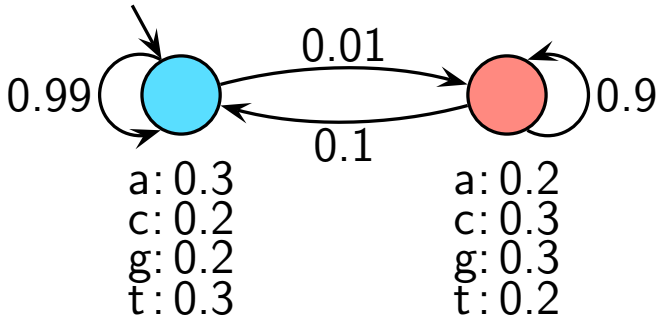
Skryté Markovove modely

Hidden Markov models (HMMs)



Reťazec: $\{a, c, g, t\}^*$

$$X = x_1, x_2, \ldots, x_n$$

Postupnosť stavov: $\{\blacksquare, \blacksquare\}^*$

$$S = s_1, s_2, \ldots, s_n$$

tatttagcgtcttctatcatccaatcactgcactttacacactataaatagagcagctca
tgggcgtatttgcgctagtgttgggtgttccgctgtgctgttttccgtcatggctcgca
ctaagcaaactgctcggaagtctactggtggcaaggcgccacgcaaacagttggccacta

HMM definuje $P(X, S)$ pre reťazce X a postupnosti stavov S:

$$\begin{aligned} P(X, S) &= P(s_1)P(x_1|s_1)P(s_1 \to s_2)P(x_2|s_2)\cdots \\ &\quad P(s_{n-1} \to s_n)P(x_n|s_n) \end{aligned}$$

Použitie HMM

- Modelovanie sekvenčných dát:

  časové rady pozorovaní, text, zvukové signály, biologické sekvencie

- Zvyčajne reťazec X známy,

  postupnosť stavov S skrytý faktor (chceme zistiť)

- Model zostavíme tak, aby realistické dvojice (X, S) mali vysokú pravdepodobnosť

Príklad použitia HMM: hľadanie génov v DNA

Vstup: DNA sekvencia $X = x_1, \ldots, x_n$

```
tatttagcgtcttctatcatccaatcactgcactttaocacactataaatagagcagctca
tgggcgtatttgcgctagtgttgggtgttccgctgtgctgttttccgtcatggctcgca
ctaagcaaactgctcggaagtctactggtggcaaggcgccacgcaaacagttggccacta
```

Výstup: poloha génov (úsekov kódujúcich proteíny)

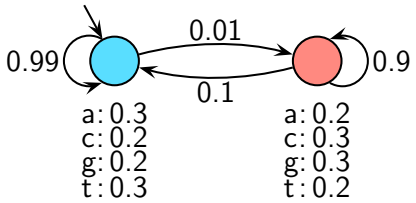
Alebo: označkuj každý znak ako kódujúci/nekódujúci

Postupnosť značiek $S = s_1, s_2, \ldots, s_n$

```
tatttagcgtcttctatcatccaatcactgcactttacacactataaatagagcagctca
tgggcgtatttgcgctagtgttgggtgttccgctgtgctgttttccgtcatggctcgca
ctaagcaaactgctcggaagtctactggtggcaaggcgccacgcaaacagttggccacta
```

Výpočtový problém

Vstup: reťazec $X = x_1, \ldots, x_n$ a HMM




```
tatttagcgtcttctatcatccaatcactgcactttacacactataaatagagcagctca
tgggcgtatttgcgctagtgttgggtgttccgctgtgctgttttccgtcatggctcgca
ctaagcaaactgctcggaagtctactggtggcaaggcgccacgcaaacagttggccacta
```

