Wprowadzenie do uczenia ze wzmocnieniem

część 2

Strategie i funkcje wartości

W przypadku każdej strategii π i dowolnego stanu s następująca zależność zachodzi między wartością stanu s i wartością jego możliwego następcy s'.

$$
\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r | s, a) \Big[r + \gamma \mathbb{E}_\pi[G_{t+1} | S_{t+1} = s'] \Big] \\
&= \sum_a \pi(a|s) \sum_{s', r} p(s', r | s, a) \Big[r + \gamma v_\pi(s') \Big]
\end{aligned}
$$

dla każdego $s \in S$.

Jest to tzw. równanie Bellmana dla funkcji stanu $v_\pi(s)$.

Strategie i funkcje wartości

Równanie Bellmana

$$v_\pi(s) \doteq \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\Big[r + \gamma v_\pi(s')\Big]$$

Wyrażenie po prawej stronie może być traktowane jako wartość oczekiwana. Jest to suma po wszystkich wartościach trzech zmiennych: a, s', r.

Dla każdej takiej trójki obliczmy prawdopodobieństwo:

$$\pi(a|s)p(s',r|s,a)$$

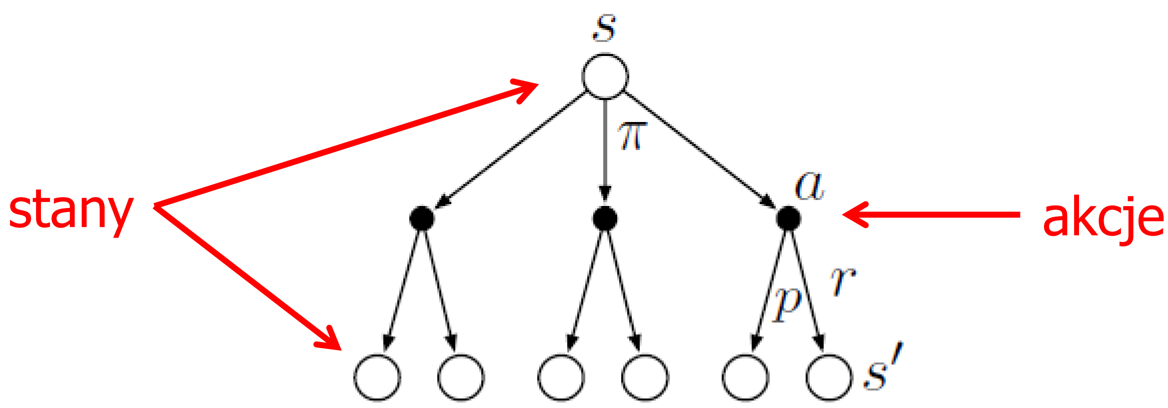
(są to wagi przez które przemnażamy wyrażenie w nawiasie i następnie sumujemy po wszystkich prawdopodobieństwach)

Strategie i funkcje wartości

Równanie Bellmana:

$$v_\pi(s) \doteq \sum_{a,s',r} \pi(a|s)\, p(s',r\,|\,s,a)\Big[r + \gamma v_\pi(s')\Big]$$

Backup diagram dla równania Bellmana:



Optymalne strategie i funkcje wartości

Rozwiązanie problemu uczenia się przez wzmacnianie polega na znalezieniu polityki (strategii) π, która zapewni najwyższą nagrodę na dłuższą metę.

W przypadku skończonych MDP możemy precyzyjnie określić optymalną politykę.

Mówimy, że polityka π jest lepsza lub równa polityce π' jeżeli oczekiwany zwrot dla polityki π jest większy lub równy oczekiwanemu zwrotowi dla polityki π'.

Formalnie:

$$\pi \geq \pi' \iff v_\pi(s) \geq v_{\pi'}(s)$$

Optymalne strategie i funkcje wartości

Zawsze istnieje przynajmniej jedna polityka, która jest lepsza lub równa każdej innej polityce. To tzw. polityka optymalna.

Oczywiście może istnieć więcej niż jedna polityka optymalna. Wszystkie takie polityki oznaczamy przez π_*.

Wszystkie polityki optymalne mają taką samą tzw. optymalną funkcję stanu zdefiniowaną następująco:

$$v_*(s) \doteq \max_{\pi} v_\pi(s)$$

dla każdego $s \in S$.

