Zachodniopomorski
Uniwersytet Technologiczny
w Szczecinie

Wydział
Informatyki

STRATEGIA OPTYMALNA

Karol Działowski

nr albumu: 39259
przedmiot: Uczenie ze wzmocnieniem

Szczecin, 24 grudnia 2020

Spis treści

1 Cel laboratorium

Celem laboratorium nr 2 było wykorzystanie równań Bellmana do wyznaczania strategii optymalnej.

2 Zadanie 3

Przedmiotem rozważań jest gra w zapałki. Gracze mają do dyspozycji pewną liczbę (np. $N = 10$) zapałek. W grze bierze udział dwóch graczy, którzy na przemian zabierają 1, 2 lub 3 zapałki. Przegrywa ten gracz, który zabiera zapałki jako ostatni.

Należało wyznaczyć funkcję wartości dla strategii: $\pi_1(x) = -1$, $\pi_2(x) = -2$, $\pi_3(x) = -3$ oraz określić, która z danych strategii jest najlepsza.

Ponadto wyznaczono optymalną strategię gry, zakładając, że nasz przeciwnik gra w sposób losowy.

Oczekiwana wartość wzmocnienia:

$$R^a_{xy} = \begin{cases} 1 & \text{dla } (x=2, a=-1, y=0) \text{ lub } (x=3, a=-2, y=0) \text{ lub } (x=4, a=-3, y=0) \\ 0 & \text{w innych przypadkach} \end{cases} \tag{1}$$

Prawdopodobieństwa przejścia rozpisano dla każdego stanu:

Stan 0:

$P^{-1}_{00} = 1 \quad P^{-2}_{00} = 1 \quad P^{-3}_{00} = 1$

Stan 1:

$P^{-1}_{10} = 1 \quad P^{-2}_{10} = 1 \quad P^{-3}_{10} = 1$

Stan 2:

$P^{-1}_{20} = 1 \quad P^{-2}_{20} = 1 \quad P^{-3}_{20} = 1$

Stan 3:

$P^{-1}_{30} = \frac{2}{3} \quad P^{-2}_{30} = 1 \quad P^{-3}_{30} = 1$
$P^{-1}_{31} = \frac{1}{3}$

Stan 4:

$P^{-1}_{40} = \frac{1}{3} \quad P^{-2}_{40} = \frac{2}{3} \quad P^{-3}_{40} = 1$
$P^{-1}_{41} = \frac{1}{3} \quad P^{-2}_{41} = \frac{1}{3}$
$P^{-1}_{42} = \frac{1}{3}$

Stan 5:

$P^{-1}_{51} = \frac{1}{3} \quad P^{-2}_{50} = \frac{1}{3} \quad P^{-3}_{50} = \frac{2}{3}$
$P^{-1}_{52} = \frac{1}{3} \quad P^{-2}_{51} = \frac{1}{3} \quad P^{-3}_{51} = \frac{1}{3}$
$P^{-1}_{53} = \frac{1}{3} \quad P^{-2}_{52} = \frac{1}{3}$

Stan 6:

$P^{-1}_{62} = \frac{1}{3} \quad P^{-2}_{61} = \frac{1}{3} \quad P^{-3}_{60} = \frac{1}{3}$
$P^{-1}_{63} = \frac{1}{3} \quad P^{-2}_{62} = \frac{1}{3} \quad P^{-3}_{61} = \frac{1}{3}$
$P^{-1}_{64} = \frac{1}{3} \quad P^{-2}_{63} = \frac{1}{3} \quad P^{-3}_{62} = \frac{1}{3}$

Stan 7:

$P^{-1}_{73} = \frac{1}{3} \quad P^{-2}_{72} = \frac{1}{3} \quad P^{-3}_{71} = \frac{1}{3}$
$P^{-1}_{74} = \frac{1}{3} \quad P^{-2}_{73} = \frac{1}{3} \quad P^{-3}_{72} = \frac{1}{3}$
$P^{-1}_{75} = \frac{1}{3} \quad P^{-2}_{74} = \frac{1}{3} \quad P^{-3}_{73} = \frac{1}{3}$