Nájdi **najpravdepodobnejšiu postupnosť stavov** $S = s_1, \ldots, s_n$

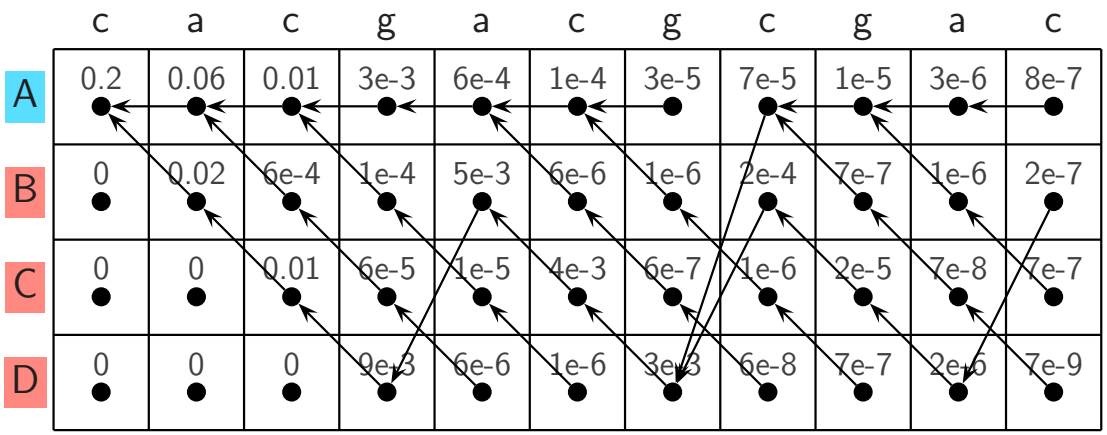
s maximálnou $P(X, S)$: **Viterbiho algoritmus**

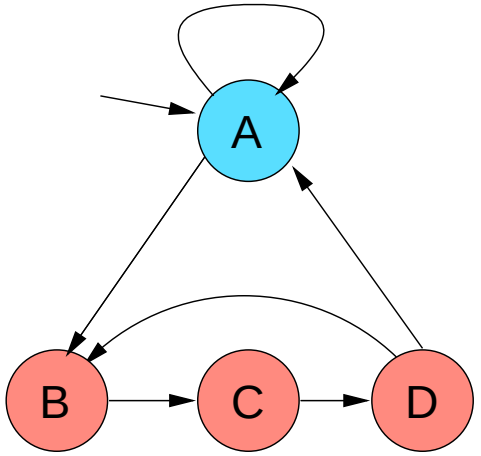
```
tatttagcgtcttctatcatccaatcactgcactttacacactataaatagagcagctca
tgggcgtatttgcgctagtgttgggtgttccgctgtgctgttttccgtcatggctcgca
ctaagcaaactgctcggaagtctactggtggcaaggcgccacgcaaacagttggccacta
```

Viterbiho algoritmus

Dynamické programovanie:

- $A[i, v]$ — pravdepodobnosť najlepšej postupnosti stavov, ktorá vygeneruje $x_1, \ldots, x_i$ a skončí v stave v
- $A[i, v] = \max_u A[i-1, u] \cdot P(u \to v) \cdot P(x_i|v)$
- Pre každé $A[i, v]$ si pamätáme **najlepší predchádzajúci stav** u

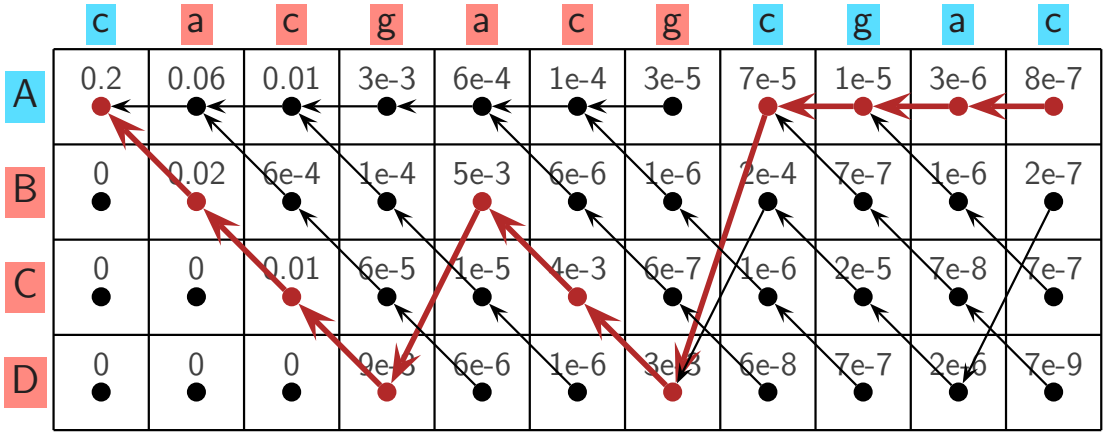


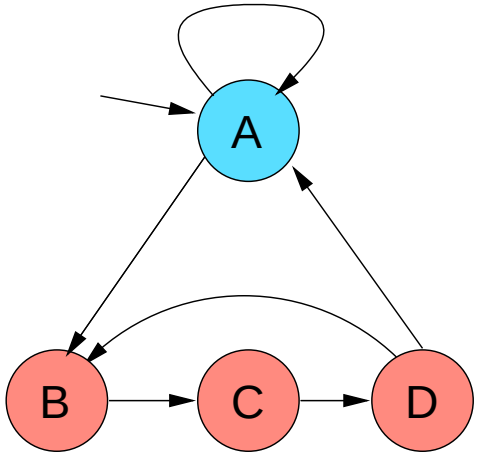


Viterbiho algoritmus

Dynamické programovanie:

- $A[i, v]$ — pravdepodobnosť najlepšej postupnosti stavov, ktorá vygeneruje $x_1, \ldots, x_i$ a skončí v stave v
- $A[i, v] = \max_u A[i-1, u] \cdot P(u \to v) \cdot P(x_i|v)$
- Pre každé $A[i, v]$ si pamätáme **najlepší predchádzajúci stav** u





Zložitosť Viterbiho algoritmu

Čas $O(nm^2)$, pamät $O(nm)$

$n =$ dĺžka sekvencie, $m =$ počet stavov modelu

Príklad: hľadanie génov na 250 MB sekvencii

so 100-stavovým HMM $\Rightarrow$ **25 GB pamäte**

Zložitosť Viterbiho algoritmu

Čas $O(nm^2)$, pamät $O(nm)$

$n =$ dĺžka sekvencie, $m =$ počet stavov modelu

Príklad: hľadanie génov na 250 MB sekvencii

so 100-stavovým HMM $\Rightarrow$ **25 GB pamäte**

Prístupy s menej pamäte

- Nasekáme X na kratšie kusy

  Problémy na hraniciach, suboptimálne riešenie celku

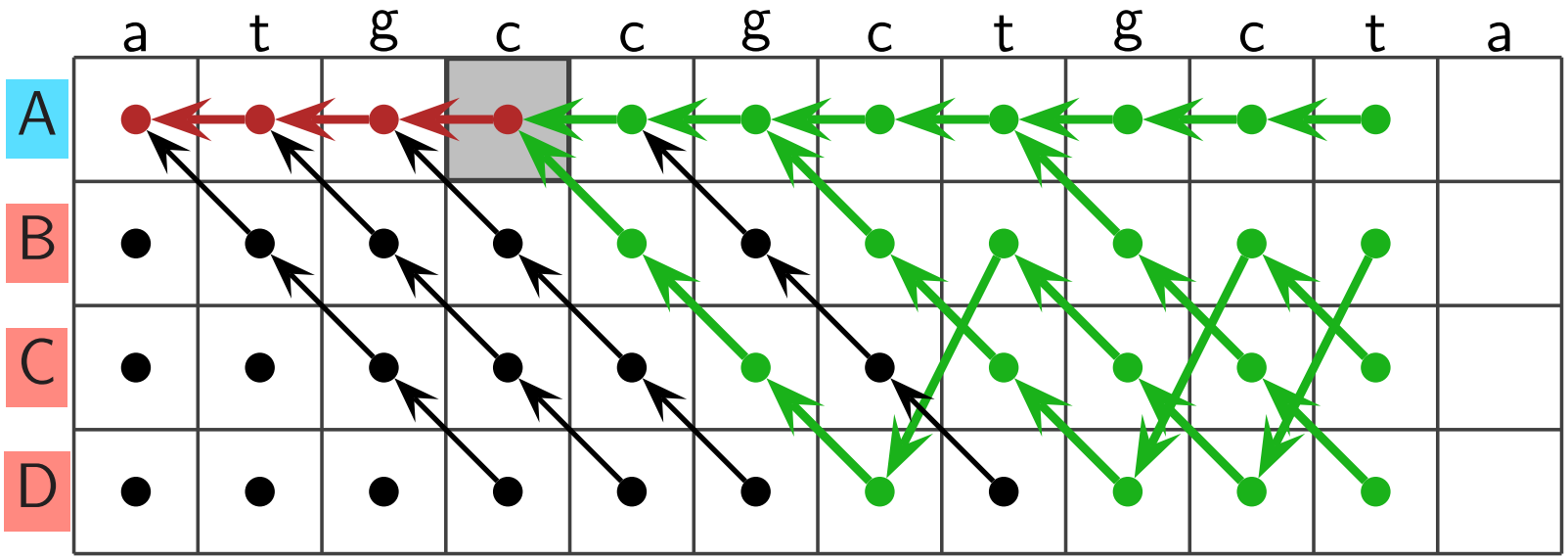
- **Check pointing** [Grice et al. 1997]