Optymalne strategie i funkcje wartości

Ponieważ v_* jest funkcją wartości stanu zatem musi spełniać równanie Bellmana.

Ponieważ jest to funkcja związana z optymalną polityką zatem w równaniu tym nie może się pojawić żadna polityka.

Wyprowadźmy równanie optymalizacyjne Bellmana dla funkcji $v_*(s)$:

$$\begin{aligned}
v_*(s) &= \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s,a) \\
&= \max_a \mathbb{E}_{\pi_*}[G_t \mid S_t = s, A_t = a] \\
&= \max_a \mathbb{E}_{\pi_*}[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a] \\
&= \max_a \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a] \\
&= \max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma v_*(s')\big].
\end{aligned}$$

Optymalne strategie i funkcje wartości

Równanie optymalizacyjne Bellmana dla funkcji $v_*(s)$:

$$v_*(s) = \max_a \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma v_*(s')\right]$$

Wyprowadźmy równanie optymalizacyjne Bellmana dla funkcji $q_*(s)$:

$$\begin{aligned} q_*(s,a) &= \mathbb{E}\left[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') \,\middle|\, S_t = s, A_t = a\right] \\ &= \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma \max_{a'} q_*(s', a')\right]. \end{aligned}$$

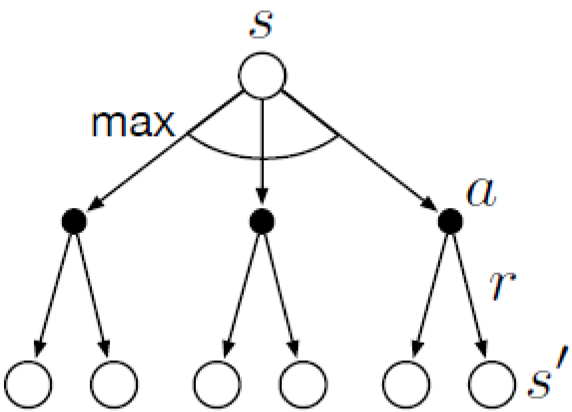
Jak wygląda backup diagram dla optymalizacyjnego równania Bellmana?

Optymalne strategie i funkcje wartości

Równanie optymalizacyjne Bellmana dla funkcji $v_*(s)$:

$$v_*(s) = \max_a \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma v_*(s')\right]$$

Backup diagram:

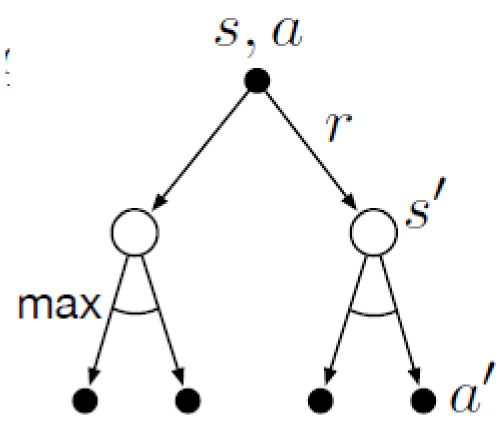


Optymalne strategie i funkcje wartości

Równanie optymalizacyjne Bellmana dla funkcji $q_*(s, a)$:

$$q_*(s,a) = \sum_{s',r} p(s',r|s,a)\Big[r + \gamma \max_{a'} q_*(s',a')\Big]$$

Backup diagram:



Optymalne strategie i funkcje wartości

$$v_*(s) = \max_a \sum_{s',r} p(s', r | s, a) [r + \gamma v_*(s')]$$

Dla skończonych MDP równanie optymalizacyjne Bellmana dla funkcji $v_*(s)$ ma unikalne rozwiązanie.

Równanie optymalizacyjne Bellmana jest w rzeczywistości układem równań, po jednym dla każdego stanu. Jeśli jest n stanów, to mamy n równań z n niewiadomymi.

Jeśli dynamika p środowiska jest znana, to w zasadzie można rozwiązać ten układ równań dla $v_*(s)$ przy użyciu dowolnej z wielu metod rozwiązywania układów równań nieliniowych. Można też rozwiązać układ równań dla $q_*(s, a)$.