Stan 8:

$P^{-1}_{84} = \frac{1}{3} \quad P^{-2}_{83} = \frac{1}{3} \quad P^{-3}_{82} = \frac{1}{3}$
$P^{-1}_{85} = \frac{1}{3} \quad P^{-2}_{84} = \frac{1}{3} \quad P^{-3}_{83} = \frac{1}{3}$
$P^{-1}_{86} = \frac{1}{3} \quad P^{-2}_{85} = \frac{1}{3} \quad P^{-3}_{84} = \frac{1}{3}$

Stan 9:

$P^{-1}_{95} = \frac{1}{3} \quad P^{-2}_{94} = \frac{1}{3} \quad P^{-3}_{93} = \frac{1}{3}$
$P^{-1}_{96} = \frac{1}{3} \quad P^{-2}_{95} = \frac{1}{3} \quad P^{-3}_{94} = \frac{1}{3}$
$P^{-1}_{97} = \frac{1}{3} \quad P^{-2}_{96} = \frac{1}{3} \quad P^{-3}_{95} = \frac{1}{3}$

Stan 10:

$P^{-1}_{10,6} = \frac{1}{3} \quad P^{-2}_{10,5} = \frac{1}{3} \quad P^{-3}_{10,4} = \frac{1}{3}$
$P^{-1}_{10,7} = \frac{1}{3} \quad P^{-2}_{10,6} = \frac{1}{3} \quad P^{-3}_{10,5} = \frac{1}{3}$
$P^{-1}_{10,8} = \frac{1}{3} \quad P^{-2}_{10,7} = \frac{1}{3} \quad P^{-3}_{10,6} = \frac{1}{3}$

2.1 Równania Bellmana

Zakładamy: $\gamma = 1$.

$$V^{k+1}(x) = \max_a \sum_y P^a_{xy} \cdot \left[R^a_{xy} + \gamma \cdot V^k(y)\right] \tag{2}$$

$$
\begin{aligned}
V^{k+1}(0) &= \text{max}\{1\cdot(0+\gamma\cdot V^k(0)),\quad 1\cdot(0+\gamma\cdot V^k(0)),\quad 1\cdot(0+\gamma\cdot V^k(0))\}\\
V^{k+1}(1) &= \text{max}\{1\cdot(0+\gamma\cdot V^k(0)),\quad 1\cdot(0+\gamma\cdot V^k(0)),\quad 1\cdot(0+\gamma\cdot V^k(0))\}\\
V^{k+1}(2) &= \text{max}\{1\cdot(1+\gamma\cdot V^k(0)),\quad 1\cdot(0+\gamma\cdot V^k(0)),\quad 1\cdot(0+\gamma\cdot V^k(0))\}\\
V^{k+1}(3) &= \text{max}\{\frac{2}{3}\cdot(0+\gamma\cdot V^k(0))+\frac{1}{3}\cdot(0+\gamma\cdot V^k(1)),\quad 1\cdot(1+\gamma\cdot V^k(0)),\quad 1\cdot(0+\gamma\cdot V^k(0))\}\\
V^{k+1}(4) &= \text{max}\{\frac{1}{3}\cdot(0+\gamma\cdot V^k(0))+\frac{1}{3}\cdot(0+\gamma\cdot V^k(1))+\frac{1}{3}\cdot(0+\gamma\cdot V^k(2)),\\
&\quad \frac{2}{3}\cdot(0+\gamma\cdot V^k(0))+\frac{1}{3}\cdot(0+\gamma\cdot V^k(1)),\\
&\quad 1\cdot(1+\gamma\cdot V^k(0))\}\\
V^{k+1}(5) &= \text{max}\{\frac{1}{3}\cdot(0+\gamma\cdot V^k(1))+\frac{1}{3}\cdot(0+\gamma\cdot V^k(2))+\frac{1}{3}\cdot(0+\gamma\cdot V^k(3)),\\
&\quad \frac{1}{3}\cdot(0+\gamma\cdot V^k(0))+\frac{1}{3}\cdot(0+\gamma\cdot V^k(1))+\frac{1}{3}\cdot(0+\gamma\cdot V^k(2)),\\
&\quad \frac{2}{3}\cdot(0+\gamma\cdot V^k(0))+\frac{1}{3}\cdot(0+\gamma\cdot V^k(1))\}\\
V^{k+1}(6+i) &= \text{max}\{\frac{1}{3}\cdot(0+\gamma\cdot V^k(i+2))+\frac{1}{3}\cdot(0+\gamma\cdot V^k(i+3))+\frac{1}{3}\cdot(0+\gamma\cdot V^k(i+4)),\\
&\quad \frac{1}{3}\cdot(0+\gamma\cdot V^k(i+1))+\frac{1}{3}\cdot(0+\gamma\cdot V^k(i+2))+\frac{1}{3}\cdot(0+\gamma\cdot V^k(i+3)),\\
&\quad \frac{1}{3}\cdot(0+\gamma\cdot V^k(i))+\frac{1}{3}\cdot(0+\gamma\cdot V^k(i+1))+\frac{1}{3}\cdot(0+\gamma\cdot V^k(i+2))\}
\end{aligned}
$$

