Metody bioinformatyki (MBI)

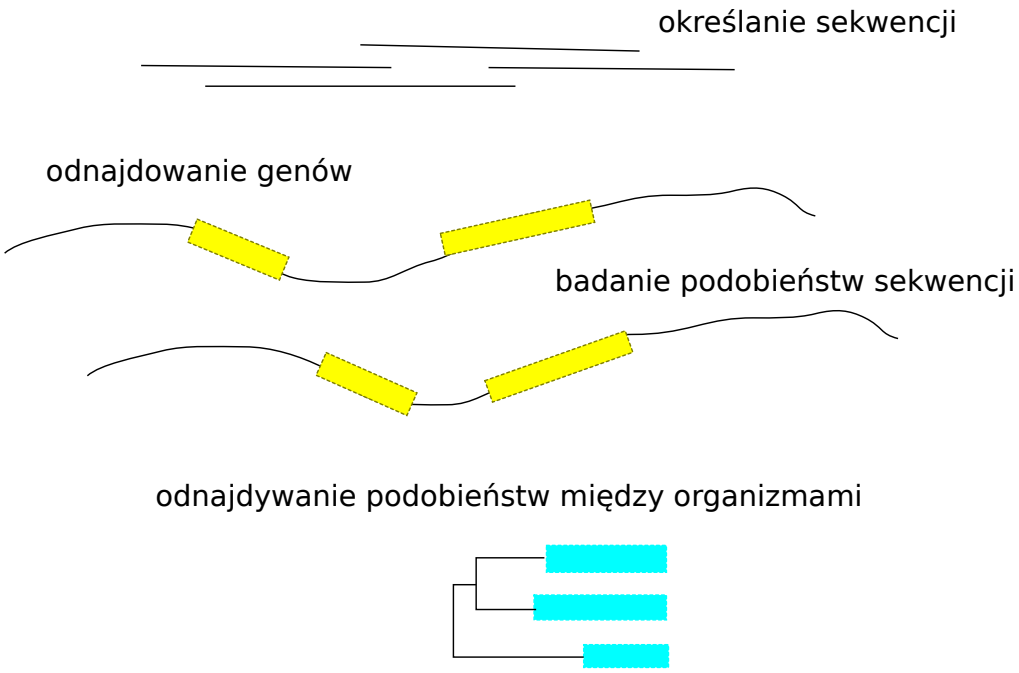
Wykład 7 - wykrywanie sygnałów, ukryte modele Markowa

Robert Nowak

2024L

Sekwencje biologiczne

- badanie podobieństw
- bioinformatyczne bazy danych
- badanie markerów genetycznych
- **wyszukiwanie genów, sekwencji regulujących itp.**,
- przewidywanie struktur przestrzennych i funkcji



Sekwencje spokrewnione

- wykorzystywanie profili
  - tworzenie profilu

A	0.75	0	0	0	1	0.75	0	0	0	0.75
C	0	0	0	0	0	0	1	0.75	0	0
G	0	0.25	1	0	0	0.25	0	0.25	1	0.25
T	0.25	0.75	0	1	0	0	0	0	0	0

  - dopasowanie sekwencji do profilu, algorytm programowania dynamicznego, gdzie ocena

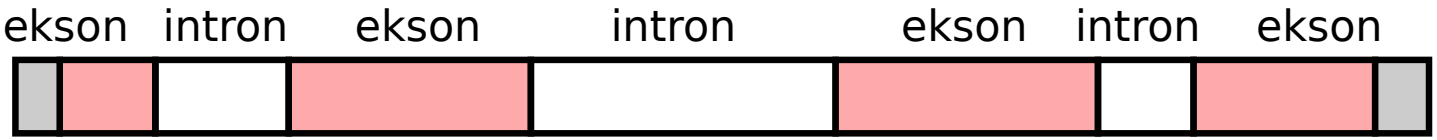
$$e(s_i, t_j) = \sum_{a \in \Sigma} p(i, a) * e(a, t_j)$$

    Złożoność: $O(N * M * |\Sigma|)$

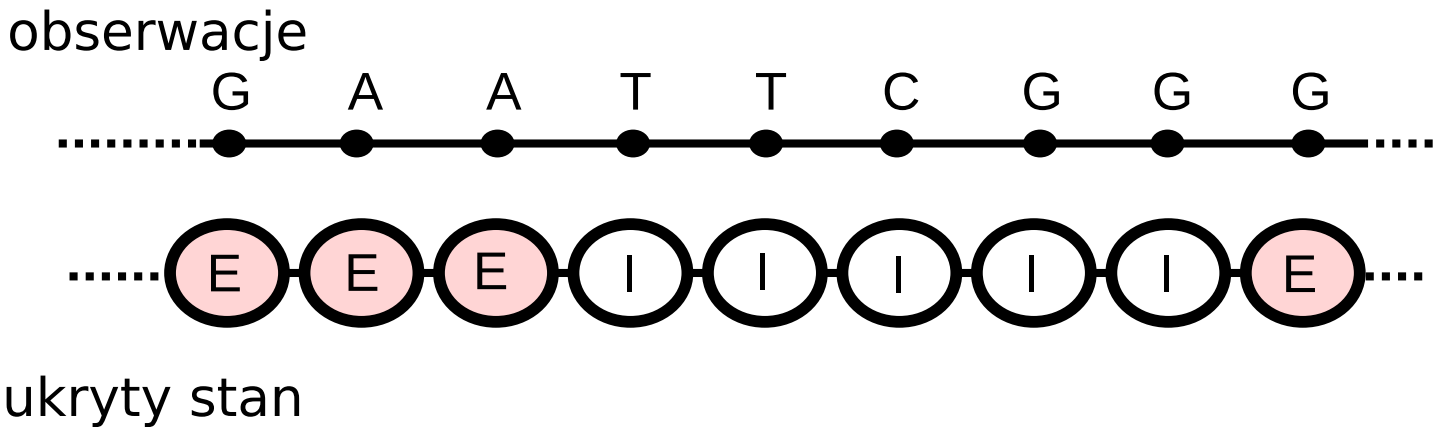
- wykorzystywanie ukrytych modeli Markowa

Ukryte modele Markowa

Cząsteczka DNA, reprezentująca gen ma odcinki kodujące gen (eksony) oddzielone sekwencjami niekodującymi (introny).



Ukryty model Markowa może dostarczyć informację dla poszczególnych symboli:



Przykład - wyspy CpG

- kolejne nukleotydy CG na tej samej nici, zapis CpG, w odróżnieniu od pary C-G (komplementarne nukleotydy na dwu niciach DNA)
- CpG jest przekształcana (w procesie metylacji) na TpG
- wyjątki - regiony otaczające początek sekwencji kodującej gen

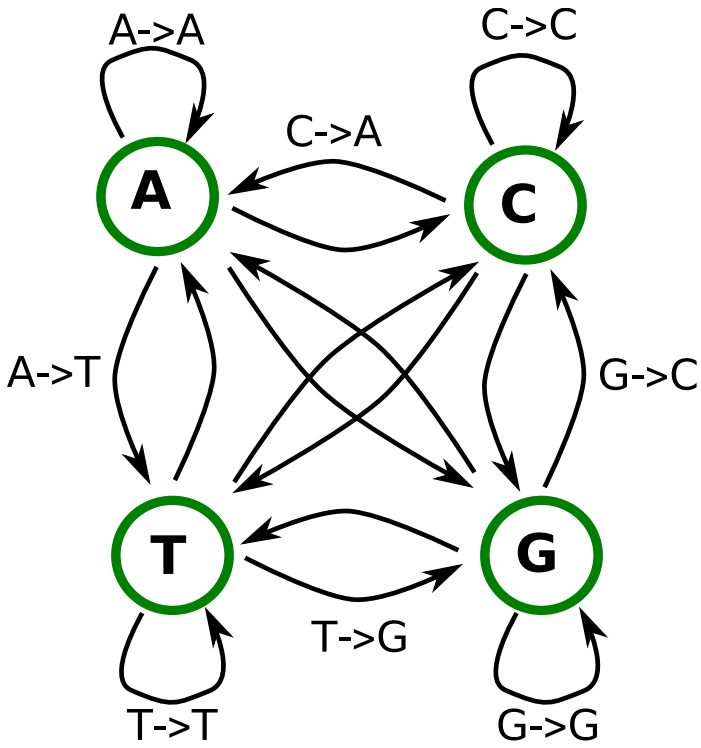
Problem: jak odnaleźć wyspę CpG (długość 100 – 3000 zasad)?

