Sprawozdanie: Model Sznajdów
Autor: Jakub Bąk

OPIS MODELU:

Model sznajdów jest to model bazujący na społecznym dowodzie słuszności. Jest to zasada, w imię której człowiek, nie wiedząc, jaka decyzja lub jaki pogląd jest słuszny (co może zależeć od różnych czynników), podejmuje decyzje lub przyjmuje poglądy takie same, jak większość grupy. W tym konkretnym modelu Sznajdów dwóch sąsiadów, którzy mają te same poglądy wpływają na swych sąsiadów, nadając im swoją opinię. Nie rozpatrujemy przypadku, gdy sąsiedzi mają różne opinie(co prowadzi do skłócenia swoich sąsiadów).

W tym projekcie stworzyłem taki model Sznajdów z założeniami zgodnymi dla każdego zadania:

- Grupa społeczna jest przedstawiona za pomocą dwuwymiarowej sieci z periodycznymi warunkami brzegowymi
- Istnieje opinia za(o wartości 1) oraz przeciw(wartość 0)
- W chwili początkowej sieć jest uzupełniana losowymi wartościami, które mają symulować rozkład opinii
- W każdym kroku czasowym wybierana jest dowolna para sąsiadów(zarówno pozioma jak i pionowa)

Zadanie 1.

Treść zadania: *„Na dwuwymiarowej sieci (kwadratowej, z periodycznymi warunkami brzegowymi) prześledźmy ewolucje niewielkiego układu dla kilku kroków czasowych. W chwili początkowej t = 0 opinie są losowe, fiftyfifty. Symulacje kończymy po dziesiątej skutecznej próbie wyboru uzgodnionej pary. Po każdym takim „szczęśliwym wyborze" wypisz współrzędne wylosowanej pary i stan sieci jako całości."*

Pierwszą rzeczą, którą wykonałem było stworzenie tablicy 10x10, która byłaby losowo wypełniona „1" oraz „0", czyli opiniami *fifty fifty*.

Następnie za pomocą kolejnej funkcji losowałem dowolną parę i sprawdzałem czy któryś z jej elementów nie „wypada" poza tablicę. Dodatkowo ustawiłem warunek, aby para nie była po skosie, tylko była wybrana poziomo, bądź pionowo.

Kolejnym krokiem było stworzenie modelu Sznajda. W zależności czy wylosowana para jest pozioma, czy pionowa, model ma różny sposób radzenia sobie, który ostatecznie prowadzi do tych samych wyników. Dla pary o różnych wartościach (0,1) nic się nie zmienia, podczas gdy dla pary o tych samych wartościach (0,0) lub (1,1), ich sąsiedzi również zmieniają się na ich wartości.

