Metody Inżynierii Wiedzy

Wyznaczanie rozkładu stacjonarnego dla łańcucha Markowa

Dr inż. Michał Majewski

mmajew@pjwstk.edu.pl

materiały: *ftp*(*public*) : *//mmajew/MIW*

Zadanie 1

Napisz program, który oblicza **prawdopodobieństwo** P(x|Q), **że osoba ma rzadką chorobę** (prawdopodobieństwo bycia chorym P(x) = 0.1% dla populacji), **biorąc pod uwagę pozytywny wynik testu** P(Q). Dane wejściowe to:

- **Prawdopodobieństwo pozytywnego wyniku testu** P(Q), **jeśli osoba ma chorobę** P(x), **to** P(Q|x)=**0.99**,
- czyli prawdopodobieństwo pozytywnego wyniku testu P(Q), jeśli osoba nie ma choroby P(-x), to P(Q|-x)=0.01.

Szkic kodu 000_Bayes_Theorem.py

Interpretacja reguły Bayesa

- Dla modelu opisanego parametrami Q prawdopodobieństwo $p(Q\backslash x)$ określa wiarygodność modelu dla danych x

$$p(x\backslash Q) = \frac{p(Q\backslash x)p(x)}{p(Q)}$$

$$posteriori = \frac{wiarygodność \cdot apriori}{normalizacja}$$

- Dla *normalizacji =1* (“model evidence”)

$$aposteriori \Leftarrow wiarygodność \cdot apriori$$

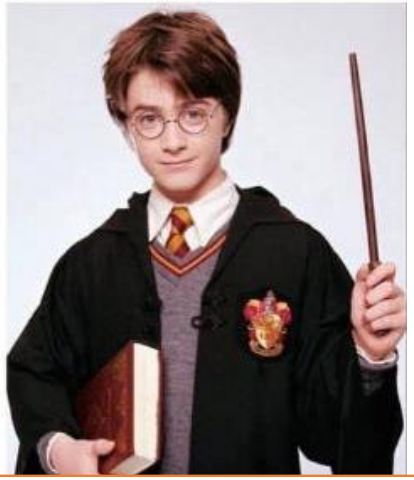
Równanie Bayesa - jak zmieniamy naszą wiarę na temat czegoś, gdy dowiadujemy się nowych informacji - prawdopodobieństwo wystąpienia pewnego zdarzenia, biorąc pod uwagę naszą wstępną wiarę oraz nowe dowody.

Prawdopodobieństwo zdarzenia Q pod warunkiem x, P(Q\x) - jest to sposób określania, jak dobrze nowe dowody Q wspierają naszą wstępną wiarę w wystąpienie zdarzenia x.

$$p(x\backslash Q) = \frac{p(Q\backslash x)p(x)}{p(Q)}$$

$$posteriori = \frac{wiarygodność \cdot apriori}{normalizacja}$$

Prawdopodobieństwo a posteriori p(x\Q) -czego dowiadujemy się po uzyskaniu nowych informacji lub dowodów - **prawdopodobieństwo wystąpienia zdarzenia x, biorąc pod uwagę nowe dowody Q.**



Prawdopodobieństwo a priori p(x) - nasza **wstępna wiedza** lub przekonanie na temat tego, jakie jest prawdopodobieństwo wystąpienia zdarzenia p(x), zanim poznamy jakiekolwiek nowe dowody.

Prawdopodobieństwo zdarzenia P(Q) - ogólne prawdopodobieństwo wystąpienia zdarzenia Q, niezależnie od tego, czy zdarzenie x ma miejsce czy nie.