Łańcuchy Markowa, automat probabilistyczny

Q skończony zbiór stanów
P prawdopodobieństwa stanów początkowych $\sum_{q \in Q} P_q = 1$
A macierz przejść $\sum_{q \in Q} a(p, q) = 1$

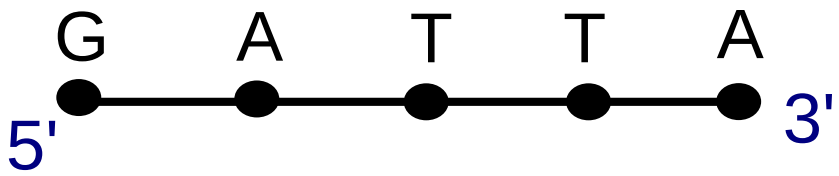
- stany: $\mathbf{Q = \{A, C, G, T\}}$
- prawdopodobieństwa początkowe: P_A, P_C, P_G, P_T
- **A** prawdopodobieństwa przejść

$$P(s) = P(x_0) \prod_{i=1}^{n-1} A_{x_{i-1} x_i}$$



Łańcuchy Markowa (2)

$$
\begin{aligned}
P(s) &= P(x_0 x_1 \ldots x_{n-1}) \\
&= P(x_0)P(x_1|x_0)\ldots P(x_{n-2}|x_0\ldots x_{n-3})P(x_{n-1}|x_0\ldots x_{n-2}) \\
P(GATTA) &= P(x_0 = G)P(x_1 = A|x_0 = G)P(x_2 = T|x_{01} = GA) \\
&\quad P(x_3 = T|x_{012} = GAT)P(x_4 = A|x_{0123} = GATT)
\end{aligned}
$$



łańcuch Markowa: symbol zależny tylko od poprzedniego

$$
\begin{aligned}
P(s) &= P(x_0)P(x_1|x_0)\ldots P(x_{n-2}|x_{n-3})P(x_{n-1}|x_{n-2}) \\
&= P(x_0)\prod_{i=1}^{n-1} P(x_i|x_{i-1}) \\
P(GATTA) &= P(G)P(A|G)P(T|A)P(T|T)P(A|T)
\end{aligned}
$$

wykorzystanie łańcuchów Markowa

- dany zbiór trenujący
  - znane wyspy CpG
  - regiony nie będące wyspami CpG
- obliczane dwa zestawy parametrów

$$a_{st}^{+} = \frac{c_{st}^{+}}{\sum_{t'} c_{st'}^{+}}$$

c_{st}^{+} przypadki gdy t następuje po s dla wysp CpG

$$a_{st}^{-} = \frac{c_{st}^{-}}{\sum_{t'} c_{st'}^{-}}$$

c_{st}^{-} przypadki gdy t następuje po s poza wyspami CpG

przykład (48 ludzkich wysp CpG)

+	A	C	G	T
A	0.18	0.27	0.43	0.12
C	0.17	0.37	0.27	0.19
G	0.16	0.34	0.37	0.13
T	0.08	0.36	0.38	0.18

-	A	C	G	T
A	0.30	0.20	0.29	0.21
C	0.32	0.30	0.08	0.30
G	0.25	0.25	0.30	0.20
T	0.18	0.24	0.30	0.30

w pierwszym rzędzie prawd., że po A następuje A, C, G, T

Czy sekwencja jest wyspą CpG?

- obliczenie logarytmu prawdopodobieństwa, że sekwencja jest wyspą CpG

$$\log P^+(s) = \log(\prod_{i=1}^{n-1} a^+_{x_{i-1}x_i}) = \sum_{i=1}^{n-1} \log(a^+_{x_{i-1}x_i})$$

- obliczenie logarytmu prawdopodobieństwa, że sekwencja jest poza wyspą CpG

$$\log P^-(s) = \sum_{i=1}^{n-1} \log(a^-_{x_{i-1}x_i})$$

- porównanie tych wielkości

Wyszukiwanie wysp CpG w sekwencji DNA

Wejście:

- sekwencja DNA
- określone modele Markowa dla wysp CpG i pozostałych sekwencji

Algorytm:

- podział sekwencji na okna (o długości np. 100 nukleotydów)
- klasyfikacja każdego okna

Problemy:

- wyspy CpG mogą mieć różną długość
- dobór wielkości okna

Ukryty model Markowa (hidden Markov model, HMM)

automat probabilistyczny z wyjściem

- Q - skończony zbiór stanów
- P - prawdopodobieństwa stanów początkowych
- A - macierz przejść $|Q| \times |Q|$

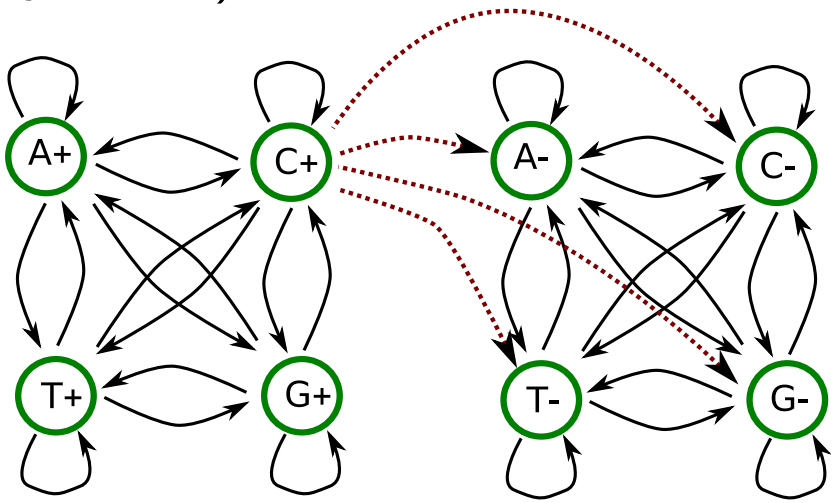
$$\sum_{q \in Q} a(p, q) = 1$$

- V - skończony zbiór obserwacji
- E - macierz emisji $|Q| \times |V|$

$$\sum_{v \in V} e(q, v) = 1$$

HMM dla wysp CpG

nie jest możliwe bezpośrednie podanie stanu obserwując wyjścia (symbole)



$$Q = \{A+, C+, G+, T+, A-, C-, G-, T-\}$$

$$V = \{A, C, G, T\}$$

Macierz emisji:

	A+	C+	G+	T+	A-	C-	G-	T-
A	1	0	0	0	1	0	0	0
C	0	1	0	0	0	1	0	0
G	0	0	1	0	0	0	1	0
T	0	0	0	1	0	0	0	1

Problem dekodowania

znaleźć najbardziej prawdopodobny ciąg stanów, który doprowadził do wyemitowania danego ciągu symboli

Przykład:

- Obserwowana sekwencja: CGCG
- Możliwe ciągi stanów : $C^+G^+C^+G^+, C^+G^+C^+G^-, ..., C^-G^-C^-G^-$

Rozwiązanie:

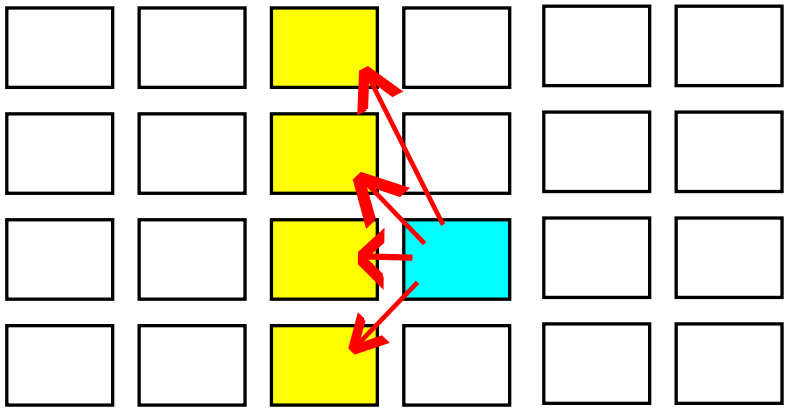
- dla każdego ciągu obliczyć jego prawdopodobieństwo

$$P(q_1 q_2 q_3 ... q_n) = P_{q_1} \prod_{i=2}^{n} e(q_i, x_i) a(q_{i-1}, q_i)$$

- zwrócić ciąg o maksymalnym prawdopodobieństwie
- **problem - wykładniczo rośnie liczba ciągów**