Dla $i \in \{0, ..., N-6\}$.

2.2 Wyznaczone wartości strategii

Wyznaczono funkcję wartości dla strategii: $\pi_1(x) = -1$, $\pi_2(x) = -2$, $\pi_3(x) = -3$. Na podstawie zebranych wyników, nie można określić, która ze strategii jest najlepsza.

x	$V^{\pi_1}(x)$	$V^{\pi_2}(x)$	$V^{\pi_3}(x)$
0	0.000000	0.000000	0.000000
1	0.000000	0.000000	0.000000
2	1.000000	0.000000	0.000000
3	0.000000	1.000000	0.000000
4	0.333333	0.000000	1.000000
5	0.333333	0.000000	0.000000
6	0.444444	0.333333	0.000000
7	0.222222	0.333333	0.000000
8	0.370370	0.333333	0.333333
9	0.333333	0.111111	0.333333
10	0.345679	0.222222	0.333333

Tabela 1: Funkcje wartości dla strategii

2.3 Optymalna strategia gry

Przy założeniu, że nasz przeciwnik gra w sposób losowy wyznaczono optymalną strategię gry.

x	$Q^{\pi^*}(x,-1)$	$Q^{\pi^*}(x,-2)$	$Q^{\pi^*}(x,-3)$
0	0.000000	0.000000	0.000000
1	0.000000	0.000000	0.000000
2	1.000000	0.000000	0.000000
3	0.000000	1.000000	0.000000
4	0.333333	0.000000	1.000000
5	0.666667	0.333333	0.000000
6	1.000000	0.666667	0.333333
7	0.888889	1.000000	0.666667
8	0.888889	0.888889	1.000000
9	0.888889	0.888889	0.888889
10	1.000000	0.888889	0.888889

Tabela 2: Funkcja wartości akcji

Optymalna funkcja wartośći : $V^*(x) = [1.0, 0.89, 1, 1, 1, 0.66, 1, 1, 1, 0, 0]$

Optymalna strategia wyznaczona jako zachłanna może być opisana sekwencją akcji: $-1, *, -3, -2, -1, -1, -3, -2, -1, *, *$.

2.4 Kod źródłowy

Kod źródłowy zamieszczono również w formie tekstowej do zadania.

