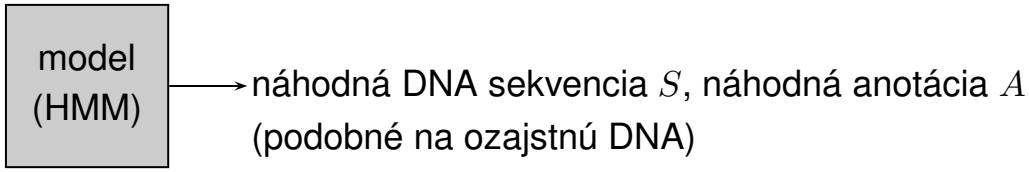
Algoritmy pre HMM

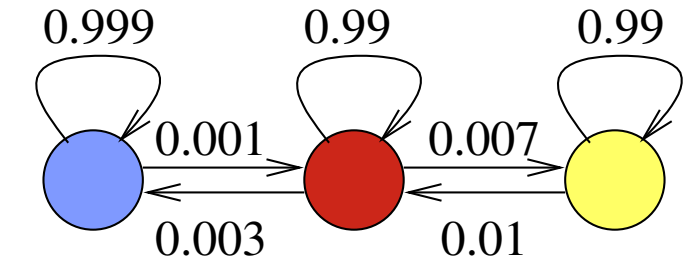
Broňa Brejová

26.10.2023

Opakovanie: HMM (skrytý Markovov model)



$\Pr(S, A)$ – pravdepodobnosť, že model vygeneruje pár (S, A).



Predpokladajme, že model vždy začína v modrom stave.

$$\Pr(\text{acag}) = 0.27 \cdot 0.001 \cdot 0.27 \cdot 0.99 \cdot 0.24 \cdot 0.99 \cdot 0.28 = 4.8 \cdot 10^{-6}$$

$$\Pr(\text{acag}) = 0.27 \cdot 0.999 \cdot 0.23 \cdot 0.999 \cdot 0.27 \cdot 0.999 \cdot 0.23 = 0.0038$$

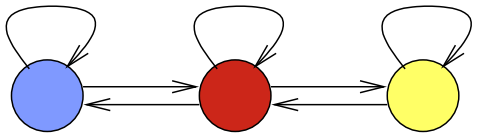
Iný hračkársky príklad: počasie

- Obdobie nízkeho tlaku vzduchu: väčšinou prší
- Obdobie nízkeho tlaku vzduchu: väčšinou slnečno

Každé obdobie trvá typicky nieľkoko dní

Cvičenie: reprezentuj ako HMM

Parametre HMM (označenie)



Sekvencia $S = S_1, \ldots, S_n$

Anotácia $A = A_1, \ldots, A_n$

Parametre modelu:

Prechodová pravdepodobnosť $a(u, v) = \Pr(A_{i+1} = v | A_i = u)$,

Emisná pravdepodobnosť $e(u, x) = \Pr(S_i = x | A_i = u)$,

Počiatočná pravdepodobnosť $\pi(u) = \Pr(A_1 = u)$.

a			
	0.99	0.007	0.003
	0.01	0.99	0
	0.001	0	0.999

e	a	c	g	t
	0.24	0.27	0.28	0.21
	0.26	0.22	0.22	0.30
	0.27	0.23	0.23	0.27

Výsledná pravdepodobnosť:

$\Pr(A, S) = \pi(A_1)e(A_1, S_1) \prod_{i=2}^{n} a(A_{i-1}, A_i)e(A_i, S_i)$

Viterbiho algoritmus

Pre dané HMM a sekvenciu S

nájdi najpravdepodobnejšiu anotáciu (postupnosť stavov)

$A = \arg\max_A \Pr(A, S) = \arg\max_A \Pr(A|S)$

Ako by ste to riešili?

Pripomeňme si príklad:

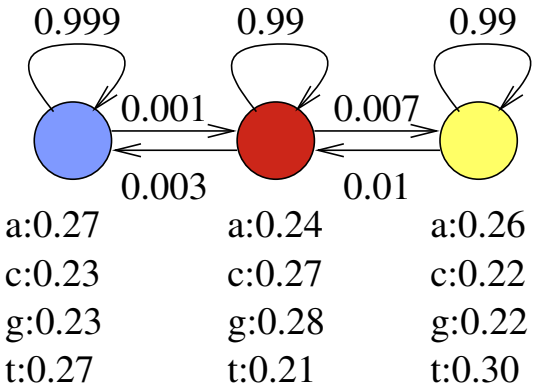
$$\Pr(\text{acag}) = 0.27 \cdot 0.001 \cdot 0.27 \cdot 0.99 \cdot 0.24 \cdot 0.99 \cdot 0.28 = 4.8 \cdot 10^{-6}$$

$$\Pr(\text{acag}) = 0.27 \cdot 0.999 \cdot 0.23 \cdot 0.999 \cdot 0.27 \cdot 0.999 \cdot 0.23 = 0.0038$$