Algorytm Viterbiego

- wejście
  - opis HMM : macierz przejść a, macierz emisji e
  - wyemitowany ciąg symboli $S = x_1 x_2 x_3 ... x_n$
- wyjście: najbardziej prawdopodobny ciąg stanów
- algorytm: programowanie dynamiczne



- $V_q(i)$ prawdopodobieństwo, że szukany ciąg stanów przechodzi przez stan q dla symbolu o indeksie i

Algorytm Viterbiego

- wejście
  - macierz przejść a
  - macierz emisji e
  - ciąg obserwowanych symboli $S = x_1 x_2 x_3 ... x_n$
- inicjacja

$$V_q(1) = P_q * e(q, x_1)$$

- krok rekurencyjny

$$V_q(i+1) = e(q, x_{i+1}) * \max_{p \in Q}(V_p(i) * a(p, q))$$

- stan końcowy

$$\arg\max_{q \in Q} V_q(n)$$

Algorytm Viterbiego - przykład

Macierz przejść:

	A+	C+	G+	T+	A-	C-	G-	T-
A+	0.17	0.257	0.409	0.114	0.009	0.014	0.021	0.006
C+	0.162	0.351	0.257	0.18	0.009	0.018	0.014	0.009
G+	0.152	0.323	0.352	0.123	0.008	0.017	0.019	0.006
T+	0.076	0.343	0.361	0.17	0.004	0.018	0.019	0.009
A-	0.03	0.02	0.029	0.021	0.27	0.18	0.261	0.189
C-	0.031	0.03	0.008	0.03	0.29	0.27	0.071	0.27
G-	0.025	0.025	0.03	0.02	0.225	0.225	0.27	0.18
T-	0.018	0.024	0.03	0.023	0.162	0.212	0.27	0.261

Macierz emisji:

	A+	C+	G+	T+	A-	C-	G-	T-
A	1	0	0	0	1	0	0	0
C	0	1	0	0	0	1	0	0
G	0	0	1	0	0	0	1	0
T	0	0	0	1	0	0	0	1

$P(A+) = P(C+) = \ldots\ P(T-) = 0.125$

Algorytm Viterbiego - przykład

przykład: sekwencja CGCG

$P_q = 0.125$ dla każdego q

	C+	G+	C-	G-
C+	0.351	0.257	0.018	0.014
G+	0.323	0.352	0.017	0.019
C-	0.03	0.008	0.27	0.071
G-	0.025	0.03	0.225	0.27

	C	G	C	G
A+	0			
C+	0.125			
G+	0			
T+	0			
A-	0			
C-	0.125			
G-	0			
T-	0			

Algorytm Viterbiego - przykład

przykład: sekwencja CGCG

$P_q = 0.125$ dla każdego q

	C+	G+	C-	G-
C+	0.351	0.257	0.018	0.014
G+	0.323	0.352	0.017	0.019
C-	0.03	0.008	0.27	0.071
G-	0.025	0.03	0.225	0.27

	C	G	C	G
A+	0			
C+	0.125			
G+	0			
T+	0			
A-	0			
C-	0.125			
G-	0			
T-	0			

$$A^+ \to A^+ : 0 \cdot 0 \cdot 0.17 = 0$$

$$\dots$$

$$C^+ \to G^+ : 0.125 \cdot 1 \cdot 0.257 \approx 0.0321$$

$$C^+ \to G^- : 0.125 \cdot 1 \cdot 0.014 \approx 0.0018$$

$$C^- \to G^+ : 0.125 \cdot 1 \cdot 0.008 = 0.001$$

$$C^- \to G^- : 0.125 \cdot 1 \cdot 0.071 \approx 0.0089$$

Algorytm Viterbiego - przykład

przykład: sekwencja CGCG

$P_q = 0.125$ dla każdego q

	C+	G+	C-	G-
C+	0.351	0.257	0.018	0.014
G+	0.323	0.352	0.017	0.019
C-	0.03	0.008	0.27	0.071
G-	0.025	0.03	0.225	0.27

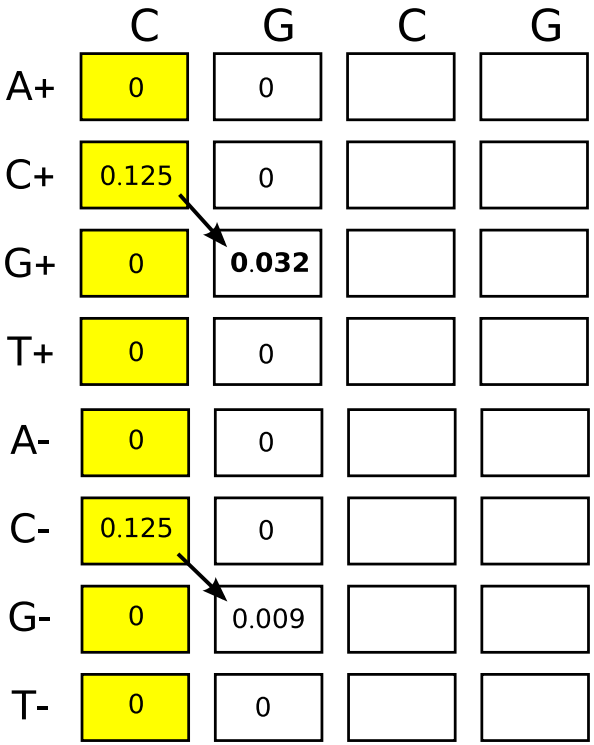
	C	G	C	G
A+	0	0		
C+	0.125	0		
G+	0	**0.032**		
T+	0	0		
A-	0	0		
C-	0.125	0		
G-	0	0.009		
T-	0	0		