Kod źródłowy 1: Zadanie 3 - zapałki
Źródło: Opracowanie własne

```python
"""
Gra w zapałki

Mamy do dyspozycji pewną liczbę N zapałek.
W grze bierze udział dwóch graczy, którzy na przemian zabierają
1, 2 lub 3 zapałki.
Przegrywa ten gracz, który zabiera zapałki jako ostatni.
Przeciwnik będzie grał w sposób losowy.
"""

import numpy as np

N = 10

def Q_next(V, gamma):
    Q = np.zeros((N+1, 3))
```

```

    # Stan 0
    Q[0, 0] = 1 * (0 + gamma * V[0])
    Q[0, 1] = 1 * (0 + gamma * V[0])
    Q[0, 2] = 1 * (0 + gamma * V[0])

    # Stan 1
    Q[1, 0] = 1 * (0 + gamma * V[0])
    Q[1, 1] = 1 * (0 + gamma * V[0])
    Q[1, 2] = 1 * (0 + gamma * V[0])

    # Stan 2
    Q[2, 0] = 1 * (1 + gamma * V[0])  # ruch zwycięski
    Q[2, 1] = 1 * (0 + gamma * V[0])
    Q[2, 2] = 1 * (0 + gamma * V[0])

    # Stan 3
    Q[3, 0] = 2/3 * (0 + gamma * V[0]) + 1/3 * (0 + gamma * V[1])
    Q[3, 1] = 1 * (1 + gamma * V[0])  # ruch zwycieski
    Q[3, 2] = 1 * (0 + gamma * V[0])

    # Stan 4
    Q[4, 0] = 1/3 * (0 + gamma * V[0]) + 1/3 * (0 + gamma * V[1]) + 1/3 * (0 + gamma
* V[2])
    Q[4, 1] = 2/3 * (0 + gamma * V[0]) + 1/2 * (0 + gamma * V[1])
    Q[4, 2] = 1 * (1 + gamma * V[0])  # ruch zwycieski

    # Stan 5
    Q[5, 0] = 1/3 * (0 + gamma * V[1]) + 1/3 * (0 + gamma * V[2]) + 1/3 * (0 + gamma
* V[3])
    Q[5, 1] = 1/3 * (0 + gamma * V[0]) + 1/3 * (0 + gamma * V[1]) + 1/3 * (0 + gamma
* V[2])
    Q[5, 2] = 2/3 * (0 + gamma * V[0]) + 1/3 * (0 + gamma * V[1])

    # Stany od 6 do N
    x = 6
    for i in range(N-5):
        Q[x + i, 0] = 1/3 * (0 + gamma * V[i+2]) + 1/3 * (0 + gamma * V[i+3]) + 1/3
* (0 + gamma * V[i+4])
        Q[x + i, 1] = 1/3 * (0 + gamma * V[i+1]) + 1/3 * (0 + gamma * V[i+2]) + 1/3
* (0 + gamma * V[i+3])
        Q[x + i, 2] = 1/3 * (0 + gamma * V[i]) + 1/3 * (0 + gamma * V[i+1]) + 1/3 *
(0 + gamma * V[i+2])

    return Q

def wartosc_strategii(id):
    """
    :param id: numer strategii gdzie 0 = -1, 1 = -2, 2 = -3
    """
    V_initial = np.zeros(N+1)
```

```python
    V = V_initial
    V_prev = V

    for i in range(100):
        Q = Q_next(V, gamma=1)
        V = Q[:, id]

        if np.sum(V_prev - V) == 0:
            break

        V_prev = V

    return Q

def optymalna_strategia():
    V_initial = np.zeros(N+1)
    V = V_initial
    V_prev = V

    for i in range(100):
        Q = Q_next(V, gamma=1)
        V = np.max(Q, axis=1)

        if np.sum(V_prev - V) == 0:
            break

        V_prev = V

    return Q


if __name__ == "__main__":
    import pandas as pd

    # Porównanie wartości strategii
    print("V pi1")
    Q1 = wartosc_strategii(0)
    print(Q1[:, 0])
    print("V pi2")
    Q2 = wartosc_strategii(1)
    print(Q2[:, 1])
    print("V pi3")
    Q3 = wartosc_strategii(2)
    print(Q3[:, 2])

    data = {"pi1": Q1[:, 0], "pi2": Q2[:, 1], "pi3": Q3[:, 2]}
    df = pd.DataFrame.from_dict(data)
    print(df.to_latex())


    # Optymalna strategia
```

```
    print("Q optymalnej strategi")
    Q = optymalna_strategia()
    df = pd.DataFrame(Q)
    print(df.to_latex())
```

3 Zadanie 4

Przedmiotem rozważań w tym zadaniu była gra w ruletkę, gdzie obstawia się wielokrotności 1 zł na kolor z prawdopodobieństwem wygranej $p = \frac{18}{37}$.

