserwowalny proces Markowa (MDP) na przestrzeni stanów przekonań. Pamiętajmy wszakże, że stany przekonań agenta są w pełni obserwowalne.

Można udowodnić, że optymalna polityka $\pi^*(b)$ dla tego MDP jest jednocześnie optymalną polityką dla oryginalnego zagadnienia POMDP.

Obliczanie optymalnej polityki POMDP

Schemat algorytmu: definiujemy politykę $\pi(b)$ dla regionów przestrzeni przekonań, gdzie dla jednego regionu polityka wyznacza jedną akcję. Następnie proces iteracyjny podobny do algorytmów iteracji wartości czy iteracji polityki aktualizuje granice regionów, i może wprowadzać nowe regiony.

Obliczona tym algorytmem optymalna polityka agenta dla powyższego przykładu daje następującą sekwencję akcji:

[L, U, U, R, U, U, (R, U, U)*]

(cyklicznie powtarzająca się nieskończona sekwencja R-U-U jest konieczna ze względu na niepewność osiągnięcia stanu terminalnego). Agent osiągnie pożądany stan docelowy z prawdopodobieństwem 0.866, a oczekiwana wartość użyteczności tego rozwiązania wynosi 0.38, czyli istotnie lepiej niż dla pierwotnie zaproponowanej naiwnej polityki (0.08).