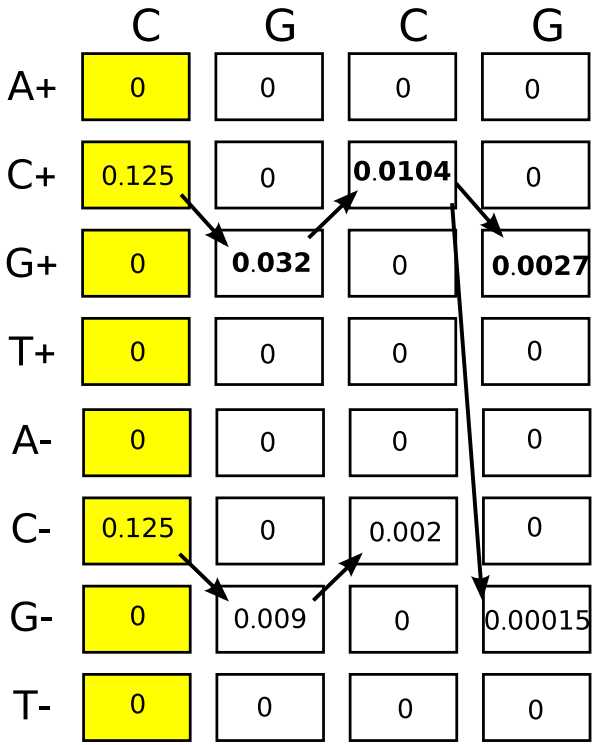
Algorytm Viterbiego - przykład

przykład: sekwencja CGCG

$P_q = 0.125$ dla każdego q

	C+	G+	C-	G-
C+	0.351	0.257	0.018	0.014
G+	0.323	0.352	0.017	0.019
C-	0.03	0.008	0.27	0.071
G-	0.025	0.03	0.225	0.27



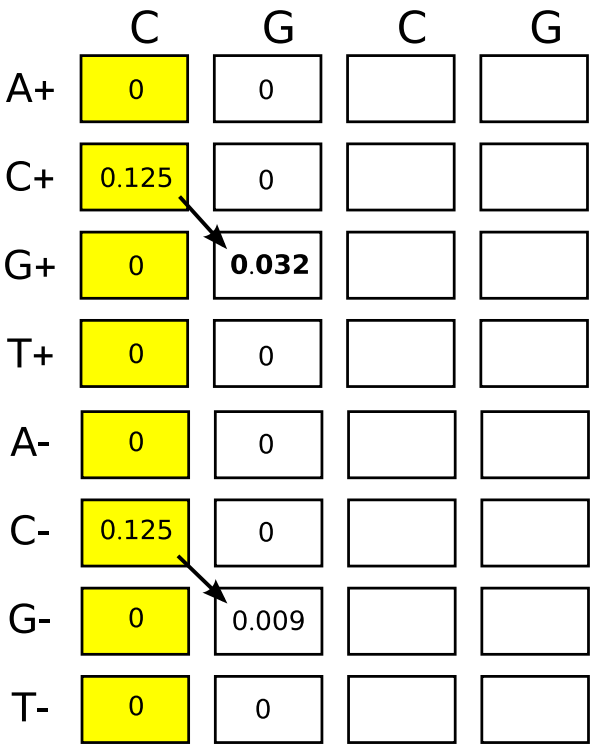


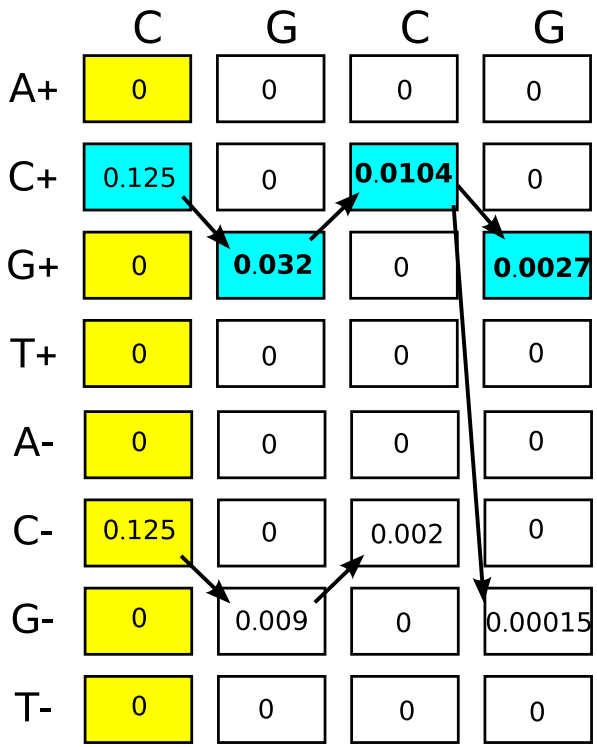
Algorytm Viterbiego - przykład

przykład: sekwencja CGCG

$P_q = 0.125$ dla każdego q

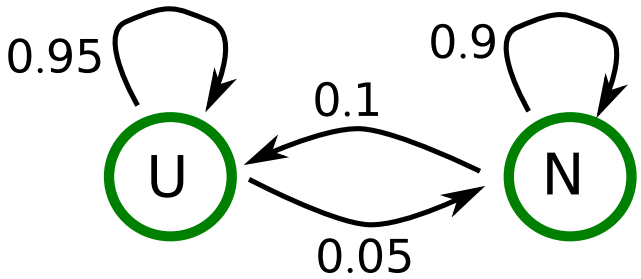
	C+	G+	C-	G-
C+	0.351	0.257	0.018	0.014
G+	0.323	0.352	0.017	0.019
C-	0.03	0.008	0.27	0.071
G-	0.025	0.03	0.225	0.27





nieuczciwe kasyno (inny przykład HMM)

- uczciwa kostka
- nieuczciwa kostka
  $P(1) = P(2) = P(3) = P(4) = P(5) = \frac{1}{10}, P(6) = \frac{1}{2}$



$$
\begin{aligned}
Q &= \{U, N\} \\
P_U &= 0.5, P_N = 0.5 \\
V &= \{1, 2, 3, 4, 5, 6\}
\end{aligned}
$$

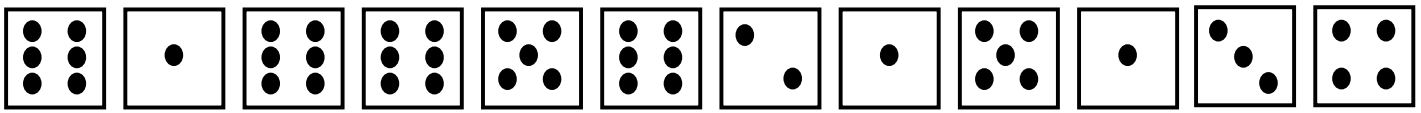
	U	N
U	0.95	0.05
N	0.1	0.9

E	U	N
1	0.167	0.1
2	0.167	0.1
3	0.167	0.1
4	0.167	0.1
5	0.167	0.1
6	0.167	0.5

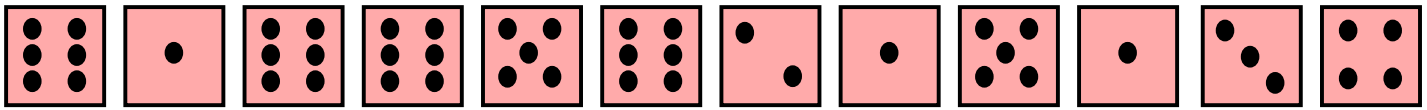
nieuczciwe kasyno: prawdopodobieństwa ciągów stanów

	U	N
U	0.95	0.05
N	0.1	0.9

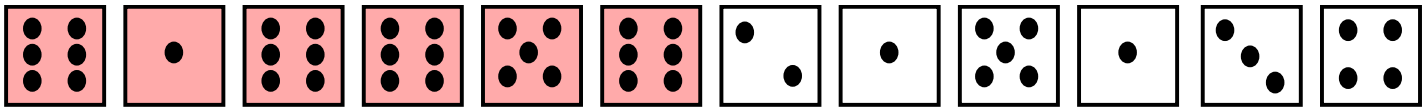
$$e(U,v) = \frac{1}{6}, e(N,v) = \begin{cases} \frac{1}{10} & : v \neq 6 \\ \frac{1}{2} & : v = 6 \end{cases}$$



$$P_1 = \frac{1}{2} * \frac{1}{6} * 0.95 * \ldots = \frac{1}{2} * (\frac{1}{6})^{12} * (0.95)^{11} \approx 1.3 * 10^{-10}$$

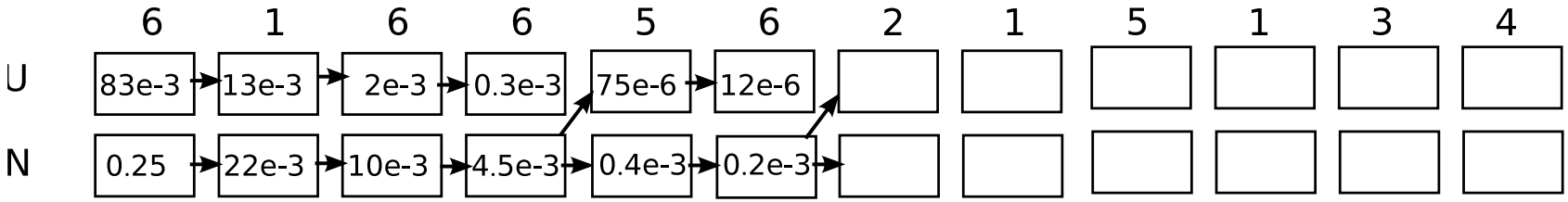


$$P_2 = \frac{1}{2} * (\frac{1}{10})^{8} * (\frac{1}{2})^{4} * (0.9)^{11} \approx 0.98 * 10^{-10}$$



$$P_3 = \frac{1}{2} * 0.9^5 * 0.1 * 0.95^5 * (\frac{1}{6})^{6} * (\frac{1}{2})^{4} * (\frac{1}{10})^{2} \approx 3.06 * 10^{-10}$$

Nieuczciwe kasyno - algorytm Viterbiego



Algorytm Viterbiego

- wykorzystuje programowanie dynamiczne
- oblicza najbardziej prawdopodobną sekwencję stanów
- złożoność pamięciowa $\mathbf{O}(|\mathbf{X}| * |\mathbf{Q}|)$
- złożoność czasowa $\mathbf{O}(|\mathbf{X}| * |\mathbf{Q}|^2)$
- zazwyczaj posługujemy się logarytmem prawdopodobieństwa
  - dodawanie zamiast mnożenia
  - brak bardzo małych liczb

Algorytm Viterbiego - zadanie

Posługujemy się trzema monetami, jedna jest uczciwa, obserwując sekwencje rzutów (orły i reszki). Zakładając, że przedstawione doświadczenie jest opisywane ukrytym modelem Markowa przedstawionym obok, podaj najbardziej prawdopodobną sekwencję stanów (sekwencję użytych monet), jeżeli wynikiem doświadczenia jest sekwencja *RRRR*.