Viterbiho algoritmus

Nájdi najpravdepodobnejšiu postupnosť stavov $A = \arg\max_A \Pr(A, S)$

Podproblém $V[u, i]$: pravdepodobnosť najpravdepodobnejšej cesty končiacej po i krokoch v stave u, pričom vygeneruje $S_1 S_2 \ldots S_i$



$V[u,i]$	a	c	a	g

Viterbiho algoritmus

Podproblém $V[u, i]$: pravdepodobnosť najpravdepodobnejšej cesty končiacej po i krokoch v stave u, pričom vygeneruje $S_1 S_2 \ldots S_i$

Rekurencia?

$V[u, 1] =$

$V[u, i] =$

Pripomeňme si označenie:

Sekvencia $S = S_1, \ldots, S_n$, anotácia (stavy) $A = A_1, \ldots, A_n$

Prechodová pravdepodobnosť $a(u, v) = \Pr(A_{i+1} = v | A_i = u)$,

Emisná pravdepodobnosť $e(u, x) = \Pr(S_i = x | A_i = u)$,

Počiatočná pravdepodobnosť $\pi(u) = \Pr(A_1 = u)$.

$\Pr(A, S) = \pi(A_1) e(A_1, S_1) \prod_{i=2}^{n} a(A_{i-1}, A_i) e(A_i, S_i)$

Viterbiho algoritmus

Podproblém $V[u, i]$: pravdepodobnosť najpravdepodobnejšej cesty končiacej po i krokoch v stave u, pričom vygeneruje $S_1 S_2 \ldots S_i$

Rekurencia:

$V[u, 1] = \pi(u) \cdot e(u, S_1)$

$V[u, i] = \max_w V[w, i-1] \cdot a(w, u) \cdot e(u, S_i)$

Algoritmus, celková odpoveď, čas výpočtu?

Pripomeňme si označenie:

Sekvencia $S = S_1, \ldots, S_n$, anotácia (stavy) $A = A_1, \ldots, A_n$

Prechodová pravdepodobnosť $a(u, v) = \Pr(A_{i+1} = v | A_i = u)$,

Emisná pravdepodobnosť $e(u, x) = \Pr(S_i = x | A_i = u)$,

Počiatočná pravdepodobnosť $\pi(u) = \Pr(A_1 = u)$.

$\Pr(A, S) = \pi(A_1) e(A_1, S_1) \prod_{i=2}^{n} a(A_{i-1}, A_i) e(A_i, S_i)$

Viterbiho algoritmus (zhrnutie)

Nájdi najpravdepodobnejšiu postupnosť stavov $A = \arg\max_A \Pr(A, S)$

Podproblém $V[u, i]$: pravdepodobnosť najpravdepodobnejšej cesty končiacej po i krokoch v stave u, pričom vygeneruje $S_1 S_2 \ldots S_i$

Rekurencia:

$V[u, 1] = \pi(u) \cdot e(u, S_1)$

$V[u, i] = \max_w V[w, i-1] \cdot a(w, u) \cdot e(u, S_i)$

Algoritmus:

Inicializuj $V[*, 1]$

for $i = 2 \ldots n$ (n=dĺžka S)

  for $u = 1 \ldots m$ (m =počet stavov)

    vypočítaj $V[u, i]$, ulož najlepšie w do $B[u, i]$

Maximálne $V[u, n]$ cez všetky u je $\max_A \Pr(A, S)$

Cestu nájdi odzadu pomocou matice B

Dynamické programovanie v čase $O(nm^2)$

Ďalší problém: celková pravdepodobnosť S

Viterbi počíta $\arg\max_A \Pr(A, S)$

Teraz chceme celkovú pravdepodobnosť, že vygenerujeme sekvenciu S

t.j. $\Pr(S) = \sum_A \Pr(A, S)$

Užitočné napr. na porovnávanie rôznych modelov,

ktorý má väčšiu šancu vygenerovať S

Ako by ste to počítali?

Pripomeňme si príklad:

$$\Pr(\text{acag}) = 0.27 \cdot 0.001 \cdot 0.27 \cdot 0.99 \cdot 0.24 \cdot 0.99 \cdot 0.28 = 4.8 \cdot 10^{-6}$$

$$\Pr(\text{acag}) = 0.27 \cdot 0.999 \cdot 0.23 \cdot 0.999 \cdot 0.27 \cdot 0.999 \cdot 0.23 = 0.0038$$

Dopredný algoritmus (forward algorithm)

Počíta celkovú pravdepodobnosť, že vygenerujeme sekvenciu S,
$\Pr(S) = \sum_A Pr(A, S)$

Podproblém $F[u, i]$: pravdepodobnosť, že po i krokoch vygenerujeme $S_1, S_2, \ldots S_i$ a dostaneme sa do stavu u.