Optymalne strategie i funkcje wartości

$$v_*(s) = \max_a \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_*(s')\right]$$

Załóżmy, że mamy rozwiązanie równania optymalizacyjnego Bellmana $v_*(s)$.

Jak wyznaczyć optymalną strategię (politykę)?

Dla każdego stanu s, będzie jedna lub więcej akcji, w których osiągane będzie maksimum wyliczone z równania optymalizacyjnego Bellmana.

Każda polityka, która przypisuje niezerowe prawdopodobieństwo tylko do tych akcji, jest polityką optymalną.

Optymalne strategie i funkcje wartości

$$v_*(s) = \max_a \sum_{s',r} p(s', r | s, a) [r + \gamma v_*(s')]$$

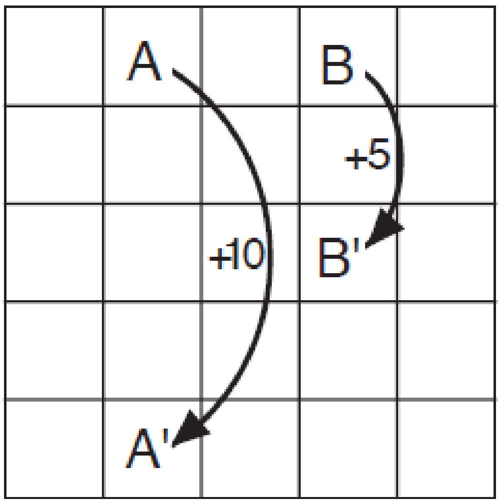
Inaczej: jeżeli mamy funkcję $v_*(s)$ wówczas akcje, które wydają się najlepsze po jednym kroku wyszukiwaniu, będą optymalne.

Innym sposobem powiedzenia tego jest stwierdzenie, że każda polityka zachłanna w odniesieniu do optymalnej funkcji oceny $v_*(s)$ jest optymalną polityką.

W przypadku funkcji $q_*(s, a)$ wybieramy po prostu akcję a dla której wartość funkcji q_* jest największa.

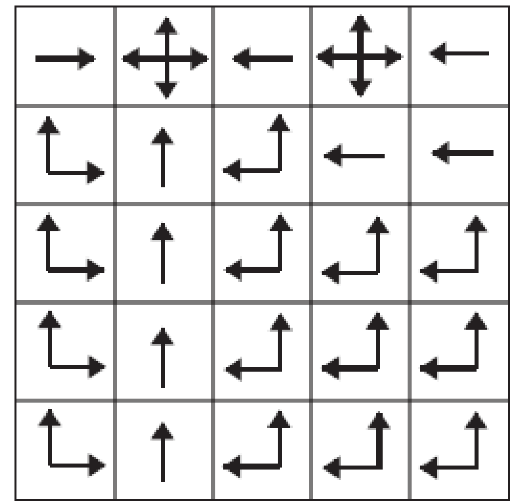
Strategie i funkcje wartości

Przykład (Gridworld)



22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

v_*



π_*

Za pomocą $v_*(s)$ optymalny oczekiwany długoterminowy zwrot jest przekształcony w wielkość, która jest lokalnie i natychmiast dostępna dla każdego stanu.

MDP - przykład

Przykład

Robot sprzątający puste puszki.

Stan opisuje poziom naładowania baterii:

$$S = \{\mathtt{high}, \mathtt{low}\}$$

Akcje: search, wait, recharge

$$A(\mathtt{low}) = \{\mathtt{search}, \mathtt{wait}, \mathtt{recharge}\}$$

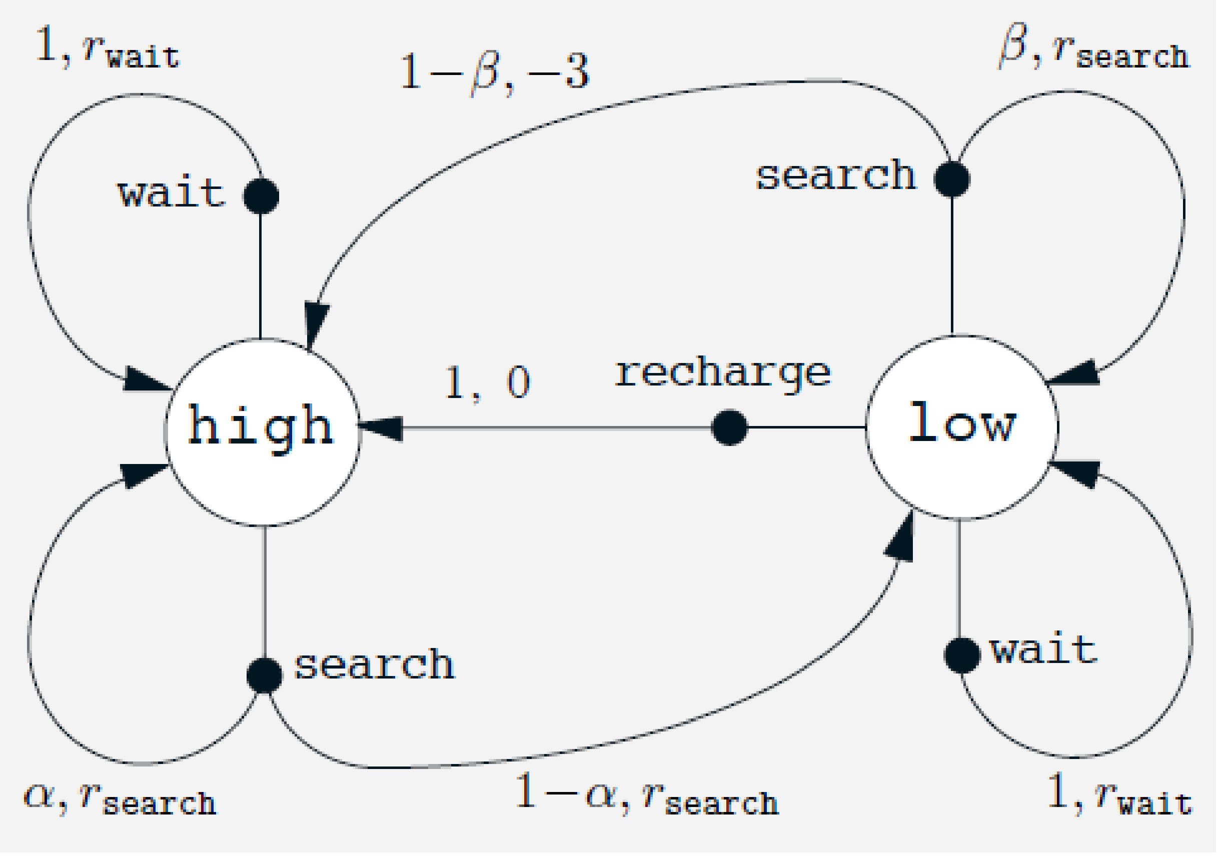
$$A(\mathtt{high}) = \{\mathtt{search}, \mathtt{wait}\}$$

Nagrody: każda zebrana puszka +1

$$r_{\mathtt{search}} > r_{\mathtt{wait}}, 0, -3$$

MDP - przykład

Robot sprzątający – graf przejścia:



MDP - przykład

Przykład

Mamy dwa stany: $S = \{\texttt{high}, \texttt{low}\}$

$$v_*(s) = \max_a \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_*(s')\right]$$

$$\begin{aligned}
&v_*(\texttt{h}) = \\
&= \max\left\{\begin{array}{l} p(\texttt{h}|\texttt{h},\texttt{s})[r(\texttt{h},\texttt{s},\texttt{h}) + \gamma v_*(\texttt{h})] + p(\texttt{l}|\texttt{h},\texttt{s})[r(\texttt{h},\texttt{s},\texttt{l}) + \gamma v_*(\texttt{l})], \\ p(\texttt{h}|\texttt{h},\texttt{w})[r(\texttt{h},\texttt{w},\texttt{h}) + \gamma v_*(\texttt{h})] + p(\texttt{l}|\texttt{h},\texttt{w})[r(\texttt{h},\texttt{w},\texttt{l}) + \gamma v_*(\texttt{l})] \end{array}\right\} \\
&= \max\left\{\begin{array}{l} \alpha[r_{\texttt{s}} + \gamma v_*(\texttt{h})] + (1-\alpha)[r_{\texttt{s}} + \gamma v_*(\texttt{l})], \\ 1[r_{\texttt{w}} + \gamma v_*(\texttt{h})] + 0[r_{\texttt{w}} + \gamma v_*(\texttt{l})] \end{array}\right\} \\
&= \max\left\{\begin{array}{l} r_{\texttt{s}} + \gamma[\alpha v_*(\texttt{h}) + (1-\alpha)v_*(\texttt{l})], \\ r_{\texttt{w}} + \gamma v_*(\texttt{h}) \end{array}\right\}.
\end{aligned}$$