Poniżej przedstawione są rezultaty losowania 10 par:

```
Result of 1 draw:
Pair's cords:((5, 7),(4, 7))
Pair's values:(1,1)
'Yes' answers:55
[[1 0 1 0 1 1 1 1 1 1]
 [1 1 1 0 1 1 1 0 1 0]
 [0 0 1 0 0 1 1 1 0 0]
 [0 1 1 1 0 1 1 1 0 1]
 [0 1 0 1 0 1 1 1 1 1]
 [1 1 0 1 1 0 1 0 1 1]
 [0 0 1 0 0 1 0 0 0 1]
 [1 1 1 1 0 0 1 0 0 1]
 [0 0 0 0 1 0 0 0 0 0]
 [0 1 0 1 0 0 1 1 0 1]]
```

```
Result of 2 draw:
Pair's cords:((2, 4),(3, 4))
Pair's values:(0,0)
'Yes' answers:51
[[1 0 1 0 1 1 1 1 1 1]
 [1 1 0 0 0 1 1 0 1 0]
 [0 0 0 0 0 1 1 1 0 0]
 [0 1 1 0 0 1 1 1 0 1]
 [0 1 0 1 0 1 1 1 1 1]
 [1 1 0 1 1 0 1 0 1 1]
 [0 0 1 0 0 1 0 0 0 1]
 [1 1 1 1 0 0 1 0 0 1]
 [0 0 0 0 1 0 0 0 0 0]
 [0 1 0 1 0 0 1 1 0 1]]
```

```
Result of 3 draw:
Pair's cords:((8, 9),(8, 10))
Pair's values:(0,1)
'Yes' answers:51
[[1 0 1 0 1 1 1 1 1 1]
 [1 1 0 0 0 1 1 0 1 0]
 [0 0 0 0 0 1 1 1 0 0]
 [0 1 1 0 0 1 1 1 0 1]
 [0 1 0 1 0 1 1 1 1 1]
 [1 1 0 1 1 0 1 0 1 1]
 [0 0 1 0 0 1 0 0 0 1]
 [1 1 1 1 0 0 1 0 0 1]
 [0 0 0 0 1 0 0 0 0 0]
 [0 1 0 1 0 0 1 1 0 1]]
```

```
Result of 4 draw:
Pair's cords:((9, 5),(10, 5))
Pair's values:(1,0)
'Yes' answers:51
[[1 0 1 0 1 1 1 1 1 1]
 [1 1 0 0 0 1 1 0 1 0]
 [0 0 0 0 0 1 1 1 0 0]
 [0 1 1 0 0 1 1 1 0 1]
 [0 1 0 1 0 1 1 1 1 1]
 [1 1 0 1 1 0 1 0 1 1]
 [0 0 1 0 0 1 0 0 0 1]
 [1 1 1 1 0 0 1 0 0 1]
 [0 0 0 0 1 0 0 0 0 0]
 [0 1 0 1 0 0 1 1 0 1]]
```

```
Result of 5 draw:
Pair's cords:((5, 2),(5, 3))
Pair's values:(1,0)
'Yes' answers:51
[[1 0 1 0 1 1 1 1 1 1]
 [1 1 0 0 0 1 1 0 1 0]
 [0 0 0 0 0 1 1 1 0 0]
 [0 1 1 0 0 1 1 1 0 1]
 [0 1 0 1 0 1 1 1 1 1]
 [1 1 0 1 1 0 1 0 1 1]
 [0 0 1 0 0 1 0 0 0 1]
 [1 1 1 1 0 0 1 0 0 1]
 [0 0 0 0 1 0 0 0 0 0]
 [0 1 0 1 0 0 1 1 0 1]]
```

```
Result of 6 draw:
Pair's cords:((4, 8),(4, 7))
Pair's values:(1,1)
'Yes' answers:52
[[1 0 1 0 1 1 1 1 1 1]
 [1 1 0 0 0 1 1 0 1 0]
 [0 0 0 0 0 1 1 1 0 0]
 [0 1 1 0 0 1 1 1 1 1]
 [0 1 0 1 0 1 1 1 1 1]
 [1 1 0 1 1 0 1 0 1 1]
 [0 0 1 0 0 1 0 0 0 1]
 [1 1 1 1 0 0 1 0 0 1]
 [0 0 0 0 1 0 0 0 0 0]
 [0 1 0 1 0 0 1 1 0 1]]
```

```
Result of 7 draw:
Pair's cords:((10, 6),(10, 7))
Pair's values:(0,1)
'Yes' answers:52
[[1 0 1 0 1 1 1 1 1 1]
 [1 1 0 0 0 1 1 0 1 0]
 [0 0 0 0 0 1 1 1 0 0]
 [0 1 1 0 0 1 1 1 1 1]
 [0 1 0 1 0 1 1 1 1 1]
 [1 1 0 1 1 0 1 0 1 1]
 [0 0 1 0 0 1 0 0 0 1]
 [1 1 1 1 0 0 1 0 0 1]
 [0 0 0 0 1 0 0 0 0 0]
 [0 1 0 1 0 0 1 1 0 1]]
```

```
Result of 8 draw:
Pair's cords:((6, 3),(6, 4))
Pair's values:(0,1)
'Yes' answers:52
[[1 0 1 0 1 1 1 1 1 1]
 [1 1 0 0 0 1 1 0 1 0]
 [0 0 0 0 0 1 1 1 0 0]
 [0 1 1 0 0 1 1 1 1 1]
 [0 1 0 1 0 1 1 1 1 1]
 [1 1 0 1 1 0 1 0 1 1]
 [0 0 1 0 0 1 0 0 0 1]
 [1 1 1 1 0 0 1 0 0 1]
 [0 0 0 0 1 0 0 0 0 0]
 [0 1 0 1 0 0 1 1 0 1]]
```

```
Result of 9 draw:
Pair's cords:((6, 1),(5, 1))
Pair's values:(1,0)
'Yes' answers:52
[[1 0 1 0 1 1 1 1 1 1]
 [1 1 0 0 0 1 1 0 1 0]
 [0 0 0 0 0 1 1 1 0 0]
 [0 1 1 0 0 1 1 1 1 1]
 [0 1 0 1 0 1 1 1 1 1]
 [1 1 0 1 1 0 1 0 1 1]
 [0 0 1 0 0 1 0 0 0 1]
 [1 1 1 1 0 0 1 0 0 1]
 [0 0 0 0 1 0 0 0 0 0]
 [0 1 0 1 0 0 1 1 0 1]]
```

```
Result of 10 draw:
Pair's cords:((10, 4),(10, 5))
Pair's values:(1,0)
'Yes' answers:52
[[1 0 1 0 1 1 1 1 1 1]
 [1 1 0 0 0 1 1 0 1 0]
 [0 0 0 0 0 1 1 1 0 0]
 [0 1 1 0 0 1 1 1 1 1]
 [0 1 0 1 0 1 1 1 1 1]
 [1 1 0 1 1 0 1 0 1 1]
 [0 0 1 0 0 1 0 0 0 1]
 [1 1 1 1 0 0 1 0 0 1]
 [0 0 0 0 1 0 0 0 0 0]
 [0 1 0 1 0 0 1 1 0 1]]
```