$F[u, i] = \Pr(A_i = u \wedge S_1, S_2, \ldots, S_i) =$
$\sum_{A_1, A_2, \ldots, A_i = u} \Pr(A_1, A_2, ..., A_i \wedge S_1, S_2, ..., S_i)$

Rekurencia?

$F[u, 1] =$
$F[u, i] =$

Pripomeňme si rekurenciu z Viterbiho:

$V[u, 1] = \pi(u) \cdot e(u, S_1)$
$V[u, i] = \max_w V[w, i-1] \cdot a(w, u) \cdot e(u, S_i)$

Dopredný algoritmus (forward algorithm)

Počíta celkovú pravdepodobnosť, že vygenerujeme sekvenciu S,
$\Pr(S) = \sum_A Pr(A, S)$

Podproblém $F[u, i]$: pravdepodobnosť, že po i krokoch vygenerujeme $S_1, S_2, \ldots S_i$ a dostaneme sa do stavu u.

Rekurencia

$F[u, 1] = \pi(u) \cdot e(u, S_1)$
$F[u, i] = \sum_w F[w, i-1] \cdot a(w, u) \cdot e(u, S_i)$

Pripomeňme si rekurenciu z Viterbiho:

$V[u, 1] = \pi(u) \cdot e(u, S_1)$
$V[u, i] = \max_w V[w, i-1] \cdot a(w, u) \cdot e(u, S_i)$

Dopredný algoritmus (forward algorithm)

Počíta celkovú pravdepodobnosť, že vygenerujeme sekvenciu S,
$\Pr(S) = \sum_A Pr(A, S)$

Podproblém $F[u, i]$: pravdepodobnosť, že po i krokoch vygenerujeme $S_1, S_2, \ldots S_i$ a dostaneme sa do stavu u.

Rekurencia

$F[u, 1] = \pi(u) \cdot e(u, S_1)$
$F[u, i] = \sum_w F[w, i-1] \cdot a(w, u) \cdot e(u, S_i)$

Výsledok?

Celková pravdepodobnosť $\Pr(S) =$

Čas výpočtu?

Dopredný algoritmus (forward algorithm)

Počíta celkovú pravdepodobnosť, že vygenerujeme sekvenciu S,
$\Pr(S) = \sum_A Pr(A, S)$

Podproblém $F[u, i]$: pravdepodobnosť, že po i krokoch vygenerujeme $S_1, S_2, \ldots S_i$ a dostaneme sa do stavu u.

$F[u, i] = \Pr(A_i = u \wedge S_1, S_2, \ldots, S_i) =$
$\sum_{A_1, A_2, \ldots, A_i = u} \Pr(A_1, A_2, ..., A_i \wedge S_1, S_2, ..., S_i)$

Výsledok

Celková pravdepodobnosť $\Pr(S) = \sum_u F[u, n]$

Čas výpočtu $O(nm^2)$

Tretí problem: pravdepodobnosť, že S_i bolo generované v stave u

$$\Pr(A_i = u \mid S) = \frac{\Pr(A_i = u, S)}{\Pr(S)}$$

$$\Pr(A_i = u, S) = \sum_{A : A_i = u} \Pr(A, S)$$

Vypočítame kombináciou dopredného a spätného algoritmu

$F[u, i]$: pravdepodobnosť, že po i krokoch vygenerujeme $S_1, S_2, \ldots S_i$ a dostaneme sa do stavu u.

$B[u, i]$: pravdepodobnosť, že ak začneme v u na pozícii i, tak vygenerujeme $S_{i+1} \ldots, S_n$ v najbližších krokoch

$$\Pr(A_i = u, S) = F[u, i] \cdot B[u, i]$$

Spätný algoritmus (backward algorithm)

Dopredný algoritmus: pravdepodobnosť, že po i krokoch vygenerujeme $S_1, S_2, \ldots S_i$ a dostaneme sa do stavu u.

$$F[u, 1] = \pi(u) \cdot e(u, S_1)$$
$$F[u, i] = \sum_w F[w, i-1] \cdot a(w, u) \cdot e(u, S_i)$$

Spätný algoritmus: $B[u, i]$: pravdepodobnosť, že ak začneme v u na pozícii i, tak vygenerujeme $S_{i+1} \ldots, S_n$ v najbližších krokoch

Ako spočítať $B[u, i]$?