MDP - przykład

Przykład

Mamy dwa stany: $S = \{\texttt{high}, \texttt{low}\}$

$$v_*(s) = \max_a \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_*(s')\right]$$

$$v_*(\texttt{l}) = \max \left\{ \begin{array}{l} \beta r_{\texttt{s}} - 3(1-\beta) + \gamma[(1-\beta)v_*(\texttt{h}) + \beta v_*(\texttt{l})], \\ r_{\texttt{w}} + \gamma v_*(\texttt{l}), \\ \gamma v_*(\texttt{h}) \end{array} \right\}.$$

Powyższy układ dwóch równań z dwiema niewiadomymi możemy rozwiązać, ze względu na $v_*(h)$ i $v_*(l)$.

Optymalne strategie i funkcje wartości

$$v_*(s) = \max_a \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_*(s')\right]$$

Rozwiązanie równania optymalizacyjnego Bellmana jest możliwe wtedy gdy spełnione są założenia:

- Znamy dynamikę p środowiska.
- Mamy zasoby pozwalające znaleźć rozwiązanie.
- Zachodzi własność Markowa.

Bardzo często wszystkie założenia te nie są spełnione!

Programowanie dynamiczne

Termin programowanie dynamiczne (DP) odnosi się do zbioru algorytmów, które można wykorzystać do obliczenia optymalnych polityk, przy założeniu, że mamy doskonały model środowiska jako proces decyzyjny Markowa (MDP).

Klasyczne algorytmy DP mają ograniczoną użyteczność w uczeniu przez wzmocnienie zarówno ze względu na założenie doskonałego modelu, jak i ze względu na ich wielki koszt obliczeniowy. Tym niemniej nadal są one teoretycznie ważne.

W rzeczywistości wszystkie metody RL mogą być postrzegane jako próby osiągnięcia tego samego efektu co DP, tylko przy mniejszej ilości obliczeń i bez założenia doskonałego modelu środowiska.

DP – obliczenie polityki

Najpierw zastanawiamy się, jak obliczyć wartość funkcji stanu v_π dla dowolnej polityki π.

Nazywamy to obliczeniem polityki (ang. policy evaluation). Inne określenie to problem przewidywania (ang. prediction problem). Wiemy już, że:

$$
\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a) \Big[r + \gamma \mathbb{E}_\pi[G_{t+1}|S_{t+1} = s'] \Big] \\
&= \sum_a \pi(a|s) \sum_{s', r} p(s', r|s, a) \Big[r + \gamma v_\pi(s') \Big]
\end{aligned}
$$

DP – obliczenie polityki

Wiemy już, że znalezienie v_π jest możliwe, chociaż może wymagać bardzo wielu obliczeń.

Dla naszych celów metody iteracyjne znajdowania rozwiązania wydają najbardziej odpowiednie.

Rozważmy sekwencję przybliżonych funkcji wartości stanów v_0, v_1, v_2,… gdzie każda funkcja $v_i : S^+ \rightarrow \mathbb{R}$.

Pierwsza aproksymacja – v_0, jest wybierana dowolnie (tylko dla stanu końcowego musi mieć wartość 0),

Jak możemy uzyskać kolejne aproksymacje?

DP – obliczenie polityki

Wykorzystujemy do tego równanie Bellmana:

$$
\begin{aligned}
v_{k+1}(s) &\doteq \mathbb{E}_\pi[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\Big[r + \gamma v_k(s')\Big]
\end{aligned}
$$

Jeżeli w ciągu funkcji $v_0, v_1, v_2, \ldots$ pojawi się funkcja v_π to będzie ona punktem stałym tego „odwzorowania" to znaczy:

$$
v_\pi(s) \doteq \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\Big[r + \gamma v_\pi(s')\Big]
$$

…bo v_π spełnia równanie Bellmana.