Zadanie 1

Oblicz prawdopodobieństwo **P(x|Q),** że osoba **ma** rzadką **chorobę** biorąc pod uwagę **pozytywny wynik** testu

Prawdopodobieństwo **bycia chorym P(x) = 0.001**

Prawdopodobieństwo **pozytywnego wyniku testu chorej osoby P(Q|x)=0.99**

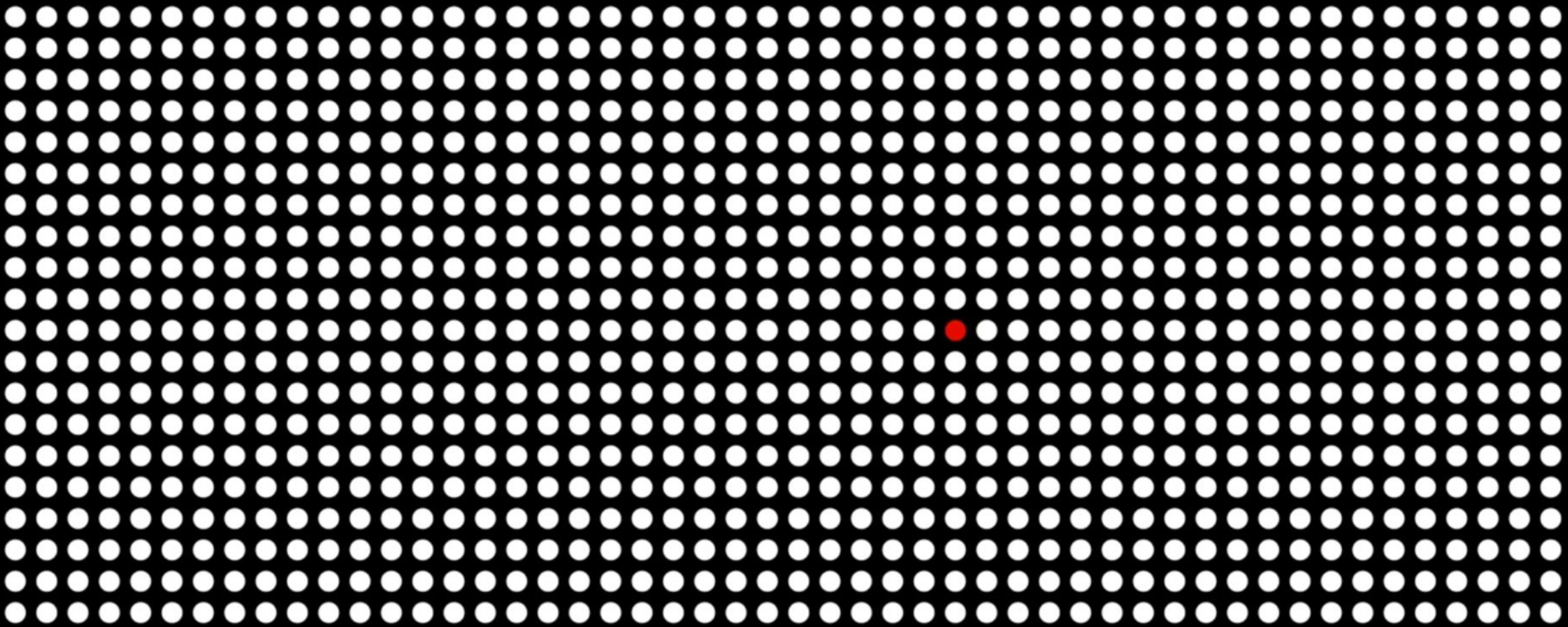
Prawdopodobieństwo **pozytywnego wyniku testu jeśli osoba nie ma choroby P(Q|-x)=0.01.**

Prawdopodobieństwo **pozytywnego wyniku testu** niezależnie od stanu zdrowia **P(Q)= P(Q|x)*P(x)+P(Q|-x)*P(-x),**