  Pamäť $O(n + m\sqrt[L]{n})$, L-násobné spomalenie

- **Náš prístup: on-line algoritmus**

  Veľkosť pamäte sa dynamicky mení, väčšinou malá

On-line Viterbiho algoritmus

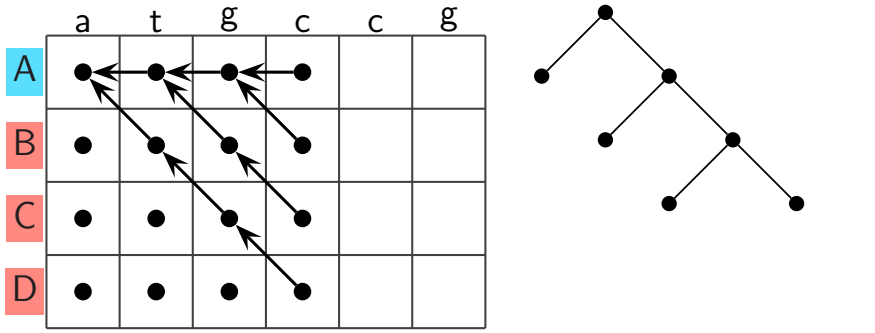


- Efektívna detekcia **sútoku**
- Vypíš cestu naľavo od sútoku
- Zmaž dáta naľavo od sútoku

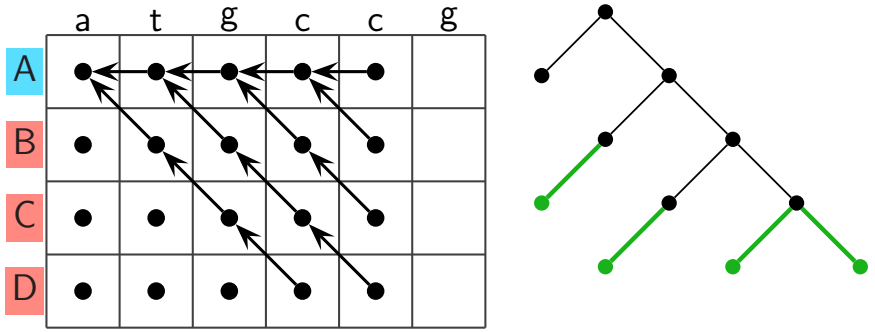
Súčasne s nami podobný algoritmus aj [Keibler, Arumugam, Brent 2007]

Efektívna detekcia sútoku

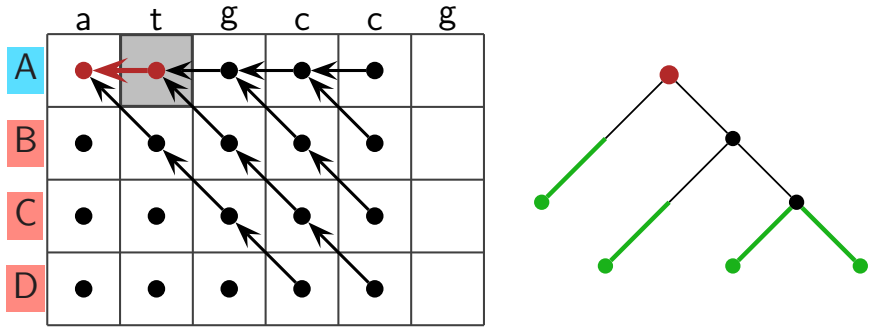
Udržuj komprimovaný strom spätných liniek



V každom kroku pridaj nové spätné linky. . .