Spätný algoritmus (backward algorithm)

Dopredný algoritmus: pravdepodobnosť, že po i krokoch vygenerujeme $S_1, S_2, \ldots S_i$ a dostaneme sa do stavu u.

$$F[u, 1] = \pi(u) \cdot e(u, S_1)$$
$$F[u, i] = \sum_w F[w, i-1] \cdot a(w, u) \cdot e(u, S_i)$$

Spätný algoritmus: $B[u, i]$: pravdepodobnosť, že ak začneme v u na pozícii i, tak vygenerujeme $S_{i+1} \ldots, S_n$ v najbližších krokoch

$$B[u, n] = 1$$
$$B[u, i] = \sum_w B[w, i+1] \cdot a(u, w) \cdot e(w, S_{i+1})$$

Cvičenie: Ako spočítať $\Pr(S)$ pomocou matice B?

Aposteriórne dekódovanie (posterior decoding)

Videli sme: $\Pr(A_i = u \mid S) = \frac{F[u,i] \cdot B[u,i]}{\Pr(S)}$

Aposteriórne pravdepodobnosti stavov:

Použitím dopredného a spätného alg. vieme teda spočítať $\Pr(A_i = u \mid S)$ pre všetky u a i v celkovom čase $O(nm^2)$

Aposteriórne dekódovanie

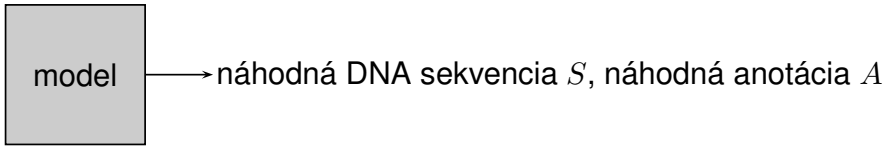
Pre dané S zvolíme A také že $A_i = \max_i \Pr(A_i = u \mid S)$

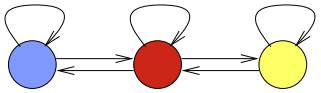
Výhoda: Berie do úvahy suboptimálne postupnosti stavov

Nevýhoda: $\Pr(A \mid S)$ môže byť 0 alebo veľmi nízka

Iná možnosť: zvolíme A Viterbiho algoritmom, aposteriórne pravdepodobnosti použijeme na priradenie dôveryhodnosti jendotlivým častiam A

Hľadanie génov s HMM



- **Určenie stavov a prechodov v modeli:** ručne, na základe poznatkov o štruktúre génu. 
- **Trénovanie parametrov:** pravdepodobnosti určíme na základe sekvencií so známymi génmi (**trénovacia množina**).
  Model zostavíme tak, aby páry (S, A) s vlastnosťami podobnými skutočným génom mali veľkú pravdepodobnosť $\Pr(S, A)$
- **Použitie:** pre novú sekvenciu S nájdi najpravdepodobnejšiu anotáciu $A = \arg\max_A \Pr(A|S)$ Viterbiho algoritmom v $O(nm^2)$

Trénovanie HMM

- Stavový priestor + povolené prechody väčšinou ručne
- Parametre (pravdepodobnosti prechodu, emisie a počiatočné) automaticky z trénovacích sekvencií
- Čím zložitejší model a viac parametrov máme, tým potrebujeme viac trénovacích dát, aby nedošlo k **preučeniu**, t.j. k situácii, keď model dobre zodpovedá nejakým zvláštnostiam trénovacích dát, nie však ďalším dátam.
- Presnosť modelu testujeme na zvláštnych testovacích dátach, ktoré sme nepoužili na trénovanie.

Trénovanie HMM z anotovaných sekvencií

Vstup: topológia modelu a niekoľko trénovacích párov $(S^{(1)}, A^{(1)}), (S^{(2)}, A^{(2)}), \ldots$

Cieľ: nastaviť $\pi(u)$, $e(u, x)$, $a(u, v)$ tak, aby $\prod_i \Pr(S^{(i)}, A^{(i)})$ bola čo najväčšia

Dosiahneme jednoduchým počítaním frekvencií

Napr. $a(u, v)$: nájdeme všetky výskyty stavu u a zistíme, ako často za nimi ide stav v

Trénovanie HMM z neanotovaných sekvencií

Vstup: topológia modelu a niekoľko trénovacích sekvencií $S^{(i)}$
anotácie $A^{(i)}$ nepoznáme

Cieľ: nastaviť $\pi(u)$, $e(u, x)$, $a(u, v)$ tak, aby $\prod_i \Pr(S^{(i)})$ bola čo najväčšia

Používajú sa heuristické iteratívne algoritmy, napr. Baum-Welchov, ktorý je verziou všeobecnejšieho algoritmu EM (expectation maximization).

V každej iterácii používa dopradný a spätný algoritmus.

Tvorba stavového priestoru modelu

Príklad HMM na hľadanie génov