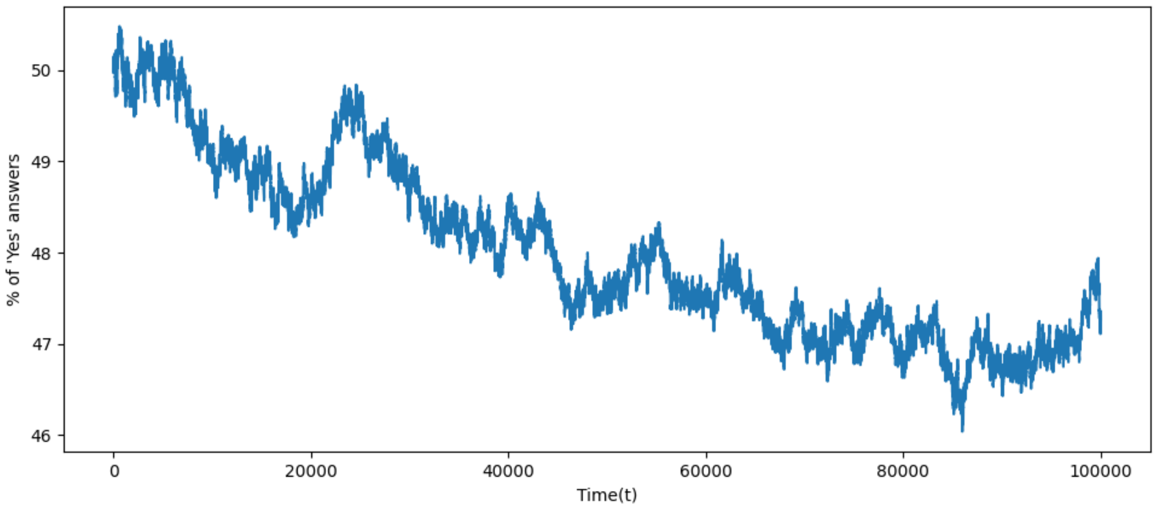
Rys 1. Wyniki losowania wraz ze współrzędnymi, wartościami pary, odpowiedziami na „tak" oraz stanem sieci

Zadanie 2.