$$p(x \backslash Q) = \frac{p(Q \backslash x) p(x)}{p(Q)}$$

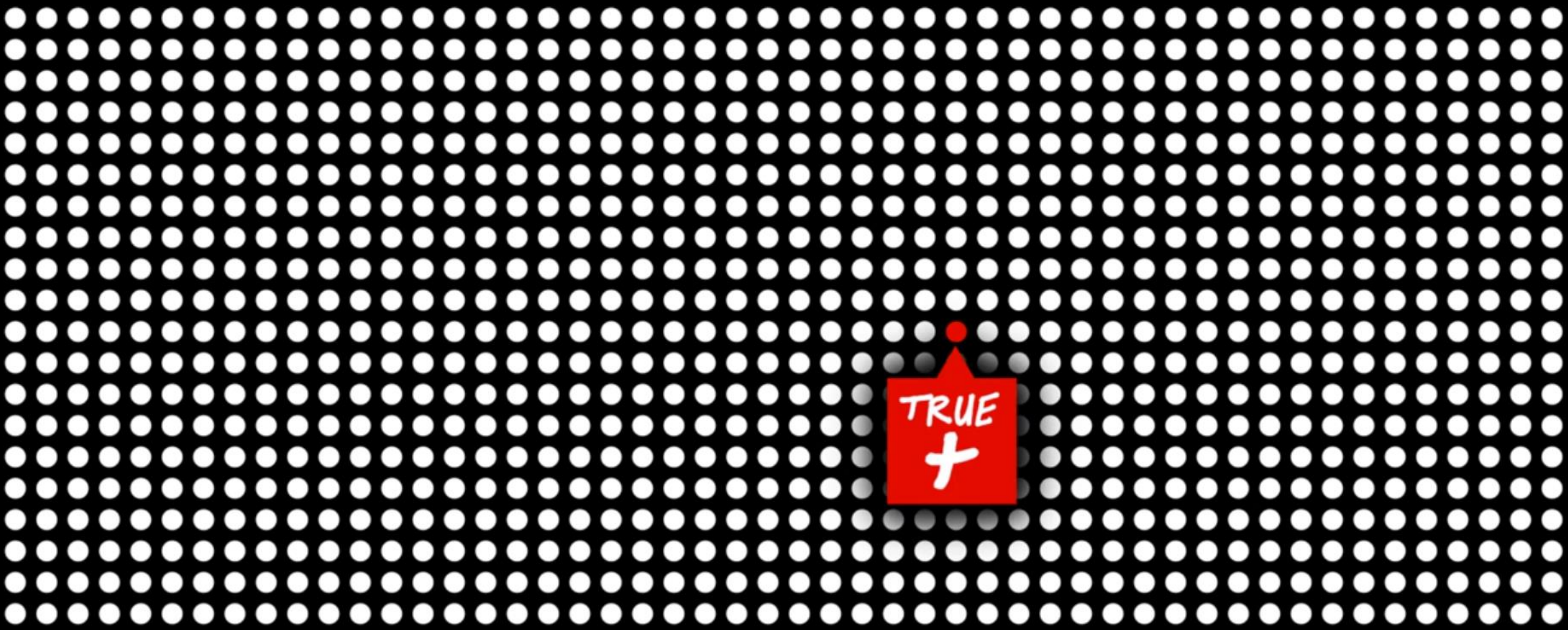
$$posteriori = \frac{wiarygodność \cdot apriori}{normalizacja}$$

