Statystyczna analiza danych

ukryte modele Markowa, algorytm EM

Anna Gambin

Instytut Informatyki

Uniwersytet Warszawski

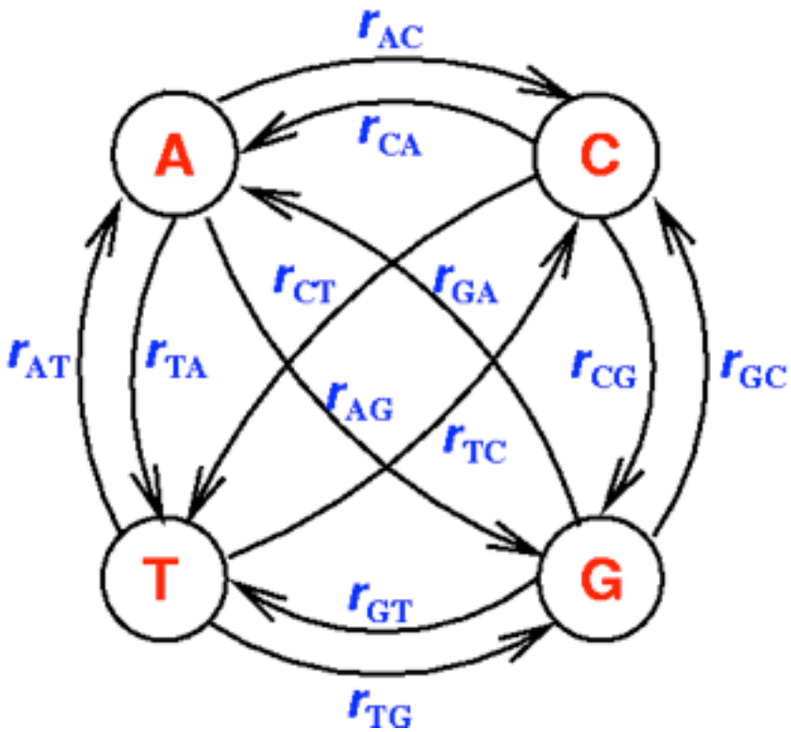
łańcuch Markowa

$Q \neq \emptyset$, zbiór stanów

$M = (p_{k,l})_{k,l \in Q}$, stochastyczna

macierz przejść

$$\sum_{l \in Q} p_{k,l} = 1.$$



ukryte modele Markowa

Niech Σ będzie alfabetem.

$$k \in Q, \xrightarrow{e_k(x)} \boxed{x \in \Sigma}$$

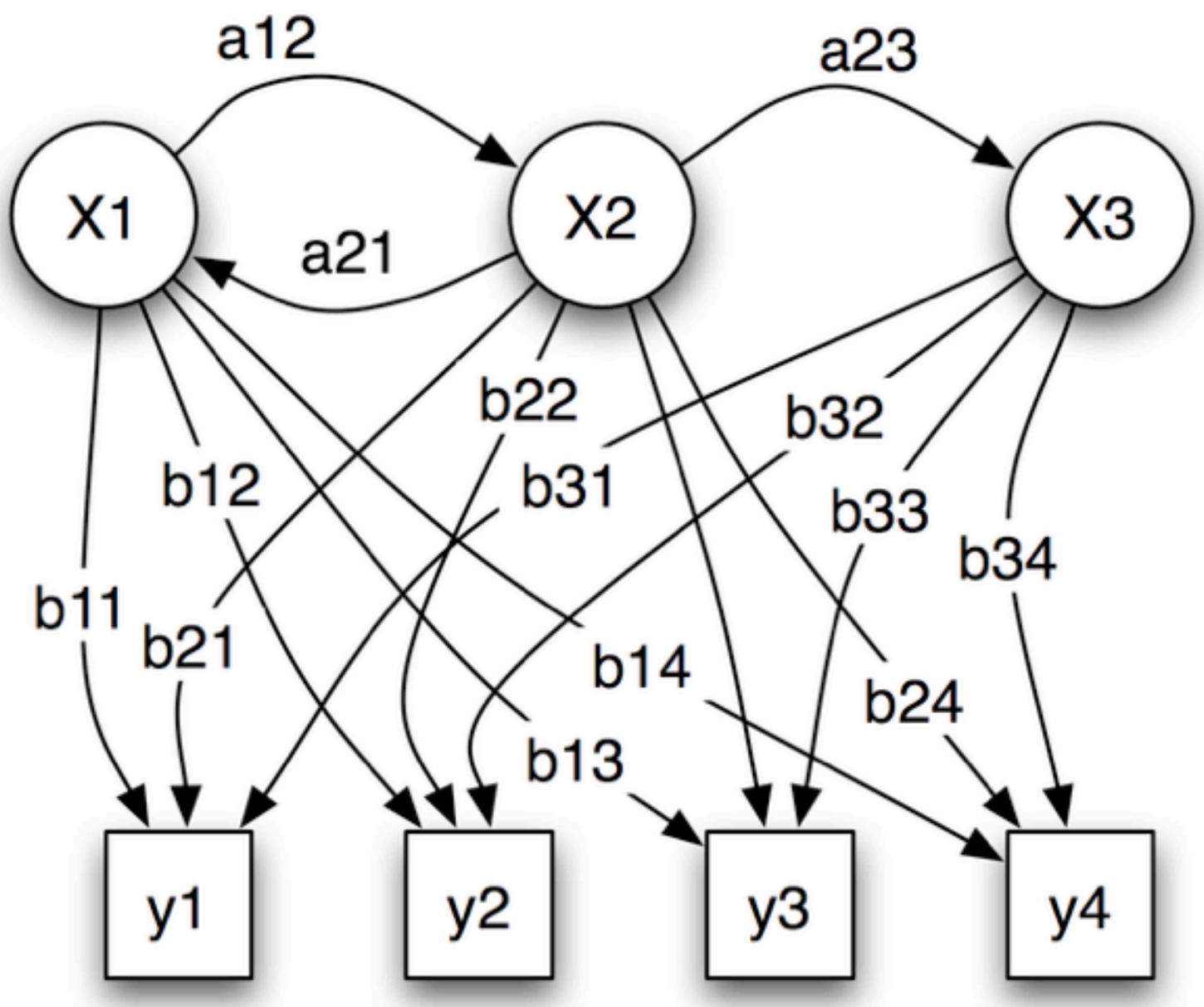
probabilistyczne automaty z wyjściem

stan ukryty obserwacja

$$\sum_{x \in \Sigma} e_k(x) = 1$$

w każdym stanie emitowany jest jakiś symbol

z wikipedii :)



x — states
y — possible observations
a — state transition probabilities
b — output probabilities

przykład: nieuczciwe kasyno

F - uczciwa kostka

L - fałszywa kostka - szóstka wypada z prawdopodobieństwem = 0.5, pozostałe z pr = 0.1