DP – obliczenie polityki

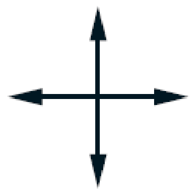
<u>Iteracyjne szacowanie polityki</u>

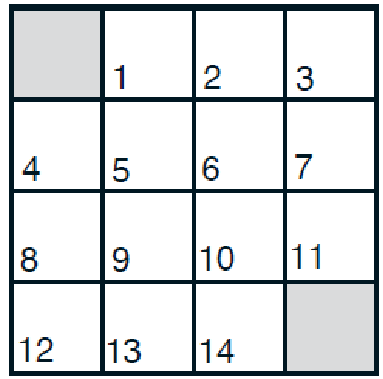
Dane: polityka do oceny π.
Parametry algorytmu: wartość progu $\theta > 0$ określającego dokładność oszacowania.
Ustalamy początkowe wartości $V(s)$ jako dowolne, dla wszystkich $s \in S^+$ z wyjątkiem $V(s_T)=0$.

Loop:
$\quad \Delta \leftarrow 0$
$\quad$ Loop for each $s \in \mathcal{S}$:
$\qquad v \leftarrow V(s)$
$\qquad V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\big[r + \gamma V(s')\big]$
$\qquad \Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

DP – obliczenie polityki

<u>Przykład</u>





$R_t = -1$

$\mathcal{S} = \{1, 2, \ldots, 14\} \qquad \mathcal{A} = \{\texttt{up}, \texttt{down}, \texttt{right}, \texttt{left}\}$

$$p(6, -1 \,|\, 5, \texttt{right}) = 1$$
$$p(7, -1 \,|\, 7, \texttt{right}) = 1$$
$$p(10, r \,|\, 5, \texttt{right}) = 0$$

Zakładamy, że $\pi(a|s)$=0.25 dla dowolnego $a \in A$.

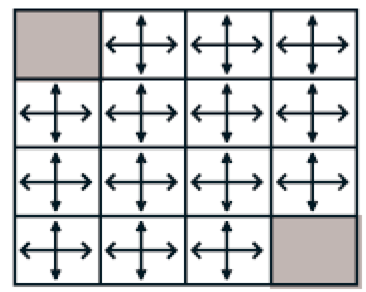
DP – obliczenie polityki

<u>Przykład</u>

Kolejne przybliżenia v_k dla losowej polityki:

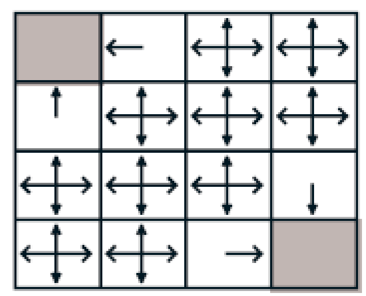
$k = 0$

0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0



$k = 1$

0.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	0.0



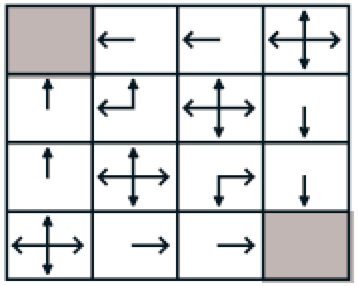
DP – obliczenie polityki

Przykład

Kolejne przybliżenia v_k dla losowej polityki:

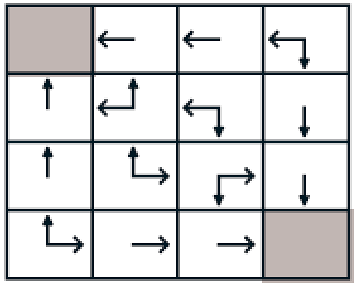
$k = 2$

0.0	-1.7	-2.0	-2.0
-1.7	-2.0	-2.0	-2.0
-2.0	-2.0	-2.0	-1.7
-2.0	-2.0	-1.7	0.0



$k = 3$

0.0	-2.4	-2.9	-3.0
-2.4	-2.9	-3.0	-2.9
-2.9	-3.0	-2.9	-2.4
-3.0	-2.9	-2.4	0.0



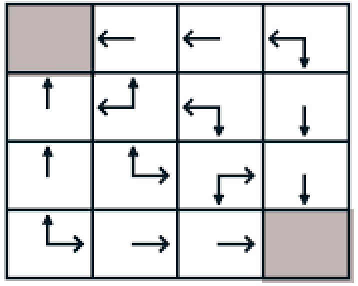
DP – obliczenie polityki

<u>Przykład</u>

… aż otrzymujemy polityką optymalną:

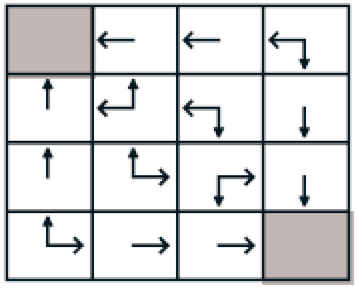
$k = 10$

0.0	-6.1	-8.4	-9.0
-6.1	-7.7	-8.4	-8.4
-8.4	-8.4	-7.7	-6.1
-9.0	-8.4	-6.1	0.0



$k = \infty$

0.0	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	0.0



DP – poprawa polityki

Załóżmy, że ustaliliśmy funkcję wartości v_π dla dowolnej polityki deterministycznej π.

Dla niektórych stanów chcielibyśmy wiedzieć, czy powinniśmy zmienić politykę i wybrać akcję $a \neq \pi(s)$?

Wiemy jak korzystne jest zastosowanie polityki π w stanie s – znamy wartość $v_\pi(s)$. Czy jednak nie byłaby korzystna zmiana polityki w stanie s i wybranie akcji $a \neq \pi(s)$???

Jak to można sprawdzić?

DP – poprawa polityki

Możemy wyliczyć:

$$
\begin{aligned}
q_\pi(s,a) &\doteq \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a] \\
&= \sum_{s',r} p(s',r|s,a)\Big[r + \gamma v_\pi(s')\Big].
\end{aligned}
$$

i porównać tę wartość z: $v_\pi(s)$

Twierdzenie o poprawie polityki

Niech π i π' będą dwiema politykami deterministycznymi takimi, że:

$$q_\pi(s,\pi'(s)) \geq v_\pi(s)$$

dla każdego $s \in S$. Wówczas:

$$v_{\pi'}(s) \geq v_\pi(s)$$

dla każdego $s \in S$.

DP – poprawa polityki

Dowód:

$$
\begin{aligned}
v_\pi(s) &\le q_\pi(s, \pi'(s)) \\
&= \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = \pi'(s)] \\
&= \mathbb{E}_{\pi'}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s] \\
&\le \mathbb{E}_{\pi'}[R_{t+1} + \gamma q_\pi(S_{t+1}, \pi'(S_{t+1})) \mid S_t = s] \\
&= \mathbb{E}_{\pi'}[R_{t+1} + \gamma \mathbb{E}_{\pi'}[R_{t+2} + \gamma v_\pi(S_{t+2}) | S_{t+1}, A_{t+1} = \pi'(S_{t+1})] \mid S_t = s] \\
&= \mathbb{E}_{\pi'}\big[R_{t+1} + \gamma R_{t+2} + \gamma^2 v_\pi(S_{t+2}) \bigm| S_t = s\big] \\
&\le \mathbb{E}_{\pi'}\big[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 v_\pi(S_{t+3}) \bigm| S_t = s\big] \\
&\vdots \\
&\le \mathbb{E}_{\pi'}\big[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \bigm| S_t = s\big] \\
&= v_{\pi'}(s).
\end{aligned}
$$

DP – poprawa polityki

Co ten wynik oznacza?

Załóżmy, że dla pewnego stanu s znamy wartość v_π(s).

Przyjmijmy, że w stanie s wybieramy akcję $a \neq \pi$(s), czyli inną niż ta wynikająca z polityki π.

Jeżeli:

$$q_\pi(s,a) \geq v_\pi(s)$$

wówczas polityka π' identyczna jak π z wyjątkiem π'(s)= $a \neq \pi$(s) jest lepsza od polityki π.