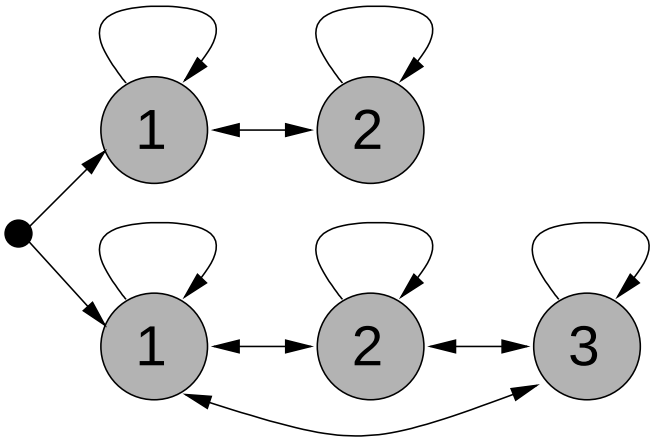


. . . vymaž nepoužité vetvy, skontrahuj cesty



Spomalenie: $\approx 5\%$

Zlé správy

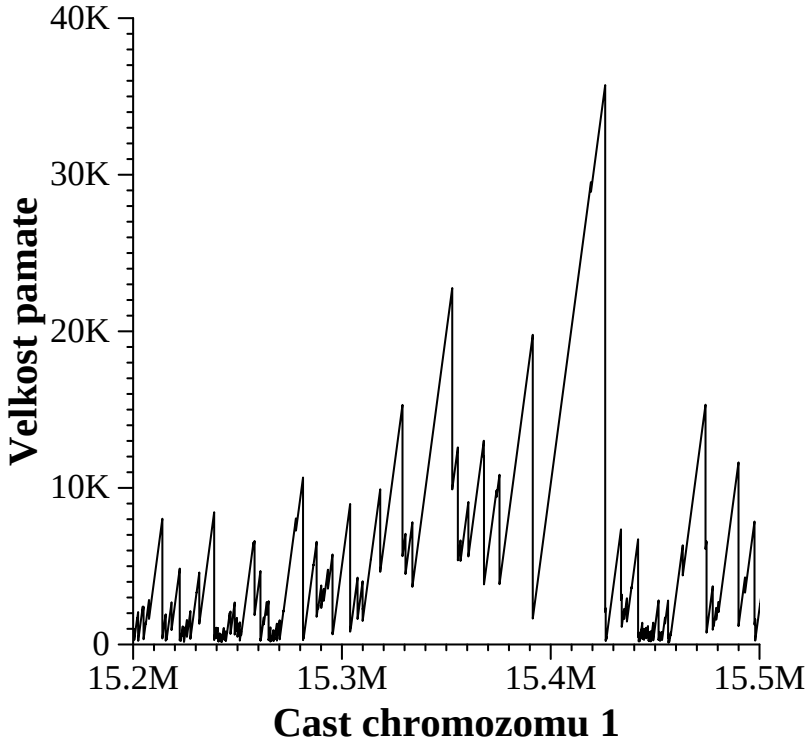


Uvažujme vstupy $1\{1,2\}^{n-2}\{1,2,3\}$

Každý algoritmus potrebuje v najhoršom prípade $\Omega(n)$ pamäť.

Na skutočných dátach ale môžeme pamäť ušetriť

- 256-stavový HMM na hľadanie génov
- 20 MB sekvencie (ľudská DNA)

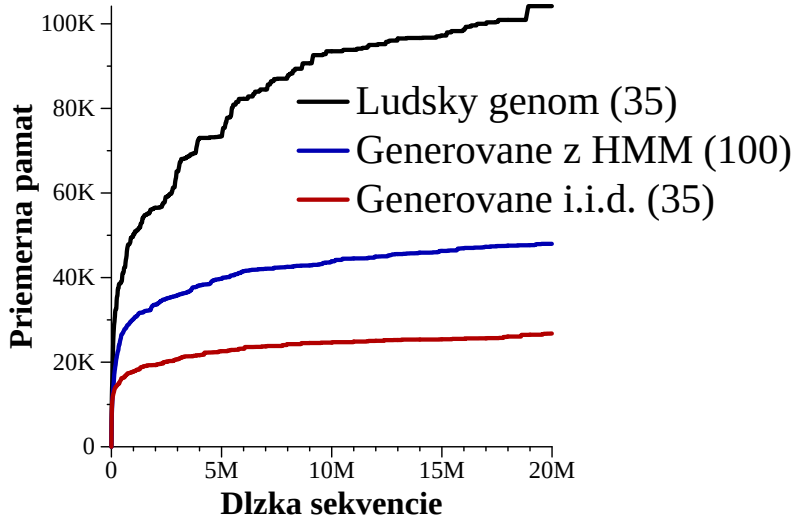


- Priemerná pamäť: $\approx 11\,000$
- Maximálna pamäť: $\approx 222\,000$
- **Priemerná maximálna** pamäť: $\approx 100\,000$