Należało wyznaczyć optymalną strategię obstawiania, tak aby wygrać dokładnie 100 zł. Wykreślono optymalną funkcję wartości $V^*(x)$ i optymalnej strategii $\pi^*(x)$.

3.1 Założenia

Stan x określa kapitał gracza w pełnych zł: $x \in \{0, 1, 2, 3, \dots, 100\}$. Możliwe akcje do wykonania w stanie x to $a \in \{1, \dots, \min(x, 100 - x)\}$.

Oczekiwana wartość wzmocnienia:

$$R^a_{xy} = \begin{cases} 0 & \text{dla } y \neq 100 \\ 1 & \text{dla } y = 100 \end{cases} \tag{3}$$

Prawdopodobieństwa przejścia:

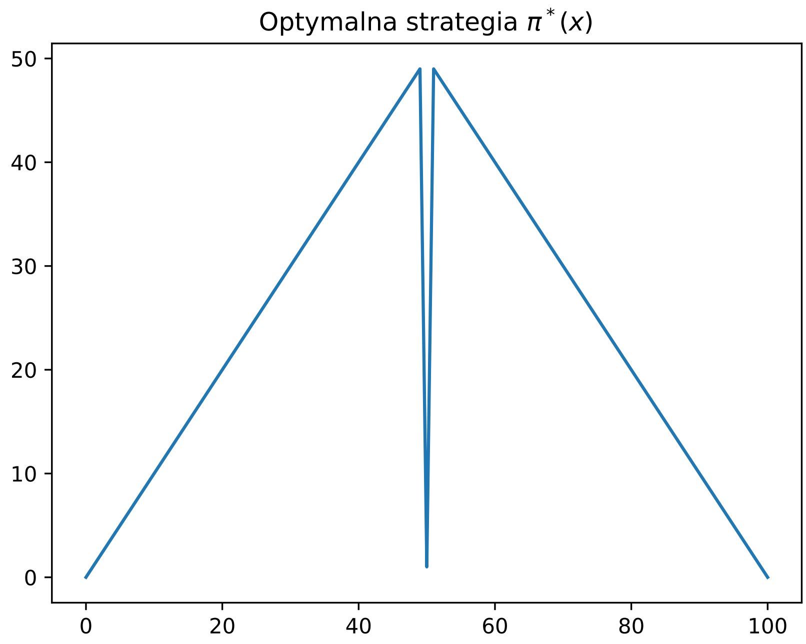
$$P^a_{xy} = \begin{cases} p & \text{dla } y = x + a \\ (1 - p) & \text{dla } y = x - a \end{cases} \tag{4}$$

3.2 Wyniki

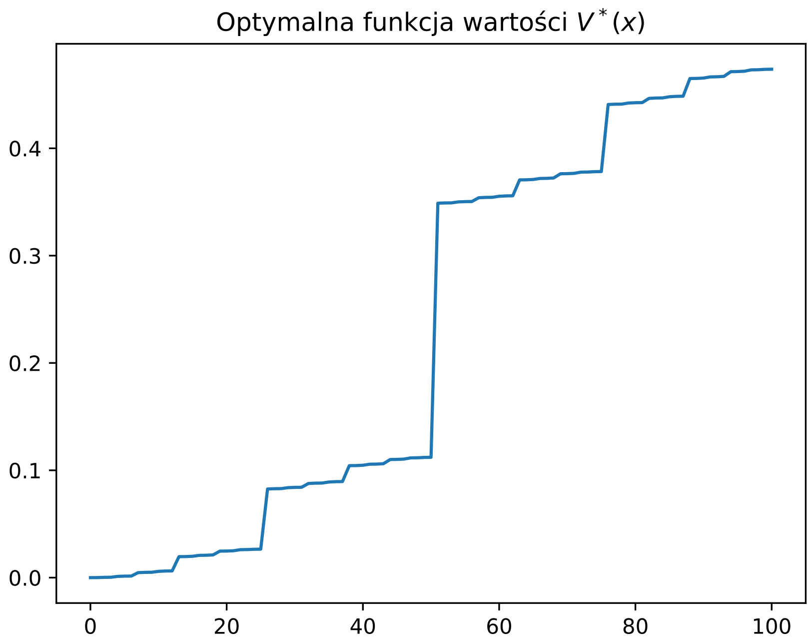
Wyznaczono wykres optymalnej funkcji wartości i optymalnej strategi korzystając z kodu z listingu 2.