Treść zadania: „*Zwiększmy rozmiar układu. Śledzimy czasowa ewolucje frakcji (t) aktorów „na tak". Parametrem wejściowym jest frakcja p0 = p(t = 0) aktorów „na tak" w chwili początkowej. Rozkład przestrzenny tych aktorów pozostaje przypadkowym.*"

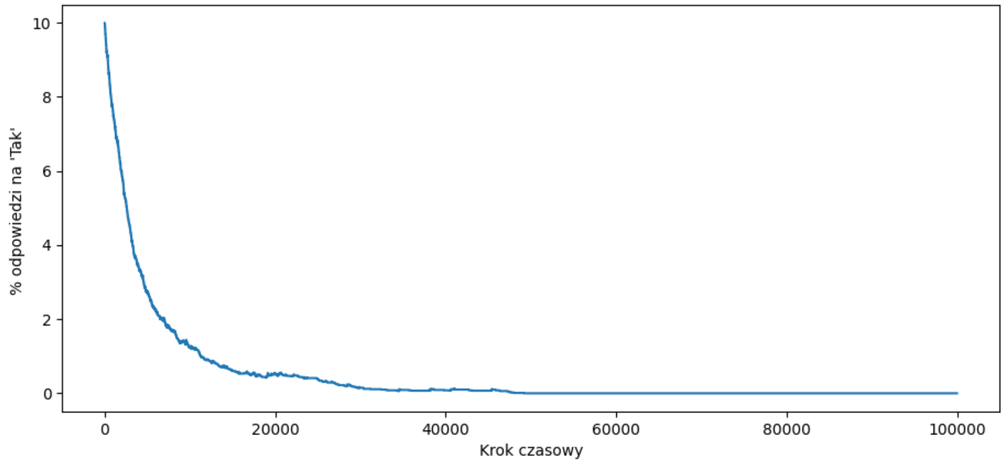
Układ zwiększyłem do tablicy 100x100 zostając przy wypełnieniu jedynkami w 50% elementów tablicy. Dodatkowo wprowadziłem zmienną *„t_count"* oraz *„p_count"*, którą posłużą do narysowania wykresu przebiegu wszystkich kroków oraz tego jak zmieniała się ilość aktorów na „tak".

Ostatnim punktem było narysowanie wykresu funkcji. Skorzystałem tutaj z biblioteki *matplotlib* w Pythonie, która pozwala na przedstawienie graficzne danych. Wykres został narysowany dla 100 tys. kroków czasowych.



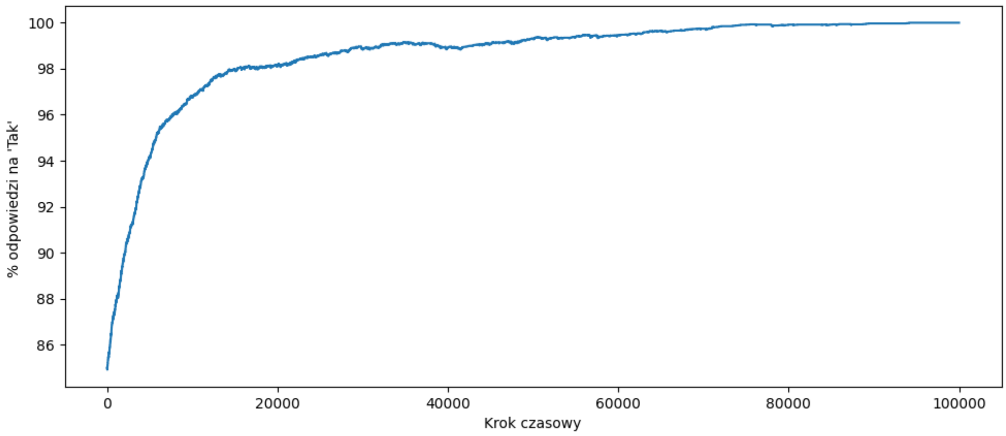
Rys.2 Wykres odpowiedzi „Tak" w okresie czasowym t = 100000, dla p0 = 50%

Dla p0 równego 50%, sieć jest w stanie względnej równowagi, żadna opinia nie przeważa znacząco nad drugą.