$Q = \{U, N1, N2\}$
$V = \{O, R\}$
$P_U = 1$
$P_{N1} = 0$
$P_{N2} = 0$

	U	**N1**	**N2**
U	$\frac{2}{3}$	$\frac{1}{3}$	0
N1	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
N2	0	$\frac{1}{3}$	$\frac{2}{3}$

	O	**R**
U	$\frac{1}{2}$	$\frac{1}{2}$
N1	$\frac{1}{4}$	$\frac{3}{4}$
N2	0	1

Algorytm Viterbiego - zadanie

Posługujemy się trzema monetami, jedna jest uczciwa, obserwując sekwencje rzutów (orły i reszki). Zakładając, że przedstawione doświadczenie jest opisywane ukrytym modelem Markowa przedstawionym obok, podaj najbardziej prawdopodobną sekwencję stanów (sekwencję użytych monet), jeżeli wynikiem doświadczenia jest sekwencja *RRRR*.

$Q = \{U, N1, N2\}$
$V = \{O, R\}$
$P_U = 1$
$P_{N1} = 0$
$P_{N2} = 0$

	U	**N1**	**N2**
U	$\frac{2}{3}$	$\frac{1}{3}$	0
N1	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
N2	0	$\frac{1}{3}$	$\frac{2}{3}$

	O	**R**
U	$\frac{1}{2}$	$\frac{1}{2}$
N1	$\frac{1}{4}$	$\frac{3}{4}$
N2	0	1

	R	**R**	**R**	**R**
U	$\frac{1}{2}$			
N1	0			
N2	0			

Algorytm Viterbiego - zadanie

Posługujemy się trzema monetami, jedna jest uczciwa, obserwując sekwencje rzutów (orły i reszki). Zakładając, że przedstawione doświadczenie jest opisywane ukrytym modelem Markowa przedstawionym obok, podaj najbardziej prawdopodobną sekwencję stanów (sekwencję użytych monet), jeżeli wynikiem doświadczenia jest sekwencja *RRRR*.

$Q = \{U, N1, N2\}$
$V = \{O, R\}$
$P_U = 1$
$P_{N1} = 0$
$P_{N2} = 0$

	U	**N1**	**N2**
U	$\frac{2}{3}$	$\frac{1}{3}$	0
N1	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
N2	0	$\frac{1}{3}$	$\frac{2}{3}$

	O	**R**
U	$\frac{1}{2}$	$\frac{1}{2}$
N1	$\frac{1}{4}$	$\frac{3}{4}$
N2	0	1

	R	**R**	**R**	**R**
U	$\frac{1}{2}$	$\frac{1}{6}$		
N1	0	$\frac{1}{8}$		
N2	0	0		

Algorytm Viterbiego - zadanie

Posługujemy się trzema monetami, jedna jest uczciwa, obserwując sekwencje rzutów (orły i reszki). Zakładając, że przedstawione doświadczenie jest opisywane ukrytym modelem Markowa przedstawionym obok, podaj najbardziej prawdopodobną sekwencję stanów (sekwencję użytych monet), jeżeli wynikiem doświadczenia jest sekwencja $RRRR$.

$Q = \{U, N1, N2\}$
$V = \{O, R\}$
$P_U = 1$
$P_{N1} = 0$
$P_{N2} = 0$

	U	**N1**	**N2**
U	$\frac{2}{3}$	$\frac{1}{3}$	0
N1	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
N2	0	$\frac{1}{3}$	$\frac{2}{3}$

	O	**R**
U	$\frac{1}{2}$	$\frac{1}{2}$
N1	$\frac{1}{4}$	$\frac{3}{4}$
N2	0	1

	R	**R**	**R**	**R**
U	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{18}$	$\frac{1}{54}$
N1	0	$\frac{1}{8}$	$\frac{1}{24}$	$\frac{1}{72}$
N2	0	0	$\frac{1}{24}$	$\frac{1}{36}$

Algorytm Viterbiego - zadanie

Posługujemy się trzema monetami, jedna jest uczciwa, obserwując sekwencje rzutów (orły i reszki). Zakładając, że przedstawione doświadczenie jest opisywane ukrytym modelem Markowa przedstawionym obok, podaj najbardziej prawdopodobną sekwencję stanów (sekwencję użytych monet), jeżeli wynikiem doświadczenia jest sekwencja *RRRR*.

$Q = \{U, N1, N2\}$
$V = \{O, R\}$
$P_U = 1$
$P_{N1} = 0$
$P_{N2} = 0$

	U	**N1**	**N2**
U	$\frac{2}{3}$	$\frac{1}{3}$	0
N1	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
N2	0	$\frac{1}{3}$	$\frac{2}{3}$

	O	**R**
U	$\frac{1}{2}$	$\frac{1}{2}$
N1	$\frac{1}{4}$	$\frac{3}{4}$
N2	0	1

$U\ N_1\ N_2\ N_2$

Prawdopodobieństwo sekwencji symboli

$$P(s) = \sum_{\pi} P(\pi) \text{ gdzie sekw. stanów } \pi \text{ daje sekw. symboli } s$$

Problem: średnia liczba sekwencji stanów dla danej sekwencji symboli rośnie wykładniczo wraz z długością sekwencji

Rozwiązania

- uwzględniać tylko najbardziej prawdopodobną sekwencję stanów (zwracaną przez alg. Viterbiego)
- wykorzystać algorytm prefiksowy (forward algorithm)
- wykorzystać algorytm sufiksowy (backward algorithm)

Algorytm prefiksowy (programowanie dynamiczne)

wejście: macierz przejść a, macierz emisji e
ciąg symboli wyjściowych $S = x_1 x_2 x_3 ... x_n$

inicjacja:

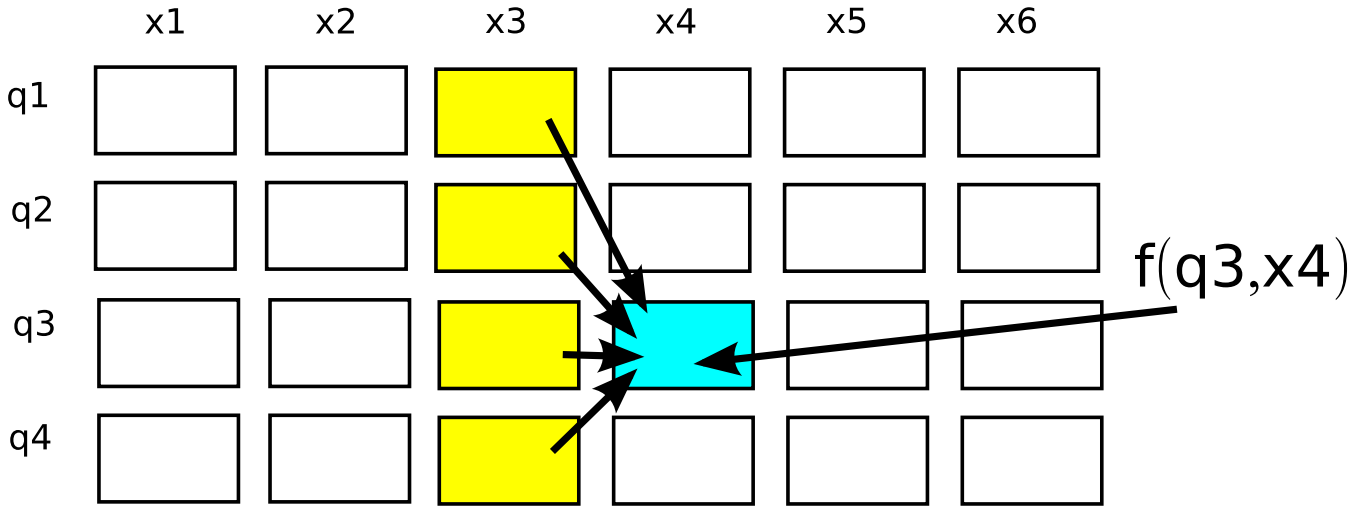
$$f_q(1) = P_q * e(q, x_1)$$

krok rekurencyjny:

$$f_q(i+1) = e(q, x_{i+1}) * \sum_{p \in Q} (f_p(i) * a(p, q))$$

prawd. sekwencji:

$$P(S) = \sum_{q \in Q} f_q(n)$$



Algorytm prefiksowy - przykład

Badanie wysp CpG

$P_q = \frac{1}{8}$ dla każdego q

P(CGCG) =?