Otrzymane wyniki są dosyć zaskakujące i trudno mi je uzasadnić. Optymalną strategią jest gra va banque, ryzykując wszystko. Jest to logiczne, ponieważ im dłużej gramy, tym częściej będziemy przegrywać. Nie rozumiem skąd wynika akcja 1 w stanie 50 zł. Być może gdzieś leży błąd w stworzonych równania Bellmana.

Optymalna funkcja wartości, przedstawiona na rysunku 2 przedstawia gwałtowne skoki w punktach takich jak 25, 50, 75. Tendencja rosnąca funkcji wartości jest zrozumiała, bo im więcej mamy pieniędzy (większy stan) tym bardziej prawdopodobne, że dojdziemy do kapitału 100 zł.



Rysunek 1: Optymalna strategia



Rysunek 2: Funkcja wartości

3.3 Kod źródłowy

Kod źródłowy 2: Zadanie 4 - ruletka
Źródło: Opracowanie własne

```python
"""
Gra w ruletkę.
Obstawiamy wielokrotność 1 zł na kolor.
Porawdopobieństwo wygranej p = 18/37

Wyznaczyć optymalną strategię obstawiania, tak aby wygrać dokładnie 100 zł.
Sporządzić wykres optymalnej funkcji wartości V*(x) i optymalnej strategii pi*(x)

Stan x określa kapitał gracza x in {0, 1, 2, 3, ..., 100}
Możliwe akcje a w stanie x: a in {1, ..., min(x, 100-x)}
    Czyli dla maksymalna akcja = 50 (bo dla x = 50 możemy postawić maksymalnie 50).
    Co dla stanu 0? Raczej nie ma żadnej możliwej akcji.
Oczekiwana wartość wzmocnienia: R_{xy} = 1 dla y = 100, w innych przypadkach R=0
Prawodopodobieństwa przejścia p dla y = x+a (wygrana), w innym przypadku (1-p)
"""


import numpy as np
import matplotlib.pyplot as plt

p = 18 / 37


def Q_next(V, gamma):
    Q = np.zeros((101, 50))  # każdy maksymalnie będzie 50 stanów (dla x = 50 najwiecej możliwych akcji do wykonania)

    for x in range(101):  # dla wszystkich stanów
        for a in range(50):  # dla wszystkich akcji
            if a > min(x, 100 - x):  # niemożliwe akcje
                Q[x, a] = 0
            else:
                loss = (1 - p) * (0 + gamma * V[x - a])
                R = 1 if x + a == 100 else 0
                win = p * (R + gamma * V[x + a])
                Q[x, a] = p * win + (1 - p) * loss

    return Q


def optymalna_strategia():
    V_initial = np.zeros(101)  # dla każdego stanu {0, 1, 2, ..., 100}
    V = V_initial
    V_prev = V

    while True:
        Q = Q_next(V, gamma=1)
        V = np.max(Q, axis=1)

        if np.sum(V_prev - V) == 0:
            break
```

```

        V_prev = V

    return Q


if __name__ == "__main__":
    print("Q optymalnej strategi")
    Q = optymalna_strategia()
    print(Q[50, :])

    # TODO: coś mi nie pasuje, bo:
    # występują skoki
    # dla stanu 50 zł powinno sie postawić złotówkę, w innych przypadkach trzeba grać va bank
    # dla stanu 50 zł wartość akcji jest wszędzie prawie identyczna

    V = np.max(Q, axis=1)
    plt.plot(V)
    plt.title("Optymalna funkcja wartości $V^*(x)$")
    plt.show()
    pi = np.argmax(Q, axis=1)
    plt.plot(pi)
    plt.title("Optymalna strategia $\pi^*(x)$")
    plt.show()
    print(Q)
```