Szkic kodu 000_Bayes_Theorem.py

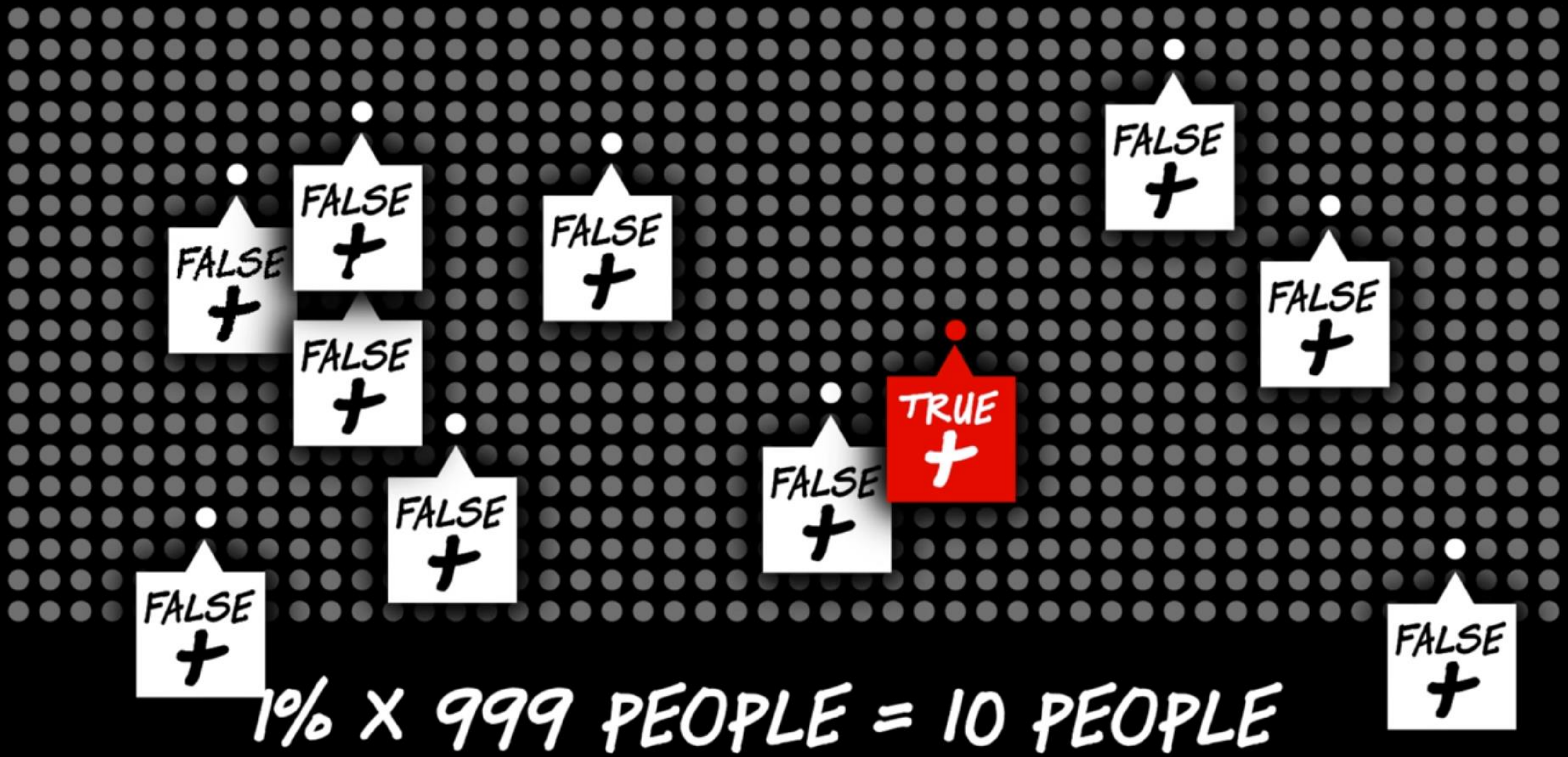


1,000 PEOPLE

https://www.youtube.com/watch?v=R13BD8qKeTg& , Veritasium ‚The Bayesian Trap'