Możemy, na przykład, mieć do czynienia z następującym modelem: $p_{F,F} = 0.95$, $p_{L,L} = 0.9$, $p_{F,L} = 0.05$, $p_{L,F} = 0.1$; ponadto prawdopodobieństwo emisji jest zdefiniowane następująco: $e_F(x) = 1/6$ dla $x \in \{1, 2, 3, 4, 5, 6\}$ oraz $e_L(6) = 0.5$ i $e_L(x) = 0.1$ dla $x \in \{1, 2, 3, 4, 5\}$ $\square$

przykład: wyspy CpG

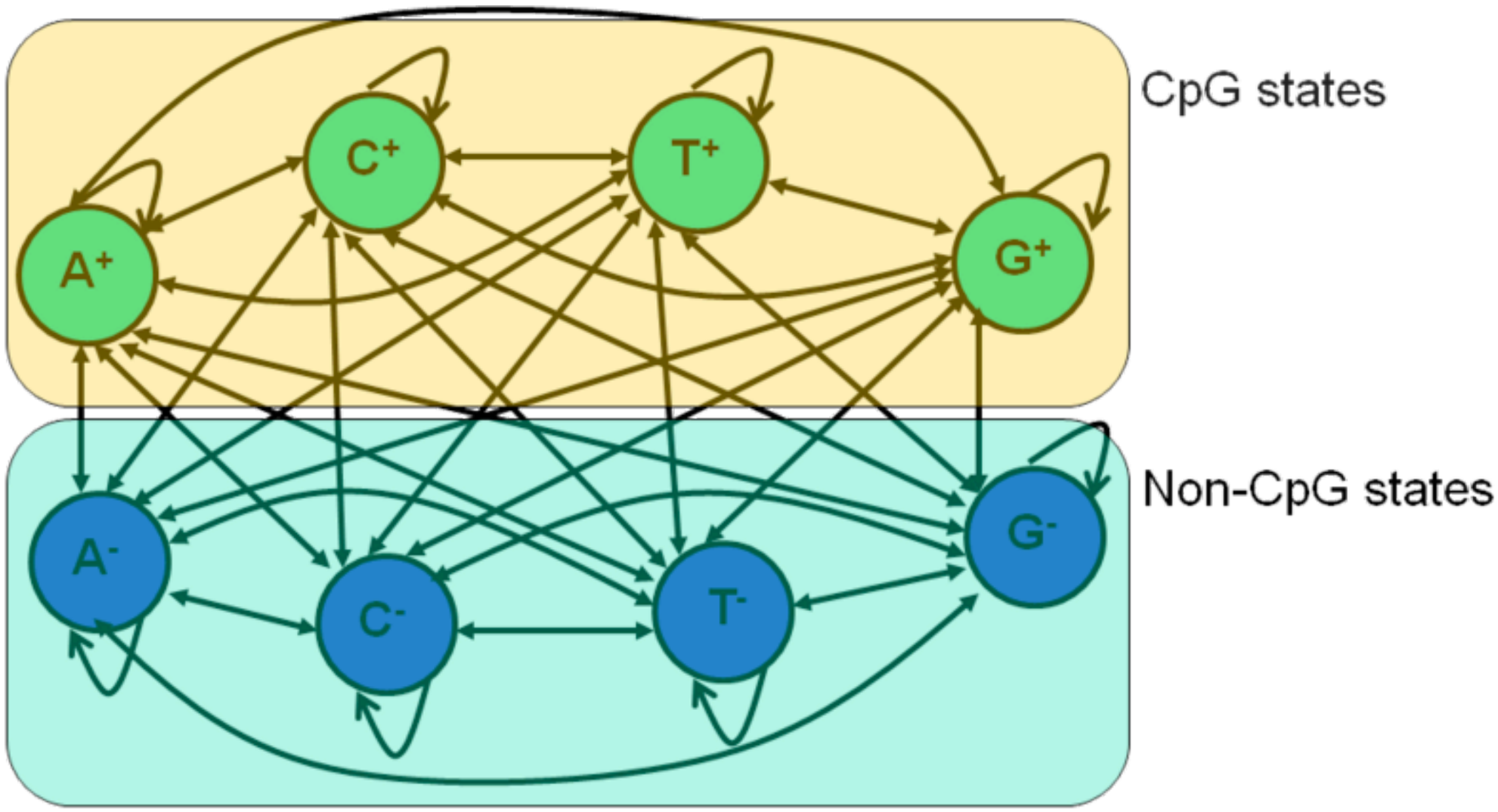
5’ —— C G —— 3’ CpG

z dużym prawdopodobieństwem C-> T (metylacja)

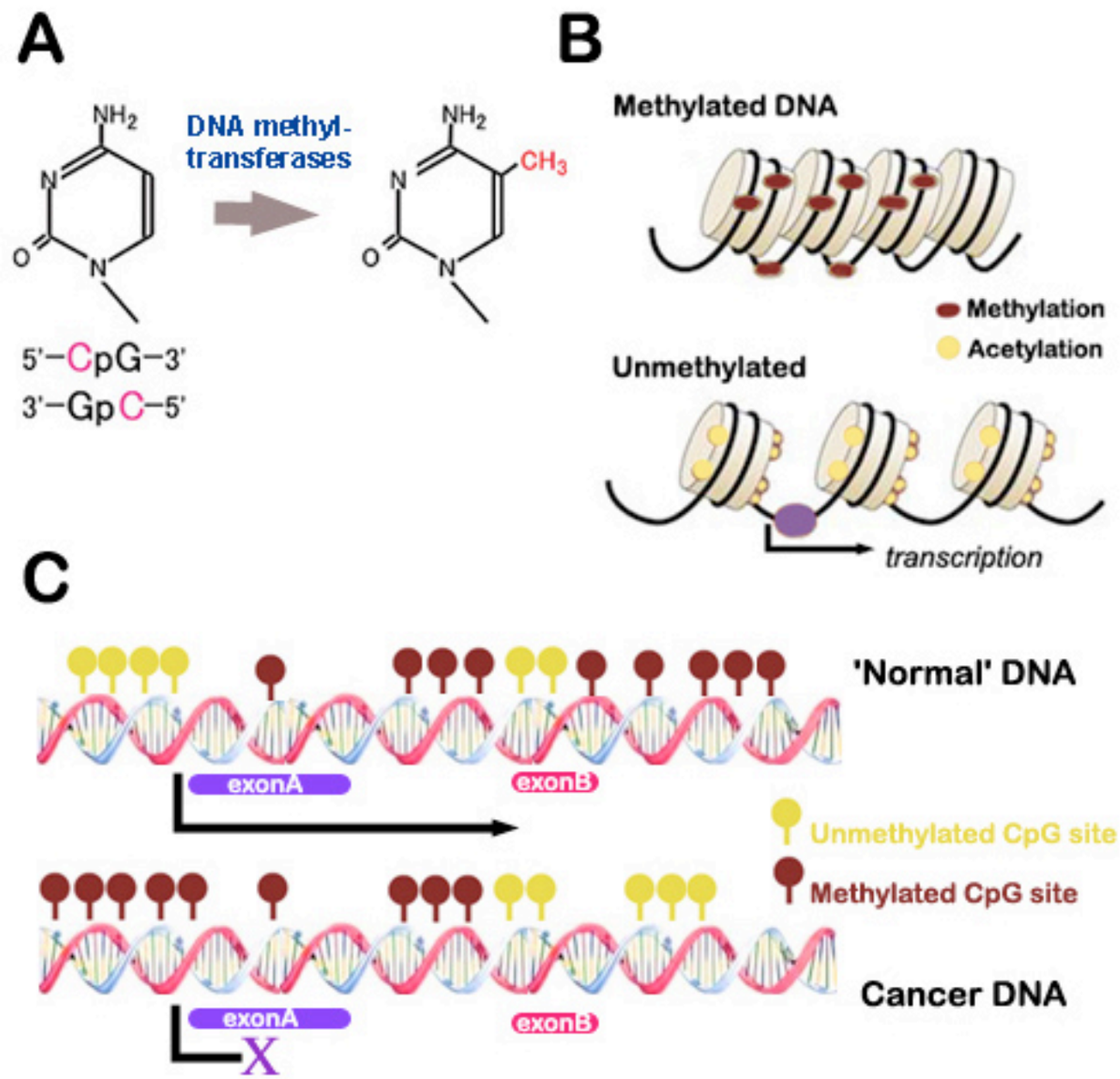
proces ten jest zahamowany w regionach promotorowych i w pobliżu miejsc rozpoczynających rejony kodujące