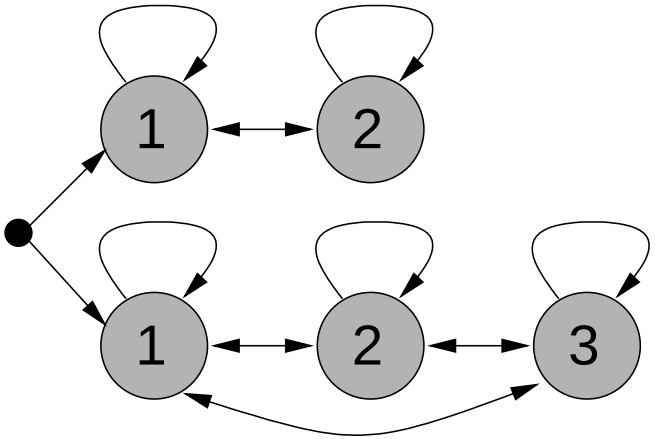
Analýza priemerného prípadu

Odhadujeme strednú hodnotu maximálnej pamäte pre náhodný reťazec

- Rozdelenie reťazcov rovnomerné (i.i.d. znaky) alebo z HMM
- Náhodný reťazec $X = x_1, \ldots, x_n$
- Nech $m(X)$ je (maximálna) pamäť náš algoritmus potrebuje pre X
- Odhadujeme $E(m(X))$ ako funkciu n



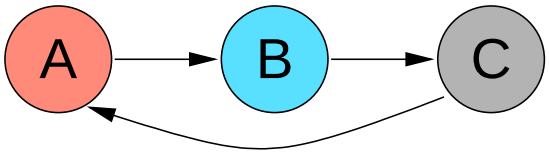
Ďalšie zlé správy



Pre rovnomerné rozdelenie priemerná pamäť $O(1)$

Pre reťazce z tohto HMM pamäť $\Theta(n)$

Ďalšie zlé správy

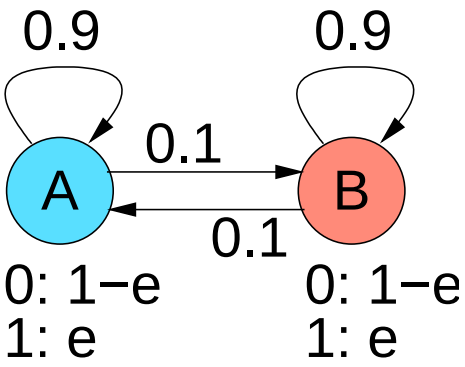


Periodický model: pre ľubovoľný reťazec $\Theta(n)$ pamäť

Tri postupnosti stavov s rôznym začiatkom:

ABCABC. . . , BCABCA. . . , CABCAB. . .

Ďalšie zlé správy



Ekvivalentné stavy: pre ľubovoľný reťazec $\Theta(n)$ pamäť

Dve postupnosti stavov s rovnakou pravdepodobnosťou:

AAA. . . , BBB. . .

Dobré správy

Pre veľa HMM priemerná maximálna pamäť $\mathrm{O}(\log n)$ pre i.i.d. reťazce.

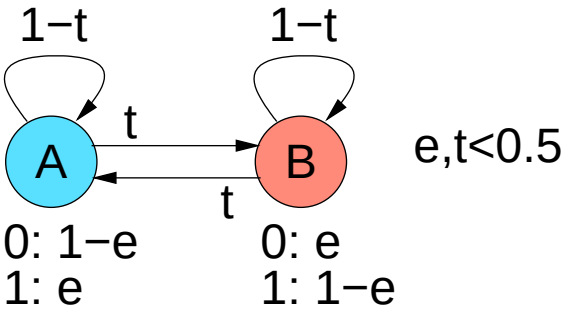
- Rozdeľme X na **bloky**: nový blok ak sútokom klesne pamäť pod c
- Pamäť v bloku dĺžky ℓ najviac $c + \ell = \mathrm{O}(\ell)$

Dobré správy

Pre veľa HMM priemerná maximálna pamäť $O(\log n)$ pre i.i.d. reťazce.

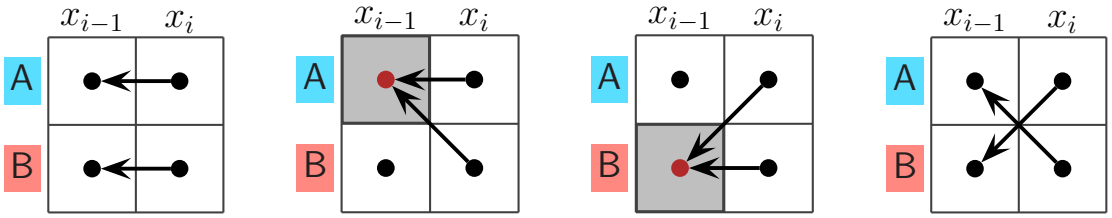
- Rozdeľme X na **bloky**: nový blok ak sútokom klesne pamäť pod c
- Pamäť v bloku dĺžky ℓ najviac $c + \ell = O(\ell)$
- **Cieľ:** ohraničíme pravdepodobnosť výskytu bloku dĺžky ℓ exponenciálne klesajúcou funkciou $a \cdot b^{\ell}$
- Celková pamäť: dĺžka najdlhšieho bloku
  t.j. stredná hodnota maxima z najviac n dĺžok
  $O(\log n)$
- Podobný problém: najdlhší úsek jedničiek v i.i.d. reťazci
  [Guibas, Odlyzko 1980; Gordon, Schilling, Waterman 1986]