DP – poprawa polityki

Możemy zatem zdefiniować nową zachłanną politykę π':

$$\begin{aligned}
\pi'(s) &\doteq \arg\max_a q_\pi(s,a) \\
&= \arg\max_a \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a] \\
&= \arg\max_a \sum_{s',r} p(s',r|s,a)\Big[r + \gamma v_\pi(s')\Big],
\end{aligned}$$

Polityka zachłanna wybiera akcję, która wygląda najlepiej krótkoterminowo - po jednym kroku - zgodnie z $v_\pi(s)$.

Z założenia polityka zachłanna π' spełnia warunki twierdzenia o poprawie polityki więc jest taka sama lub lepsza od polityki początkowej π.

DP – poprawa polityki

Proces tworzenia nowej polityki przez ulepszanie oryginalnej polityki, czyniąc ją chciwą ze względu na funkcję wartości stanu, nazywamy poprawą polityki.

Przypuśćmy, że nowa chciwa polityka π' jest tak dobra, ale nie lepsza niż stara polityka π. Oznacza to, że:

$$v_\pi = v_{\pi'}$$

Z definicji polityki π' wynika, że:

$$\begin{aligned} v_{\pi'}(s) &= \max_a \mathbb{E}[R_{t+1} + \gamma v_{\pi'}(S_{t+1}) \mid S_t = s, A_t = a] \\ &= \max_a \sum_{s',r} p(s', r \mid s, a)\Big[r + \gamma v_{\pi'}(s')\Big]. \end{aligned}$$

DP – poprawa polityki

$$\begin{aligned} v_{\pi'}(s) &= \max_a \mathbb{E}[R_{t+1} + \gamma v_{\pi'}(S_{t+1}) \mid S_t = s, A_t = a] \\ &= \max_a \sum_{s',r} p(s', r \mid s, a)\Big[r + \gamma v_{\pi'}(s')\Big]. \end{aligned}$$

A to oznacza, że $v_{\pi'}$ spełnia równanie optymalizacyjne Bellmana:

$$v_*(s) = \max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma v_*(s')\big]$$

Stąd wniosek, że: $v_{\pi'} = v_{\pi'} = v_*$ czyli π i π' są politykami optymalnymi.

DP – iteracja polityki

Z powyższych rozważań wynika, że możemy postępować według następujących kroków:

1. Możemy oszacować pewną politykę π czyli znaleźć funkcję wartości stanów V_π.
2. Funkcja V_π może nam posłużyć do polepszenia polityki π i uzyskania lepszej polityki π'.
3. Możemy oszacować politykę π' czyli znaleźć funkcję wartości stanów $V_{\pi'}$.
4. itd…

DP – iteracja polityki

W efekcie otrzymamy ciąg kolejnych oszacowań i ulepszeń:

$$\pi_0 \xrightarrow{\mathrm{E}} v_{\pi_0} \xrightarrow{\mathrm{I}} \pi_1 \xrightarrow{\mathrm{E}} v_{\pi_1} \xrightarrow{\mathrm{I}} \pi_2 \xrightarrow{\mathrm{E}} \cdots \xrightarrow{\mathrm{I}} \pi_* \xrightarrow{\mathrm{E}} v_*,$$

Ponieważ skończony MDP ma tylko skończoną liczbę polityk, proces ten musi zbiegać się do optymalnej polityki i optymalnej funkcji wartości stanu w skończonej liczbie iteracji.

Ten sposób znalezienia optymalnej polityki nazywa się iteracją polityki.

DP – iteracja polityki

Algorytm:

1. Initialization

   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation

   Loop:

   $\quad \Delta \leftarrow 0$

   $\quad$ Loop for each $s \in \mathcal{S}$:

   $\quad\quad v \leftarrow V(s)$

   $\quad\quad V(s) \leftarrow \sum_{s',r} p(s', r \,|\, s, \pi(s))\big[r + \gamma V(s')\big]$

   $\quad\quad \Delta \leftarrow \max(\Delta, |v - V(s)|)$

   until $\Delta < \theta$

DP – iteracja polityki

Algorytm:

3. Policy Improvement

   policy-stable $\leftarrow$ *true*

   For each $s \in \mathcal{S}$:

   $\quad$ *old-action* $\leftarrow \pi(s)$

   $\quad$ $\pi(s) \leftarrow \operatorname{argmax}_a \sum_{s',r} p(s', r | s, a)\big[r + \gamma V(s')\big]$

   $\quad$ If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*

   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

Koniec części 2