$A_+, C_+, G_+, T_+, A_-, C_-, G_-, T_-$ stany ukryte

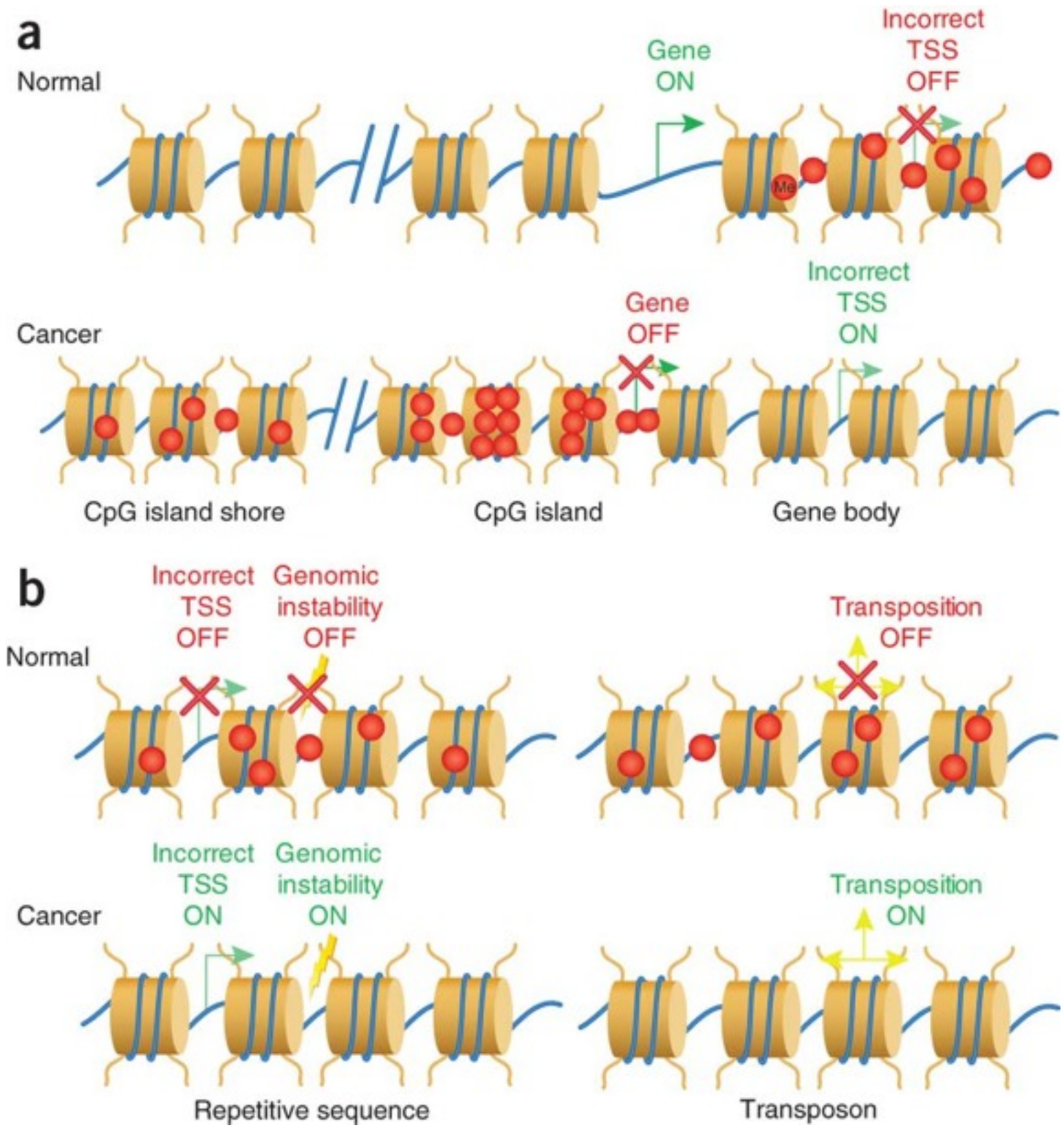
w stanie A_ξ (gdzie $\xi \in \{-, +\}$), emitujemy A