Analýza pre 2-stavové symetrické HMM



e,t<0.5

Možné konfigurácie smerníkov:



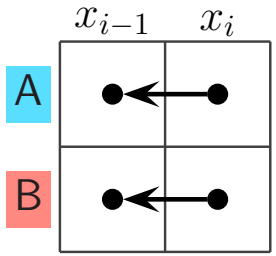
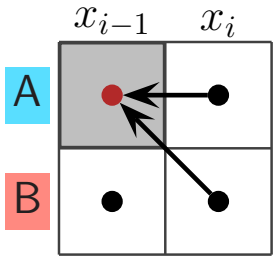
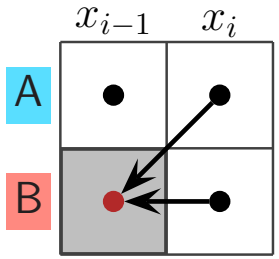
Ktorá konfigurácia?

Záleží na **pomere** $A[i-1,A]$ **a** $A[i-1,B]$:

$$A_{i-1} = \frac{\log A[i-1,A] - \log A[i-1,B]}{\log(1-e) - \log e}$$

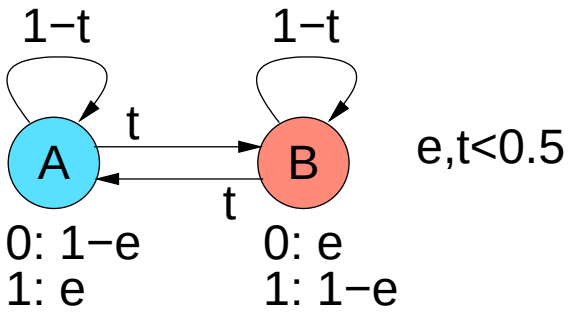
Konfigurácie smerníkov

$$A_{i-1} = \frac{\log A[i-1,A] - \log A[i-1,B]}{\log(1-e) - \log e} \qquad L = \left\lceil \frac{\log(1-t) - \log t}{\log(1-e) - \log e} \right\rceil$$

$-L < A_{i-1} < L$	$A_{i-1} \geq L$	$A_{i-1} \leq -L$
		
$A_i := A_{i-1} \pm 1,$	**sútok**	**sútok**
$+1$ ak $x_i = 0,$	$A_i := L \pm 1$	$A_i := -L \pm 1$
-1 ak $x_i = 1$	podľa x_i	podľa x_i

- Premenná A_i je **náhodná prechádzka** na intervale $(-L, L)$
- Odhadujeme čas, kým narazí na kraj intervalu

Aké dlhé sú bloky?



- Dobre študovaný problém náhodných prechádzok [Feller 1968]
- **Stredná hodnota dĺžky:** $\left\lceil 2\frac{\log(1-t)-\log t}{\log(1-e)-\log e}\right\rceil - 1$
- **Rozdelenie dĺžky ohraničené exponenciálnou funkciou:**

  R_ℓ: pravdepodobnosť dĺžky $2\ell+1$ alebo $2\ell+2$

  $\boxed{b\cdot\alpha^{2\ell} \le P(R_\ell) \le c\cdot\alpha^{2\ell}}$, pre nejaké $b, c > 0, \alpha < 1$
- Dostávame **priemernú pamäť približne** $(2L^2/\pi^2)\ln n$

Zovšeobecnenia

Nesymetrické dvojstavové HMM

$O(\log n)$ odhad platí pre takmer všetky 2-stavové HMM

aj pre reťazce generované HMM

(okrem rôznych symetrií a nulových pravdepodobností)

Viacstavové HMM

Postačujúca podmienka na $O(\log n)$: **synchronizujúci reťazec σ**

- Výskyt σ vždy spôsobí koniec bloku
- Rozdelenie dĺžky medzi výskytmi σ exponenciálne ohraničené

Špeciálny prípad synchronizujúceho reťazca

Pre reťazec α zostrojme **graf** $G(\alpha)$:

vrcholy $=$ stavy modelu

cena hrany $(u, v) =$ log prob. najlepšej postupnosti stavov generujúcej α

Lema: Ak $G(\alpha)$ je silne súvislý a existuje v ňom slučka s cenou ostro väčšou ako priemerná cena každého iného cyklu, α^k je synchronizujúci reťazec pre dosť veľké k.