	C+	G+	C-	G-
C+	0.351	0.257	0.018	0.014
G+	0.323	0.352	0.017	0.019
C-	0.03	0.008	0.27	0.071
G-	0.025	0.03	0.225	0.27

	A^+	C^+	G^+	T^+	A^-	C^-	G^-	T^-
A	1	0	0	0	1	0	0	0
C	0	1	0	0	0	1	0	0
G	0	0	1	0	0	0	1	0
T	0	0	0	1	0	0	0	1

	C	G	C	G
C+	0.125			
G+	0			
C-	0.125			
G-	0			

Algorytm prefiksowy - przykład

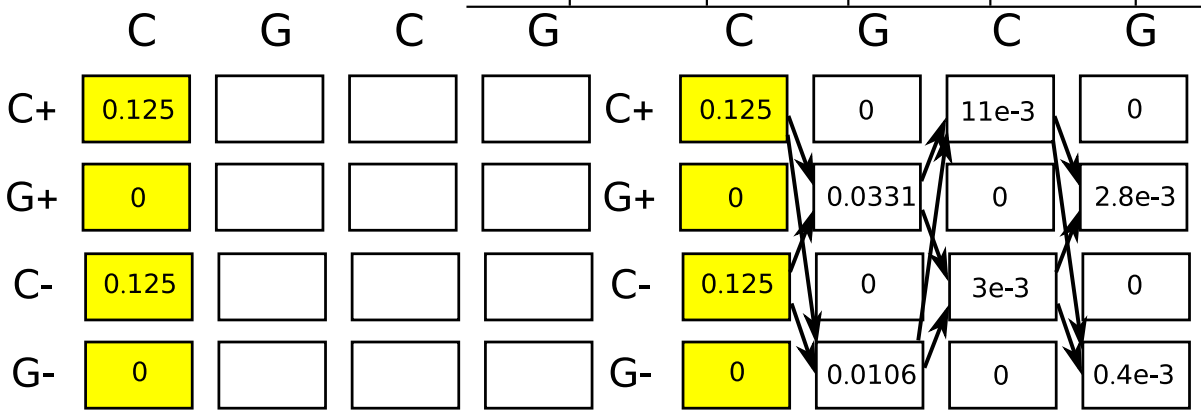
Badanie wysp CpG

$P_q = \frac{1}{8}$ dla każdego q

P(CGCG) =?

	C+	G+	C-	G-
C+	0.351	0.257	0.018	0.014
G+	0.323	0.352	0.017	0.019
C-	0.03	0.008	0.27	0.071
G-	0.025	0.03	0.225	0.27

	A^+	C^+	G^+	T^+	A^-	C^-	G^-	T^-
A	1	0	0	0	1	0	0	0
C	0	1	0	0	0	1	0	0
G	0	0	1	0	0	0	1	0
T	0	0	0	1	0	0	0	1



Algorytm prefiksowy - przykład

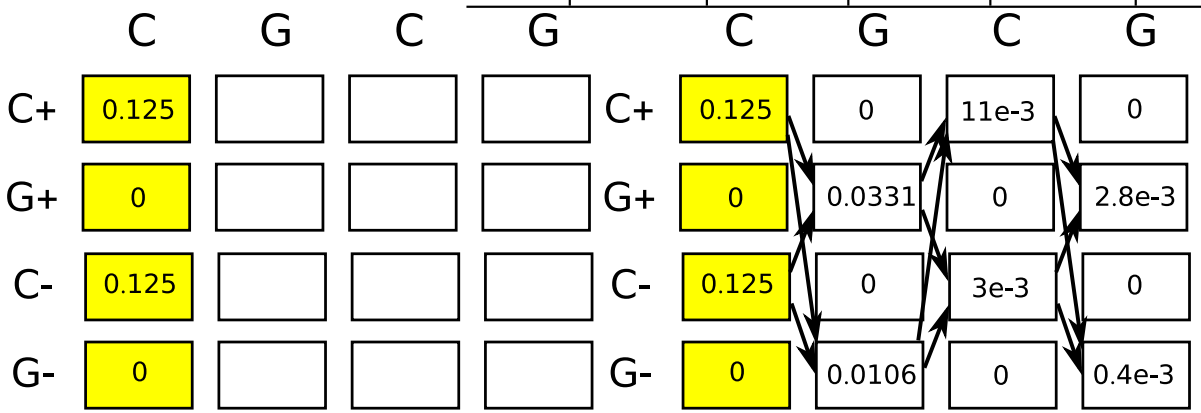
Badanie wysp CpG

$P_q = \frac{1}{8}$ dla każdego q

P(CGCG) =?

	C+	G+	C-	G-
C+	0.351	0.257	0.018	0.014
G+	0.323	0.352	0.017	0.019
C-	0.03	0.008	0.27	0.071
G-	0.025	0.03	0.225	0.27

	A^+	C^+	G^+	T^+	A^-	C^-	G^-	T^-
A	1	0	0	0	1	0	0	0
C	0	1	0	0	0	1	0	0
G	0	0	1	0	0	0	1	0
T	0	0	0	1	0	0	0	1



$P_{CGCG} = 0.0032$

$P_{C^+G^+C^+G^+} = 0.0027$

Algorytm sufiksowy (prawdopodobieństwo sekwencji)

wejście: macierz przejść a, macierz emisji e
sekw. symboli wejściowych $S = x_1x_2x_3...x_n$

$b_q(i)$ $P(x_{i+1}...x_n)$, gdy q jest stanem początkowym

inicjacja:

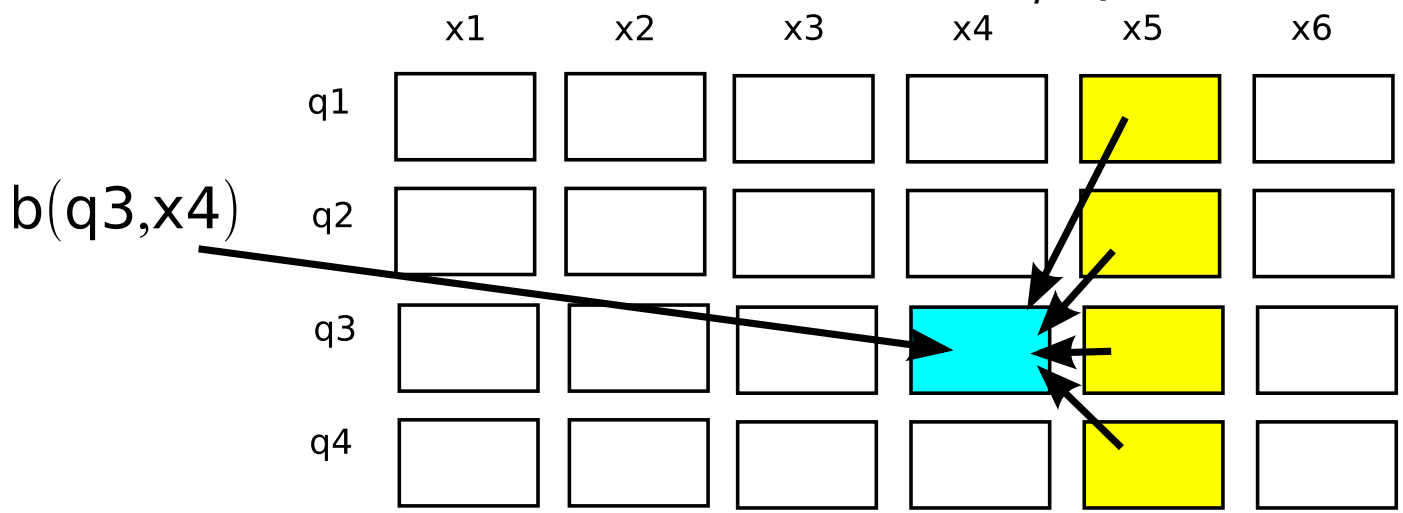
$$b_q(n) = 1$$

krok rekurencyjny:

$$b_q(i-1) = \sum_{p \in Q} a(q,p) * e(p, x_i) * b_p(i)$$

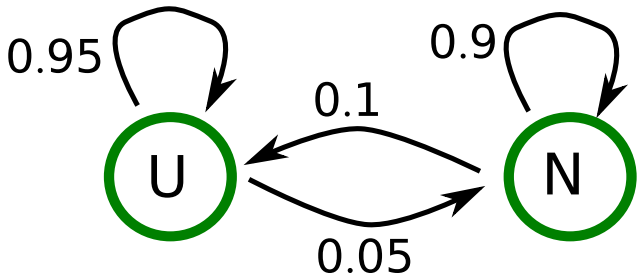
prawd. sekwencji:

$$P(S) = \sum_{q \in Q} P_q * e(q, x_1) * b_q(1)$$



nieuczciwe kasyno (Przykład HMM)