CpG states
Non-CpG states
A+
C+
T+
G+
A-
C-
T-
G-

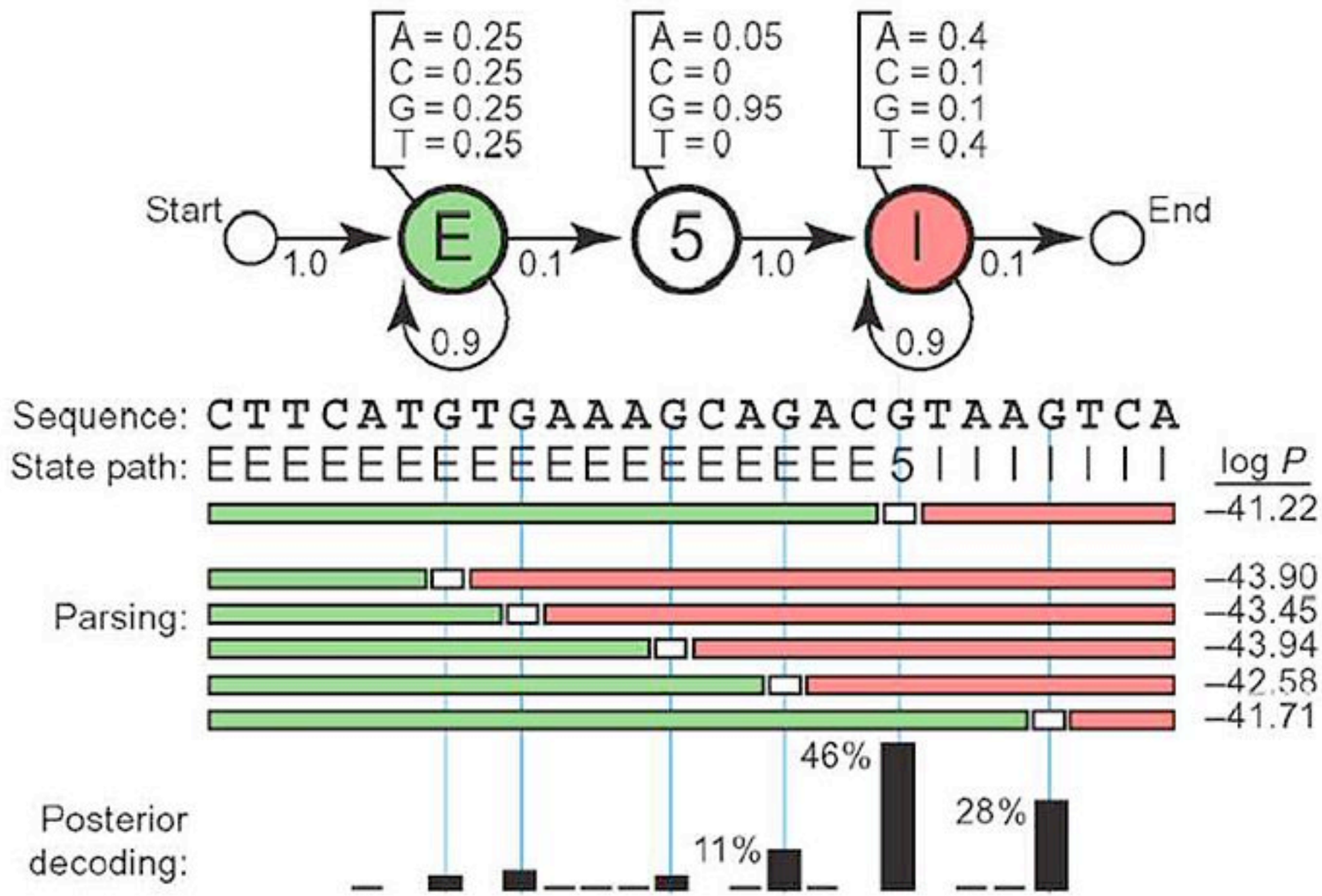
przykład: wyspy CpG



http://www.cellscience.com/


a
Normal
Gene
ON
Incorrect
TSS
OFF
Me
Cancer
Gene
OFF
Incorrect
TSS
ON
CpG island shore
CpG island
Gene body
b
Normal
Incorrect
TSS
OFF
Genomic
instability
OFF
Transposition
OFF
Cancer
Incorrect
TSS
ON
Genomic
instability
ON
Transposition
ON
Repetitive sequence
Transposon

przykład: introny/eksony

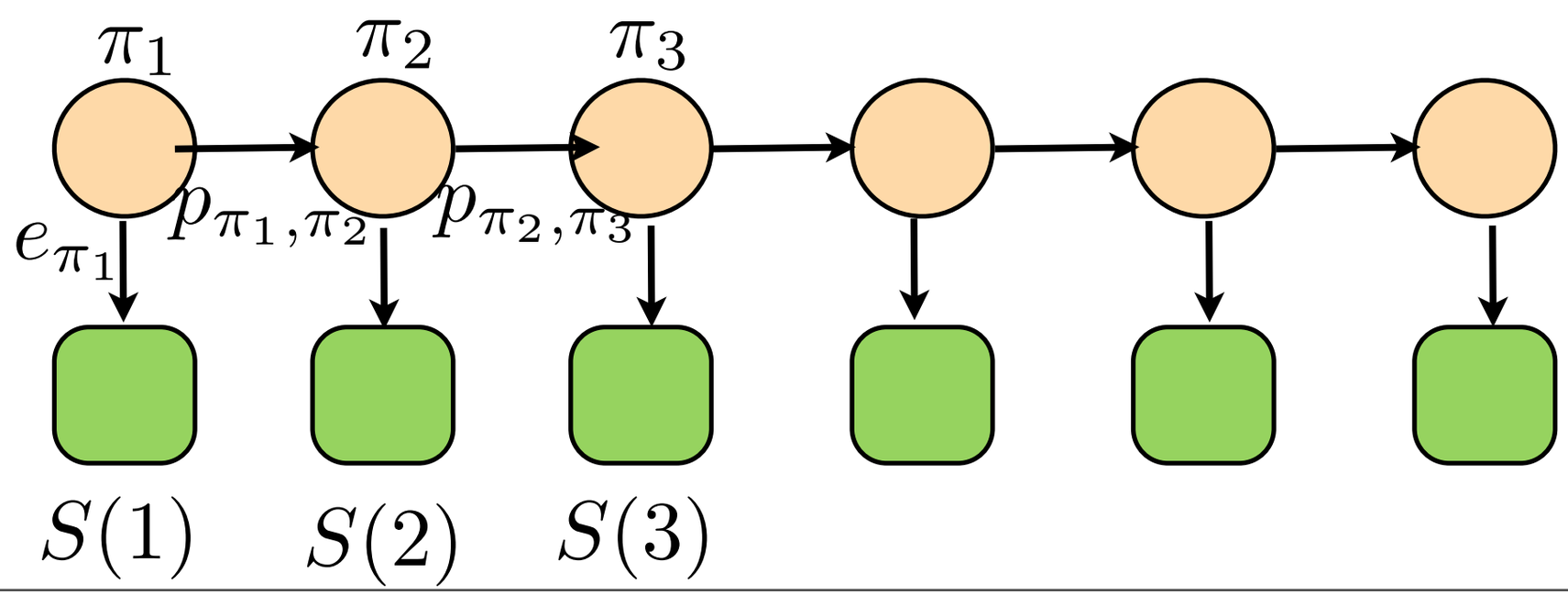


ukryte modele Markowa c.d.