1,000 PEOPLE

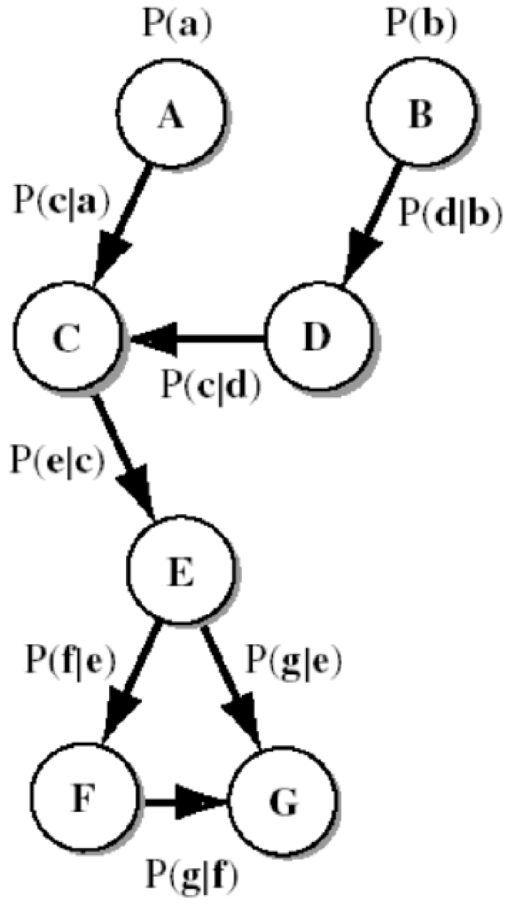

FALSE +
FALSE +
FALSE +
FALSE +
FALSE +
FALSE +
FALSE +
TRUE +
FALSE +
FALSE +
FALSE +
1% X 999 PEOPLE = 10 PEOPLE

1 IN 11 PEOPLE = 9%

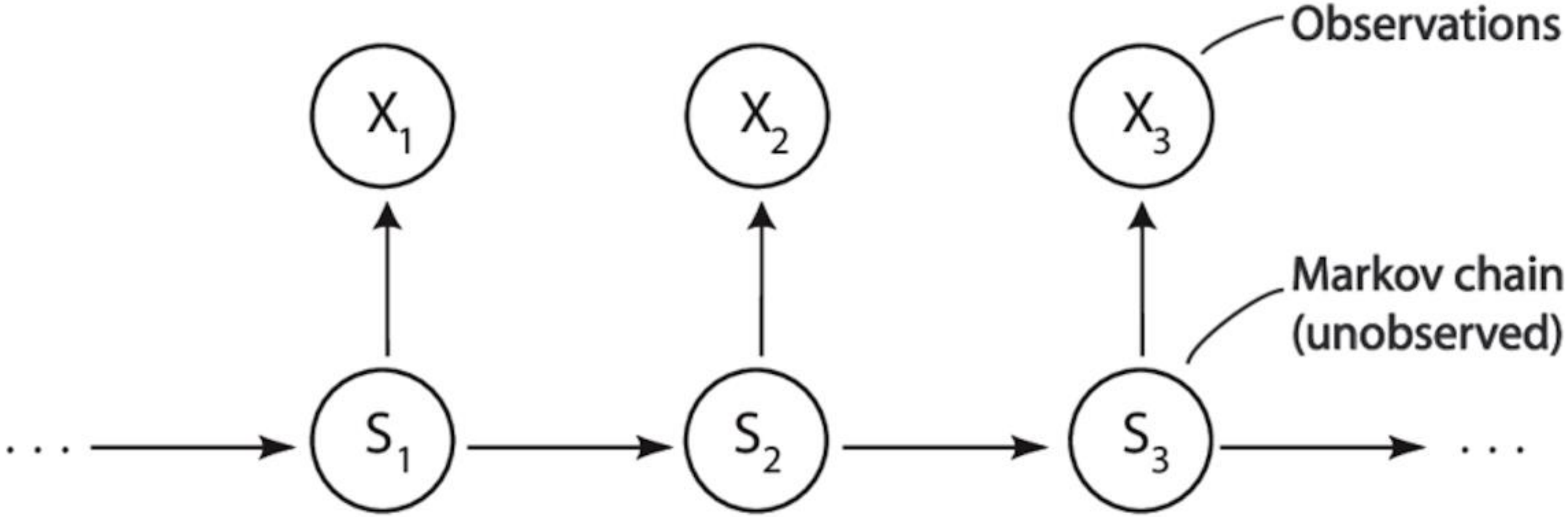
Sieci Bayesowskie

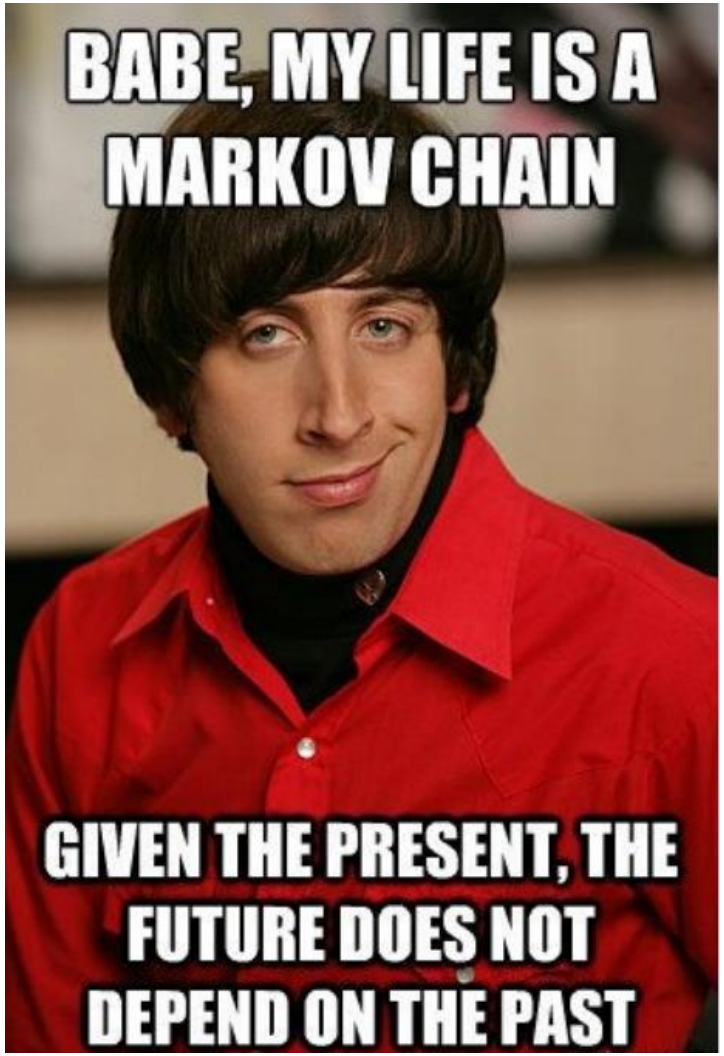
Sieć Bayesowska jest grafem, który:

- Zbiór zmiennych losowych X, Y, ... jest reprezentowany poprzez węzły,
- Krawędzie (strzałki) łączą poszczególne węzły.
  Znaczenie strzałki pomiędzy węzłem X i Y:
  X ma bezpośredni wpływ na Y.
- Każdy węzeł posiada tablice prawdopodobieństw warunkowych swoich poprzedników.
- Graf nie posiada cykli określonych przez strzałki.



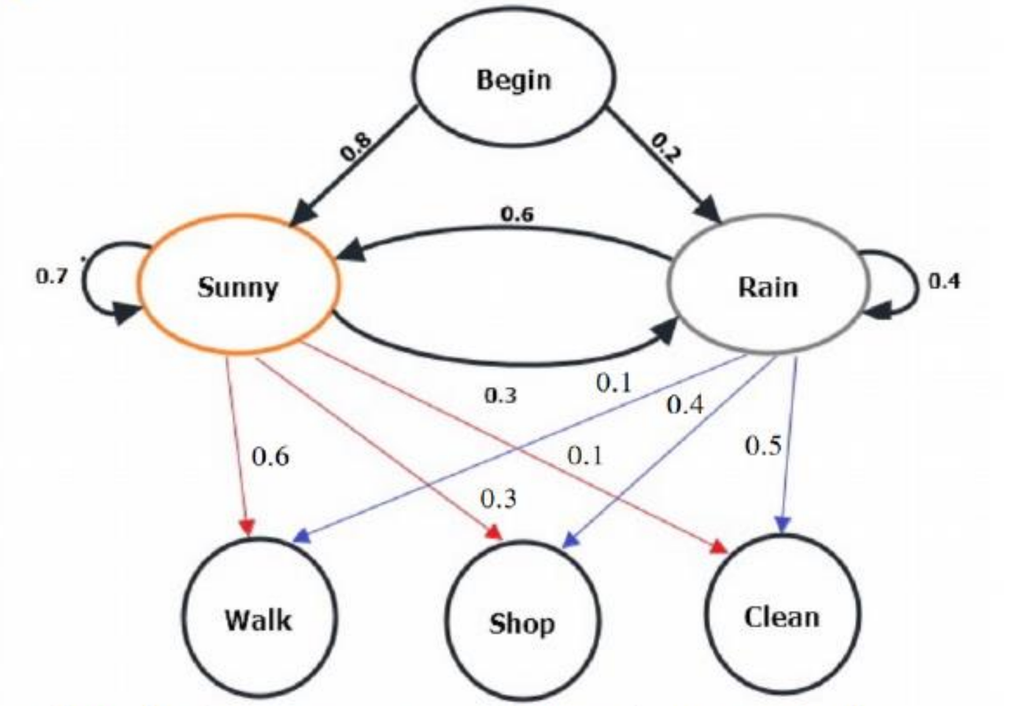
Ukryte łańcuchy Markowa





Zadanie 2

HMM przykład



- Możliwe są 3 zajęcia: spacer, zakupy i sprzątanie.
- Wybór, zależy wyłącznie od pogody danego dnia - *macierz prawdopodobieństwa emisji.*
- Pogoda na następny dzień zależy od pogody z poprzedniego dnia - *macierz prawdopodobieństwa przejścia*