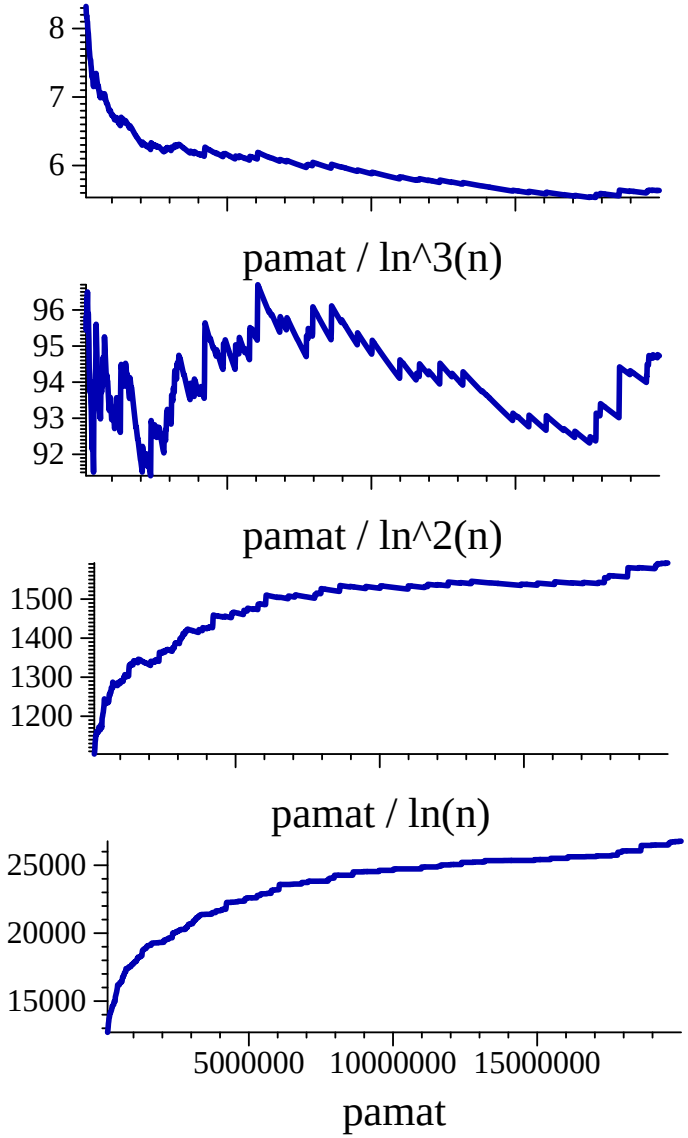
- Sledy dĺžky k v $G(\alpha)$: postupnosti stavov pre α^k
- Pre dosť veľké k najdrahšia cesta krúži okolo najdrahšej slučky
- Dôjde k sútoku najlepších ciest pre rôzne koncové vrcholy

Špeciálny prípad synchronizujúceho reťazca

Lema: Ak $G(\alpha)$ je silne súvislý a existuje v ňom slučka s cenou ostro väčšou ako priemerná cena každého iného cyklu, α^k je synchronizujúci reťazec pre dosť veľké k.

Dá sa pre dané α a model algoritmicky testovať.

Pamäť pre hľadač génov, i.i.d. reťazce



- Je pamäť $O(\log n)$?
- Vieme nájsť synchronizujúci reťazec?
- Alebo je pamäť $\omega(\log n)$?

Zhrnutie

- Viterbiho algoritmus je pre dlhé reťazce pamäťovo náročný
- On-line Viterbiho algoritmus používa pamäť premenlivej veľkosti
- Pri jednoduchom hľadači génov 200-násobná úspora
- Mnohé HMM potrebujú v priemernom prípade pamäť $\Theta(\log n)$
- Najhorší prípad $\Theta(n)$

Otvorené problémy

- Je naša podmienka iba postačujúca? Dá sa zvšeobecniť?
- Je nejaký odhad na dĺžku α, ktoré potrebujeme uvažovať?
- Majú nejaké modely pamäť inú ako $\Theta(1)$, $\Theta(\log n)$, $\Theta(n)$?