Niech $S \in \Sigma^*$ oraz $\pi \in Q^*$ będą niepustymi ciągami równej długości $n = |S| = |\pi| > 0$. Prawdopodobieństwo tego że S zostanie wyemitowane oraz układ będzie zmieniał stany według kolejności π wynosi

$$P(S, \pi) = \prod_{t=0}^{n-1} e_{\pi(t+1)}(S(t+1)) \cdot p_{\pi(t),\pi(t+1)},$$

gdzie w powyższym wzorze przyjmujemy, że $\pi(0) = k_0$, jest stanem początkowym.



algorytm Viterbiego

$S \in \Sigma^*$ słowo zaobserwowane na wyjściu

poszukiwany π_*

ciąg stanów, który doprowadził do wyemitowania S

$$P(S, \pi_*) = max\{P(S, \pi) \mid \pi \in Q^*, |\pi| = |S|\}.$$

algorytm Viterbiego

$0 < i \leq |S|$

$$v(i,k) = max\{P(S[1..i],\pi) \mid \pi \in Q^i, \quad \pi(i) = k\}.$$

prawdopodobieństwo najlepszej drogi kończącej się w k

$$v(0,k) = \begin{cases} 1 & \text{gdy } k = k_0, \\ 0 & \text{gdy } k \neq k_0. \end{cases}$$

$$v(i,k) = e_k(S(i)) \cdot \max_{l \in Q}[v(i-1,l) \cdot p_{l,k}]$$

algorytm Viterbiego

maksymalne prawdopodobieństwo znajdujemy ze wzoru:

$$P(S, \pi_*) = \max_{k \in Q} [v(|S|, k)]$$

optymalną ścieżkę odtwarzamy z zapamiętanych wskaźników

algorytm Viterbiego-złożoność

czas $O(|S| \cdot |Q|^2)$

pamięć $O(|S| \cdot |Q|)$

UWAGA problem: mnożenie małych liczb

przykład: nieuczciwe kasyno

```
Rolls   3151162464466442453113216311641521336251445436316566265666 66
Die     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLL
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLL

Rolls   6511664531326512456366646316366631623264552362666666251516 31
Die     LLLLLLFFFFFFFFFFFLLLLLLLLLLLLLLLFFLLLLLLLLLLLLLFFFFFFFFF
Viterbi LLLLLLFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLFFFFFFFF

Rolls   2225554416665665635643243641315134651463534111264146262533 56
Die     FFFFFFFLLLLLLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLL
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL

Rolls   3661636664662325344136616611632525624622552652522664353533 36
Die     LLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Viterbi LLLLLLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Rolls   2331216253644144323351632436336655624666626326666123552452 42
Die     FFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLFFFFFFFFFFF
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLFFFFFFFFFF
```

prawdopodobieństwo emisji słowa

$$P(S) = \sum_{\pi} P(S, \pi)$$

$$P(S, \pi) = 0 \quad \text{dla} \quad |S| \neq |\pi|.$$

UWAGA: liczba ścieżek wykładnicza !

można aproksymować za pomocą π_*

ale nie trzeba ...

algorytm prefiksowy

$0 < i \leq |S| \qquad k \in Q$

$$f(i,k) = \sum_{\{\pi \mid \pi(i)=k\}} P(S[1..i], \pi).$$

prawdopodobieństwo wyemitowania prefiksu długości i droga kończy sie w stanie k

algorytm prefiksowy

$$f(0,k) = \begin{cases} 1 & \text{gdy } k = k_0, \\ 0 & \text{gdy } k \neq k_0. \end{cases}$$

Dla $0 < i \leq |S|$ *oraz* $k \in Q$ *mamy*

$$f(i,k) = e_k(S(i)) \cdot \sum_{l \in Q} f(i-1,l) \cdot p_{l,k}$$

Prawdopodobieństwo wyemitowania słowa S

$$\boxed{P(S) = \sum_{k \in Q} f(|S|,k)}$$

algorytm sufiksowy

$$P(S) = b(0, k_0) \qquad 0 \leq i \leq |S| = m$$

$$Dla\ 0 \leq i < |S|\ oraz\ k \in Q\ mamy$$

$$b(i,k) = \sum_{l \in Q} p_{k,l} \cdot e_l(S(i+1)) \cdot b(i+1, l).$$

prawdopodobieństwo wyemitowania sufiksu i..m
i-ty symbol emitowany w stanie k

algorytm prefiksowy sufiksowy - złożoność

czas $O(|S| \cdot |Q|^2)$

pamięć $O(|S| \cdot |Q|)$

i-ty krok w stanie k
pod warunkiem,
że nastąpiła emisja słowa S

$$P(\pi(i) = k \mid S) = \frac{P(\pi(i) = k \ \& \ S)}{P(S)} = \frac{f(i,k) \cdot b(i,k)}{P(S)}$$

prawdopodobieństwo **a posteriori**

estymacja parametrów HMM

$S_1, \ldots, S_n,$ zaobserwowany zbiór słów uczących

Σ alfabet

Q zbiór stanów

znamy

$p_{k,l}$ oraz $e_k(x)$

szukamy

estymacja parametrów HMM

słowa są emitowane niezależnie

$$P_{\mathcal{M}}(S_1 \ \& \ \ldots \ \& \ S_n) = \prod_{j=1}^{n} P_{\mathcal{M}}(S_j)$$

prawdopodobieństwo wygenerowania słów

$S_1 \ldots S_n$

w modelu $\mathcal{M}$

estymacja parametrów HMM

$$\log(P_{\mathcal{M}}(S)) \longrightarrow Score_{\mathcal{M}}(S)$$

szukamy $\mathcal{M}$

który maksymalizuje $\displaystyle\sum_{j=1}^{n} Score_{\mathcal{M}}(S_j)$

estymacja parametrów HMM uproszczenie

dla słowa S_j znamy ciąg stanów π_j

Niech $P_{k,l}$ będzie równe liczbie przejść ze stanu k w stan l w ciągach $\pi_1, \ldots, \pi_n$.