Szkic kod 001_Bayesian_network.py

Wektor stacjonarny macierzy przejść – analityczne podejście

$$\bar{\rho} = (x, y, 1-x-y) \qquad \Pi = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{4} & 0 & \frac{3}{4} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$$

$$(x, y, 1-x-y) \cdot \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{4} & 0 & \frac{3}{4} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} = (x, y, 1-x-y)$$

$$\begin{cases} \frac{1}{2}x + \frac{1}{4}y + \frac{1}{3}(1-x-y) = x \\ \frac{1}{2}x + \frac{1}{3}(1-x-y) = y \\ \frac{3}{4}y + \frac{1}{3}(1-x-y) = 1-x-y \end{cases} \rightarrow \begin{cases} x = \frac{10}{27} \\ y = \frac{8}{27} \end{cases} \rightarrow \bar{\rho} = \left(\frac{10}{27}, \frac{8}{27}, \frac{9}{27}\right)$$

Zadanie 3

Napisz program, który symuluje proces wyboru posiłków w bistro i oblicza prawdopodobieństwo wystąpienia każdego z posiłków na podstawie zebranych danych.

- Stwórz początkowy wektor stanów (a priori), zakładając, że każdy posiłek ma równą szansę na wybór.
- Ustal macierz przejść (transition matrix), która określa prawdopodobieństwo wyboru kolejnego posiłku na podstawie aktualnego posiłku.

- Przeprowadź symulację wyboru posiłków w bistro dla 10 000 prób. W każdej iteracji:
  - Wylosuj kolejny posiłek na podstawie macierzy przejść.
  - Zwiększ licznik wystąpień danego posiłku.
  - Zaktualizuj poprzedni posiłek.
- Oblicz prawdopodobieństwo wystąpienia każdego z posiłków na podstawie zebranych 10000 danych.

- Wypełnij macierz przejść na podstawie zdefiniowanej macierzy transition_matrix, następnie:
  - Iteracyjnie oblicz wektor stacjonarny, który odpowiada stabilnemu rozkładowi prawdopodobieństwa stanów procesu.
  - Skorzystaj z funkcji np.linalg.eig do obliczenia wektora stacjonarnego

Warto zauważyć, że stosowanie macierzy przejść w tym zadaniu umożliwia symulację procesu wyboru posiłków oraz obliczenie wektora stacjonarnego, co pozwala na określenie stabilnego rozkładu prawdopodobieństwa stanów tego procesu.

Szkic kod 002_stationary_vector.py

Projekt 1

- Napisz program, uczący się gry “papier, kamień, nożyce”:
  - Komputer ma zadaną taktykę w postaci macierzy przejść:

```
transition_matrix_computer = {
    "Paper": {"Paper": 2/3, "Rock": 1/3, "Scissors": 0/3},
    "Rock": {"Paper": 0/3, "Rock": 2/3, "Scissors": 1/3},
    "Scissors": {"Paper": 2/3, "Rock": 0/3, "Scissors": 1/3} }
```

  tj. na podstawie swojego poprzedniego wyboru wybiera aktualny „papier, kamień, nożyce”
- Taktyka gracza to (napisz kod dla obu wersji):
  - Wersja 1: bazując na wektorze stacjonarnym transition_matrix_computer
  - Wersja 2: macierz przejść jest aktualizowana w trakcie gry
- Wartość wypłaty dla gracza to:
  - 1 w przypadku wygranej,
  - −1 w przypadku przegranej
  - 0 w przypadku remisu,
- Przeprowadź ciąg 10 000 gier “kamień, papier, nożyce” i sporządź wykres jak zmienia się stan kasy gracza w każdym kroku gry.

Przykładowy szkic: Projekt_01_papier_nozyce_kamien.py, NIE JEST OBOWIĄZKOWY