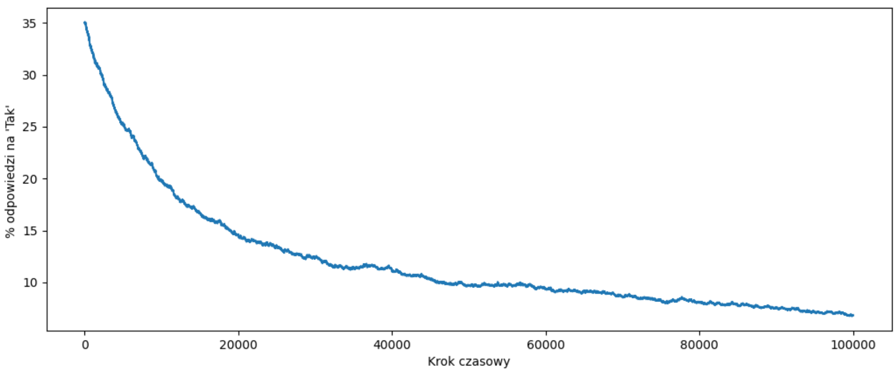
Rys.3 Wykres odpowiedzi „Tak” w okresie czasowym t = 100000, dla p0 = 10%

Jak widzimy, dla małej liczby początkowej odpowiedzi na Tak, sieć bardzo szybko dąży do 0% opinii Za, co udaje się sieci w około 50 tys. Krokach.



Rys.4 Wykres odpowiedzi „Tak” w okresie czasowym t = 100000, dla p0 = 85%

Podobnie jak w przypadku wcześniejszego przykładu, tak tutaj sieć przy dużej przewadze odpowiedzi Za dąży do wypełnienia tylko jedną opinią, w tym wypadku opiniami na tak.



Rys.5 Wykres odpowiedzi „Tak” w okresie czasowym t = 100000, dla p0 = 35%

Zadanie 3.

Treść zadania:

Wielokrotnie zaczynamy symulację z taką samą liczbą aktorów „na tak” ($\rho_0 = \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$) – ale inaczej rozłożonych na sieci. Dla ustalonego ρ_0 po długim czasie zapisujemy końcową frakcję $\rho_\infty = \rho(t \to \infty)$ aktorów „na tak”. Wyniki uśredniamy $\langle \cdots \rangle$ po R symulacjach i szacujemy niepewność tej średniej. Sporządź tabelę $\langle \rho_\infty \rangle$ vs. ρ_0. Jak zmieniają się wyniki dla $R = 10$, 10^3, 10^5?

Pierwszą rzeczą było stworzenie trzech różnych tablic o wymiarach 20x20 i p0 =(¼, ½, ¾). Następnie stworzyłem trzy funkcje, które odpowiadały za przejście przez 10, 10^3 oraz 10^5 symulacji dla każdego p0, a następnie funkcje, która oblicza średnią oraz niepewność tej średniej. Posłużą one potem do stworzenia tabeli z podsumowaniem każdego rodzaju symulacji.

Po przejściu przez wszystkie rodzaje p0, a także ilości symulacji (W sumie 9 różnych ilości symulacji, po trzy wartości dla każdego p0) stworzyłem tabelę, która podsumowuje wszystkie wyniki.

p0	pAvg	R	pStd
0.25	5,1%	10	0.768
0.25	4,4%	1000	0.811
0.25	4,41%	10000	0.834
0.5	50,9%	10	2.54
0.5	50,3%	1000	2.759
0.5	49,72%	10000	2.721
0.75	92,75%	10	1.13
0.75	95,3%	1000	0.82
0.75	95,5%	10000	0.828

Rys 6. Tabela przedstawiająca wyniki wszystkich symulacji.

Jak możemy zobaczyć dla małej ilości symulacji(R=10) dla każdego p0 wyniki są najbardziej różniące się od innych. Wynika to z tego, że dla takiej małej ilości R nawet jedno odstępstwo może znacząco wpłynąć na średni wynik. Dla większej ilości symulacji wyniki uśredniają się do pewnego momentu, w przypadku mojego modelu do odpowiednio 4,4%, 50% oraz 95%. Zwiększenie ilości symulacji do 10000 nie wpływa znacząco na te wyniki, co może sugerować, że 1000 symulacji jest wystarczającą ilością do wyciągnięcia wniosków oraz określenia pewnych średnich wartości.