$E_k(x)$ będzie równe liczbie emisji symbolu x w stanie k.

$$p_{k,l} = \frac{P_{k,l}}{\sum_{q \in Q} P_{k,q}}, \qquad e_k(x) = \frac{E_k(x)}{\sum_{y \in \Sigma} E_k(y)}.$$

zadanie: estymacja parametrów HMM

$f^{(j)}_{\mathcal{M}}(i,k)$

$b^{(j)}_{\mathcal{M}}(i,k)$

$$f(i,k) = \sum_{\{\pi \mid \pi(i)=k\}} P(S[1..i],\pi).$$

$$b(i,k) = \sum_{l \in Q} p_{k,l} \cdot e_l(S(i+1)) \cdot b(i+1,l)$$

$S_1, \ldots, S_n \in \Sigma^*$

Q

HMM $\mathcal{M}$

maksymalizujący

$\sum_{j=1}^{n} Score_{\mathcal{M}}(S_j).$

algorytm Bauma Welcha

Krok 1: (*Inicjalizacja*) model $\mathcal{M}$ Σ oraz Q

Krok 2: Obliczmy wartość oczekiwaną liczby przejść w modelu $\mathcal{M}$ ze stanu k do stanu l.

$$\frac{f_{\mathcal{M}}^{(j)}(i,k) \cdot p_{k,l}^{\mathcal{M}} \cdot e_l^{\mathcal{M}}(S_j(i+1)) \cdot b_{\mathcal{M}}^{(j)}(i+1,l)}{P_{\mathcal{M}}(S_j)}$$

prawdopodobieństwo: i-ty krok emisji S_j w stanie k $i+1$ w stanie l

algorytm Bauma Welcha

wartość oczekiwana liczby przejść ze stanu k w l

$$P_{k,l} = \sum_{j=1}^{n} \sum_{i=1}^{|S_j|} \frac{f_{\mathcal{M}}^{(j)}(i,k) \cdot p_{k,l}^{\mathcal{M}} \cdot e_l^{\mathcal{M}}(S_j(i+1)) \cdot b_{\mathcal{M}}^{(j)}(i+1,l)}{P_{\mathcal{M}}(S_j)}.$$

wartość oczekiwaną liczby emisji symbolu x w stanie k

$$E_k(x) = \sum_{j=1}^{n} \sum_{i \in I_j(x)} \frac{f_{\mathcal{M}}^{(j)}(i,k) \cdot b_{\mathcal{M}}^{(j)}(i,k)}{P_{\mathcal{M}}(S_j)}$$

gdzie

$$I_j(x) = \{i \mid S_j(i) = x\}$$

algorytm Bauma Welcha

Krok 3:

Mając nowe wartości $P_{k,l}$ oraz $E_k(x)$ wyznaczamy

$$p_{k,l} = \frac{P_{k,l}}{\sum_{q \in Q} P_{k,q}}, \qquad e_k(x) = \frac{E_k(x)}{\sum_{y \in \Sigma} E_k(y)}.$$

otrzymujemy nowy model

$$\mathcal{M}'$$

algorytm Bauma Welcha

Krok 4: Obliczamy $\sum_{j=1}^{n} Score_{\mathcal{M}'}(S_j)$

jeśli nie różni się wiele
od poprzedniej wartości
to przerywamy

różne pkty startowe

iterujemy, aż poprawa
będzie niewielka

lokalne maksimum

algorytm Bauma Welcha

szczególny przypadek algortymu EM (Expectation- Maximization) służącego do estymacji parametrów modelu stochastycznego

wady:

- lokalne maksimum
- powolna (liniowa) zbieżność
- wrażliwość na punkt startowy

ogólniej: algorytm EM

$\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ z rozkładu $\mathcal{P}(\mathbf{X}|\theta)$

wiarygodność danych:

$$p(\mathcal{X}|\Theta) = \prod_{i=1}^{N} p(\mathbf{x}_i|\Theta) = \mathcal{L}(\Theta|\mathcal{X}).$$

estymator największej wiarygodności

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}}\ \mathcal{L}(\Theta|\mathcal{X}).$$

często łatwiej maskymalizować $L(\theta) = \ln \mathcal{P}(\mathbf{X}|\theta)$

ogólniej: algorytm EM

algorytm przydatny w dwóch przypadkach:

- zakładamy niepełne dane (np. niektóre jest ciężko zaobserwować)

- funkcja wiarygodności jest analitycznie nie do ugryzienia, ale zmienia się to jeśli uwzględnimy dodatkowe (ukryte) parametry.

ogólniej: algorytm EM

efektywna iteracyja metoda maksymalizacji logarytmu wiarygodności przy założeniu ukrytych/ niepełnych danych. W każdej iteracji dwa kroki:

E-krok: liczymy warunkową wartość oczekiwaną wiarygodności, pod warunkiem obserwowanych danych oraz aktualnych oszacowań parametrów.

M-krok: szukamy parametrów, które maksymalizują funkcję wiarygodności.
zbieżność zagwarantowana, bo wiarygodność rośnie w każdej iteracji