- uczciwa kostka
- nieuczciwa kostka
  $P(1) = P(2) = P(3) = P(4) = P(5) = \frac{1}{10}, P(6) = \frac{1}{2}$

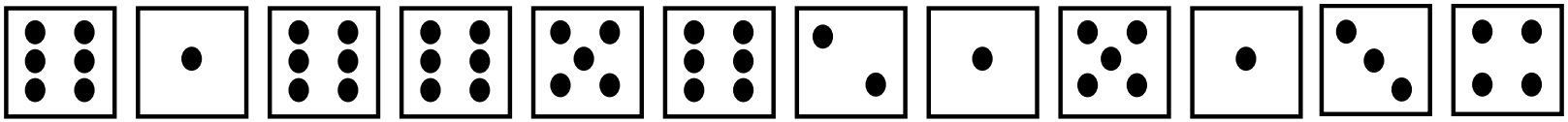


$$
\begin{aligned}
Q &= \{U, N\} \\
P_U &= 0.5, P_N = 0.5 \\
V &= \{1, 2, 3, 4, 5, 6\}
\end{aligned}
$$

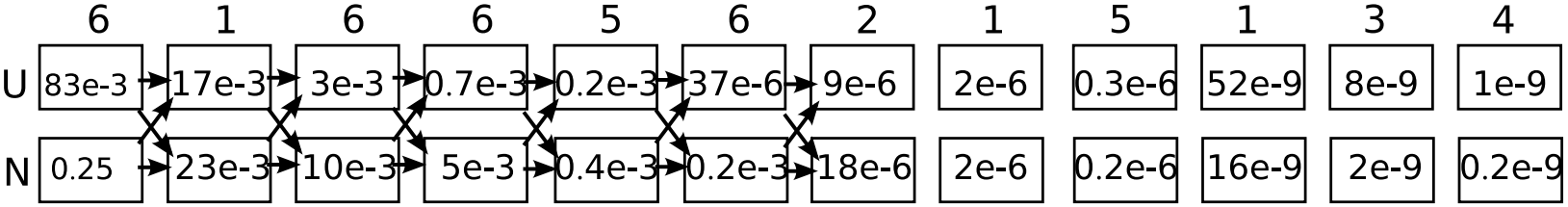
	U	N
U	0.95	0.05
N	0.1	0.9

E	U	N
1	0.167	0.1
2	0.167	0.1
3	0.167	0.1
4	0.167	0.1
5	0.167	0.1
6	0.167	0.5

Algorytm prefiksowy i sufiksowy : przykład



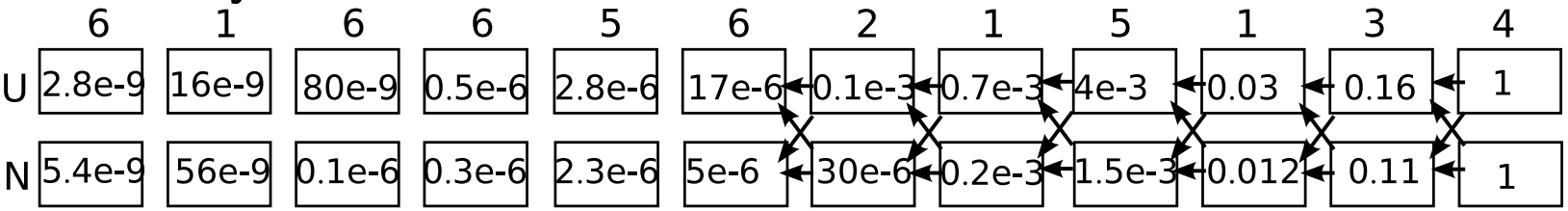
Prefiksowy:



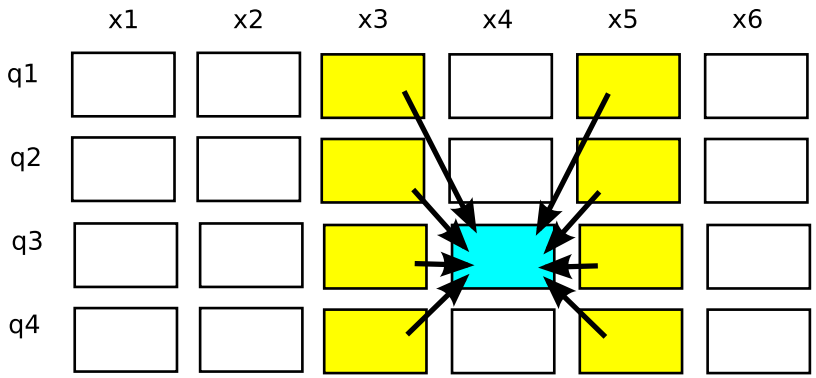
$$P(S) \simeq 1.57 * 10^{-9}$$

$$P(S|NNNNNNUUUUUU) \simeq 0.3 * 10^{-9}$$

Sufiksowy:

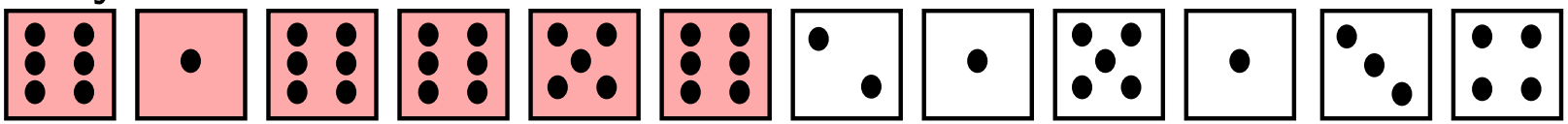


Dekodowanie za pomocą alg. prefiksowego i sufiksowego



$$P(q_i = k|x) = \frac{f_k(i) * b_k(i)}{P(S)}$$

Przykład:



$$P(q_7 = U|S) = \frac{f_U(7) * b_U(7)}{P(S)} \simeq 0.65$$

$$P(q_7 = N|S) = \frac{f_N(7) * b_N(7)}{P(S)} \simeq 0.35$$

Algorytm prefiksowy - zadanie

Posługujemy się monetami A i B, obserwując sekwencje rzutów (orły i reszki). Zakładając, że to doświadczenie jest opisywane ukrytym modelem Markowa przedstawionym niżej, podaj prawdopodobieństwo wyrzucenia sekwencji OO. Następnie podaj prawdopodobieństwo wyrzucenia sekwencji OR oraz prawdopodobieństwo wyrzucenia sekwencji RO.

$Q = \{A, B\}$
$V = \{O, R\}$
$P_A = \frac{1}{2}, P_B = \frac{1}{2}$

	A	**B**
A	$\frac{4}{5}$	$\frac{1}{5}$
B	$\frac{1}{5}$	$\frac{4}{5}$

	O	**R**
A	1	0
B	$\frac{1}{2}$	$\frac{1}{2}$

Algorytm prefiksowy - zadanie

Posługujemy się monetami A i B, obserwując sekwencje rzutów (orły i reszki). Zakładając, że to doświadczenie jest opisywane ukrytym modelem Markowa przedstawionym niżej, podaj prawdopodobieństwo wyrzucenia sekwencji OO. Następnie podaj prawdopodobieństwo wyrzucenia sekwencji OR oraz prawdopodobieństwo wyrzucenia sekwencji RO.