funkcja wypukła

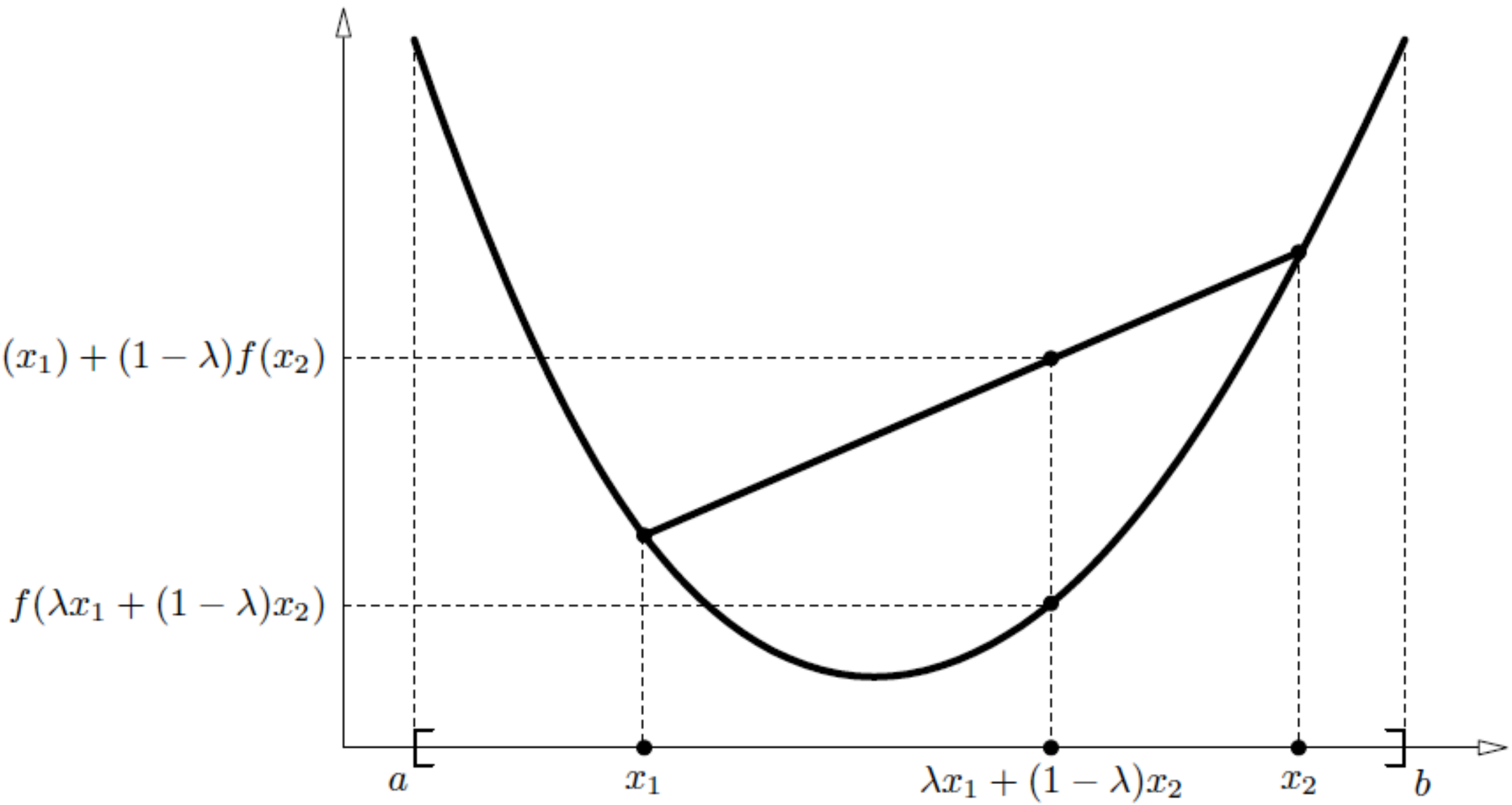


Figure 1: f is *convex* on $[a, b]$ if $f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$ $\forall x_1, x_2 \in [a, b], \quad \lambda \in [0, 1]$.

Theorem 2 (Jensen's inequality) *Let f be a convex function defined on an interval I. If $x_1, x_2, \ldots, x_n \in I$ and $\lambda_1, \lambda_2, \ldots, \lambda_n \geq 0$ with $\sum_{i=1}^{n} \lambda_i = 1$,*

$$f\left(\sum_{i=1}^{n} \lambda_i x_i\right) \leq \sum_{i=1}^{n} \lambda_i f(x_i)$$

$$\frac{1}{n}\sum_{i=1}^{n} x_i \geq \sqrt[n]{x_1 x_2 \cdots x_n}.$$

$$\ln \sum_{i=1}^{n} \lambda_i x_i \geq \sum_{i=1}^{n} \lambda_i \ln(x_i).$$

algorytm EM

maksymalizujemy: $L(\theta) = \ln \mathcal{P}(\mathbf{X}|\theta)$

w kroku *n* oszacowanie dla $\theta \longrightarrow \theta_n$

chcemy znaleźć nowe oszacowanie, takie że

$$L(\theta) > L(\theta_n)$$

czyli maksymalizujemy różnicę:

$$L(\theta) - L(\theta_n) = \ln \mathcal{P}(\mathbf{X}|\theta) - \ln \mathcal{P}(\mathbf{X}|\theta_n)$$

algorytm EM

ukryte dane - zmienna losowa $\longrightarrow$ $\mathbf{Z}$

rozkład łączny: $$\mathcal{P}(\mathbf{X}|\theta) = \sum_{\mathbf{z}} \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)$$

$$L(\theta) - L(\theta_n) =$$
$$= \ln\left(\sum_{\mathbf{z}} \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n)$$

The Expectation Maximization Algorithm
A short tutorial

Sean Borman

Theorem 2 (Jensen's inequality) *Let f be a convex function defined on an interval I. If $x_1, x_2, \ldots, x_n \in I$ and $\lambda_1, \lambda_2, \ldots, \lambda_n \geq 0$ with $\sum_{i=1}^{n} \lambda_i = 1$,*

$$f\left(\sum_{i=1}^{n} \lambda_i x_i\right) \leq \sum_{i=1}^{n} \lambda_i f(x_i)$$

$$\ln \sum_{i=1}^{n} \lambda_i x_i \geq \sum_{i=1}^{n} \lambda_i \ln(x_i)$$

$$L(\theta) - L(\theta_n) =$$
$$= \ln\left(\sum_{\mathbf{z}} \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n)$$

λ_i	$\mathcal{P}(\mathbf{z}\|\mathbf{X},\theta_n)$
$\lambda_i \geq 0$	$\mathcal{P}(\mathbf{z}\|\mathbf{X},\theta_n) \geq 0$
$\sum_{i=1}^{n} \lambda_i = 1$	$\sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}\|\mathbf{X},\theta_n) = 1$

algorytm EM

$$
\begin{aligned}
L(\theta) - L(\theta_n) &= \ln\left(\sum_{\mathbf{z}} \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n) \\
&= \ln\left(\sum_{\mathbf{z}} \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta) \cdot \frac{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)}\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n) \\
&= \ln\left(\sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)}\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n) \\
&\geq \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln\left(\frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)}\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n) \\
\sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) = 1 \quad &= \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln\left(\frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)\mathcal{P}(\mathbf{X}|\theta_n)}\right) \\
&\triangleq \Delta(\theta|\theta_n).
\end{aligned}
$$