$Q = \{A, B\}$
$V = \{O, R\}$
$P_A = \frac{1}{2}, P_B = \frac{1}{2}$

	A	**B**
A	$\frac{4}{5}$	$\frac{1}{5}$
B	$\frac{1}{5}$	$\frac{4}{5}$

	O	**R**
A	1	0
B	$\frac{1}{2}$	$\frac{1}{2}$

$P_{OO} = 0.6,\ P_{OR} = 0.15,\ P_{RO} = 0.15,\ P_{RR} = 0.1$

Ukryty model Markowa

- Q skończony zbiór stanów
- V skończony zbiór obserwacji
- P prawdopodobieństwa stanów początkowych
- A macierz przejść $|Q| \times |Q|$
- E macierz emisji $|Q| \times |V|$

- dana sekwencja stanów, sekwencja obserwacji oraz model
  - obl. prawdopodobieństwa
  - porównywanie dla różnych sekwencji stanów
- dana sekwencja obserwacji oraz model
  - najbardziej prawdopodobna sekwencja stanów (algorytm Viterbiego)
  - prawdopodobieństwo obserwacji (algorytm prefiksowy lub sufiksowy)
- dana sekwencja obserwacji
  - obliczanie najbardziej prawdopodobnej macierzy przejść
  - obliczanie najbardziej prawdopodobnej macierzy emisji

Estymacja parametrów ukrytego modelu Markowa

Dobór macierzy przejść i macierzy emisji w oparciu o zbiór sekwencji (zbiór uczący lub zbiór trenujący)

- wejście:
  - zbiór stanów **Q**
  - zbiór obserwacji **V**
  - zbiór uczący $S_1, ..., S_N$
- wyjście
  - macierz przejść
  - macierz emisji

Estymacja parametrów dla znanej sekwencji stanów

Gdy znamy sekwencje stanów dla słów ze zbioru trenującego $S_1, ..., S_n$ oraz sekwencje obserwacji

tzn. wiemy, że słowo $S = x_1 x_2 ... x_n$ zostało wygenerowane przez sekwencję stanów $q_1 q_2 ... q_n$

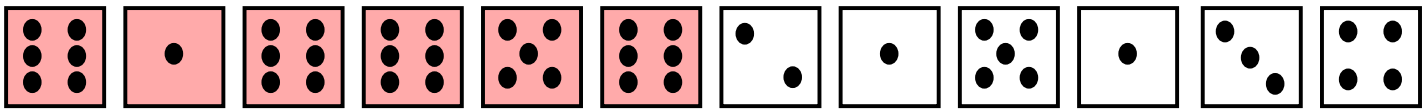
$$
\begin{aligned}
A(p,q) &= \text{liczba przejść ze stanu } p \text{ do stanu } q \\
E(p,x) &= \text{liczba emisji symbolu } x \text{ gdy układ był w stanie } p \\
a(p,q) &= \frac{A(p,q)}{\sum_{r \in Q} A(p,r)} \\
e(p,x) &= \frac{E(p,x)}{\sum_{v \in V} E(p,v)}
\end{aligned}
$$

Estymacja parametrów: przykład

Unikanie zerowych wartości : m-szacowanie, *Laplace smoothing*, *add-one smoothing*

$$a(p,q) = \frac{A(p,q)+1}{|Q|+\sum_{r\in Q} A(p,r)}$$

$$e(p,x) = \frac{E(p,x)+1}{|V|+\sum_{v\in V} E(p,v)}$$



Macierz przejść:

	U	N
U	1.0	0.0
N	0.17	0.83

Macierz przejść (m-szacowanie):

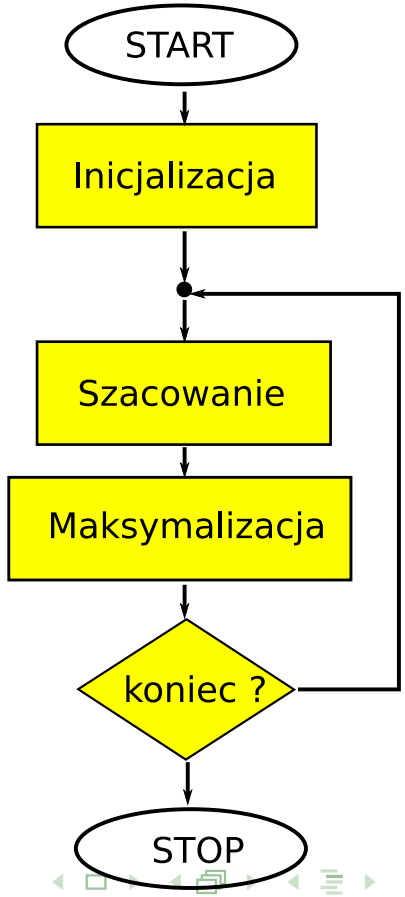
	U	N
U	0.86	0.14
N	0.25	0.75

Nieznana sekwencja stanów

1. szacowanie wartości $A(p, q)$ i $E(p, x)$
2. obliczane parametry modelu $a(p, q)$ i $e(p, x)$ (tak jak poprzednio)
3. powtarzaj krok 1 uwzględniając parametry z kroku 2

Algorytm EM:

- cykliczne powtarzanie:
  - przewidywania parametrów (krok E)
  - maksymalizacja funkcji celu (krok M)
- kryterium stopu: brak zmian w kolejnych cyklach
- optymalizacja lokalna
- szybka zbieżność



Algorytm uczenia Viterbiego

- we: zbiór $\mathbf{Q}$, obserwacji $\mathbf{V}$, zbiór uczący $S_1, ..., S_N$
- Inicjacja (równe prawdopodobieństwa):
  $a^0(p,q) = \frac{1}{|Q|}, e^0(p,x) = \frac{1}{|V|}$
- Iteracja (krok i algorytmu EM):
  - sekwencja stanów $Q_1^i, ..., Q_n^i$ (algorytm Viterbiego), używając a^i oraz e^i
  - oblicza $A^{i+1}(p,q)$ oraz $E^{i+1}(p,x)$
  - oblicza $a^{i+1}(p,q)$ oraz $e^{i+1}(p,x)$ zgodnie z zależnościami

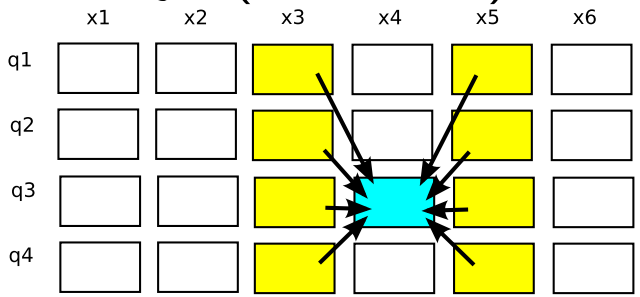
$$a(p,q) = \frac{A(p,q)}{\sum_{r \in Q} A(p,r)}$$

$$e(p,x) = \frac{E(p,x)}{\sum_{v \in V} E(p,v)}$$

- Stop:
  - liczba iteracji
  - zmiana w kolejnych cyklach mniejsza niż ϵ

Algorytm Bauma-Welcha

- we: zbiór **Q**, obserwacji **V**, sekwencja ucząca S
- Inicjacja (tak jak poprzednio): $a^0(p,q) = \frac{1}{|Q|}, e^0(p,x) = \frac{1}{|V|}$
- Iteracja (krok EM):



$$a'(p,q) = \sum_{i:x_i=p,x_{i+1}=q} \frac{f_p(i) * a(p,q) * e(q,x_{i+1}) * b_q(i+1)}{P(S)}$$

$$e'(p,x) = \sum_{i:x_i=x} \frac{f_p(i) * b_p(i)}{P(S)}$$

Podsumowanie

Obliczanie parametrów ukrytego modelu Markowa

- gdy znane sekwencje stanów dla zbioru uczącego
  - zliczanie odpowiednio liczby przejść ($A(p, q)$) oraz emitowanych symboli ($E(p, x)$) dla danego stanu
  - wykorzystanie tej informacji do obliczenia parametrów modelu (macierz przejść, macierz emisji)
- gdy nie są znane sekwencje stanów dla zbioru uczącego
  - wykorzystanie najbardziej prawdopodobnej sekwencji stanów (alg. uczenia Viterbiego)
  - wykorzystuje informacje o wszystkich sekwencjach stanów (alg. Bauma-Welcha)

Dziękuję