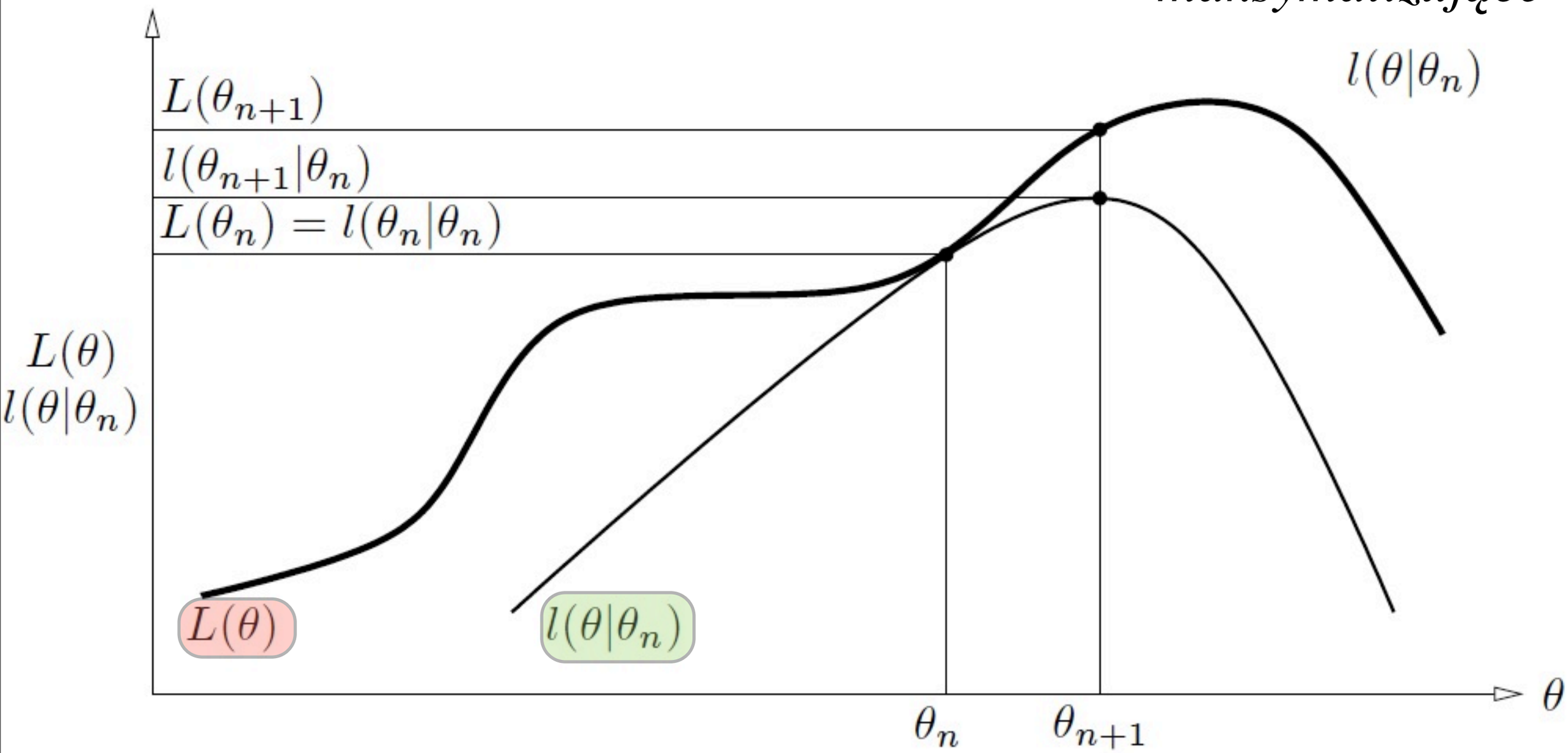
Johan Jensen

$$L(\theta) \geq L(\theta_n) + \Delta(\theta|\theta_n)$$

$$l(\theta|\theta_n) \triangleq L(\theta_n) + \Delta(\theta|\theta_n)$$

$$L(\theta) \geq l(\theta|\theta_n)$$

EM wybiera θ_{n+1} *maksymalizujące*



algorytm EM

$$
\begin{aligned}
l(\theta_n|\theta_n) &= L(\theta_n) + \Delta(\theta_n|\theta_n) \\
&= L(\theta_n) + \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta_n)\mathcal{P}(\mathbf{z}|\theta_n)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)\mathcal{P}(\mathbf{X}|\theta_n)} \\
&= L(\theta_n) + \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \frac{\mathcal{P}(\mathbf{X},\mathbf{z}|\theta_n)}{\mathcal{P}(\mathbf{X},\mathbf{z}|\theta_n)} \\
&= L(\theta_n) + \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln 1 \\
&= L(\theta_n),
\end{aligned}
$$

$$\theta = \theta_n \longrightarrow l(\theta|\theta_n) = L(\theta)$$

algorytm EM

EM wybiera θ_{n+1}

maksymalizujące $l(\theta|\theta_n)$

$$\begin{aligned}\theta_{n+1} &= \arg\max_{\theta} \{l(\theta|\theta_n)\} \\ &= \arg\max_{\theta} \left\{ L(\theta_n) + \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{X}|\theta_n)\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)} \right\}\end{aligned}$$

wyrzucamy czynniki niezależne od θ

algorytm EM

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$$\begin{aligned}
\theta_{n+1} &= \arg\max_{\theta} \left\{ \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X}, \theta_n) \ln \mathcal{P}(\mathbf{X}|\mathbf{z}, \theta) \mathcal{P}(\mathbf{z}|\theta) \right\} \\
&= \arg\max_{\theta} \left\{ \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X}, \theta_n) \ln \frac{\mathcal{P}(\mathbf{X}, \mathbf{z}, \theta)}{\mathcal{P}(\mathbf{z}, \theta)} \frac{\mathcal{P}(\mathbf{z}, \theta)}{\mathcal{P}(\theta)} \right\} \\
&= \arg\max_{\theta} \left\{ \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X}, \theta_n) \ln \mathcal{P}(\mathbf{X}, \mathbf{z}|\theta) \right\} \\
&= \arg\max_{\theta} \left\{ \mathrm{E}_{\mathbf{Z}|\mathbf{X}, \theta_n} \left\{ \ln \mathcal{P}(\mathbf{X}, \mathbf{z}|\theta) \right\} \right\}
\end{aligned}$$

warunkowa wartość oczekiwana

$$E[h(Y)|X = x] = \int_y h(y) f_{Y|X}(y|x) dy$$

E- krok kontemplacja

- czasami rozkład brzegowy jest prostym wyrażeniem zależnym od danych i paramaterów z poprzedniego kroku...

- czasami jest trudno...
- często używamy gęstości w postaci:

$$f(\mathbf{y}, \mathcal{X}|\Theta^{(i-1)}) = f(\mathbf{y}|\mathcal{X}, \Theta^{(i-1)}) f(\mathcal{X}|\Theta^{(i-1)})$$

nie zależy ona wtedy od wcześniejszych kroków...

algorytm EM

E-krok: $\mathrm{E}_{\mathbf{Z}|\mathbf{X},\theta_n}\{\ln \mathcal{P}(\mathbf{X},\mathbf{z}|\theta)\}$

M-krok: $\arg\max_{\theta}\left\{\mathrm{E}_{\mathbf{Z}|\mathbf{X},\theta_n}\left\{\ln \mathcal{P}(\mathbf{X},\mathbf{z}|\theta)\right\}\right\}$

uogólniony algorytm EM

θ_{n+1} *wybieraliśmy, żeby maksymalizowało* $\Delta(\theta|\theta_n)$

A wystarczy, żeby:

$$\Delta(\theta_{n+1}|\theta_n) \geq \Delta(\theta_n|\theta_n)$$

Generalized Expectation Maximization (GEM)

algorytm EM: zbieżność

$$\Delta(\theta_{n+1}|\theta_n) \geq \Delta(\theta_n|\theta_n) = 0$$

czyli wiarygodność $L(\theta)$ *w każdej iteracji nie maleje*

dodatkowo jeśli θ_n *maksymalizuje* $l(\theta|\theta_n)$

oraz $l(\theta_n|\theta_n) = L(\theta_n)$

to θ_n *jest punktem krytycznym również dla* $L(\theta)$