

Sztochasztikus modellek a bioinformatikában

1 TÉTEL

• Transzformációs nyelvtanok

- $\{T, N, S, R\}$ transzformációs nyelvtan, ha T és N diszjunkt szimbólumok véges halmaza, $S \in N$, R pedig egy $\alpha \rightarrow \beta$ alakban írt átírási szabályok véges halmaza, ahol $\alpha \in (T \cup N)^*/T^*$, $\beta \in (T \cup N)^*$ (*: az adott ABC feletti véges hosszú szavak halmaza beleértve az üres stringet is). N : nemterminálisok halmaza, T : terminálisok halmaza, S : kezdőnemterminális.
- A nyelvtan egy $A \in T^*$ szekvenciát le tud vezetni, ha létezik $A_1, \dots, A_n$ szekvenciáknak egy sorozata ($n < \infty$) a következő tulajdonságokkal: $A_1 = S, A_n = A$ és minden $1 < i < n$ -re, létezik $\alpha \rightarrow \beta$ szabály és B, C stringek, hogy $A_i = B\alpha C, A_{i+1} = B\beta C$.
- Nem létezik olyan determinisztikus Turing gép, amely minden A stringre és $\{T, N, S, R\}$ nyelvtanra véges időben eldönti, hogy a string a nyelvtan által generált nyelv eleme-e.

• Transzformációs nyelvtanok Chomsky-féle hierarchiája

- **Megszorítás nélküli nyelvtanok**
Az összes lehetséges nyelvtan.
- **Környezetfüggő nyelvtanok**
Azokat a nyelvtanok, ahol az átírási szabálya $\gamma_1 W \gamma_2 \rightarrow \gamma_1 \beta \gamma_2$ alakú, ahol $\gamma_1, \gamma_2 \in (T \cup N)^*$, de nem az üres string.
- **Környezetfüggetlen nyelvtanok**
Azokat a nyelvtanok ahol az átírási szabálya $W \rightarrow \beta$ alakú, ahol $W \in N, \beta \in (T \cup N)^*$, de nem az üres string.
- **Reguláris nyelvtanok**
Azokat a nyelvtanok, ahol az átírási szabálya $W \rightarrow aW'$ vagy $W \rightarrow a$ alakú, ahol $W, W' \in N$, és $a \in T$.

Környezetfüggő nyelvtanokra a levezetési probléma eldönthető (NP-nehéz).

• HMM

- Egy rejtett Markov modell egy $\{G(E, V), \Sigma, T, e\}$ négyes, ahol $G(E, V)$ egy irányított gráf, Σ karakterek véges halmaza, $T : E \rightarrow \mathbb{R}^+$, $e : (\Sigma, V) \rightarrow \mathbb{R}_0^+$ képző függvények, melyekre a következő tulajdonságok teljesülnek:
 - * G egyszerű gráfnak van 2 kitüntetett pontja, START és END, melyeknek rendre a be illetve kifoka 0.
 - * Minden pontra a T egy eloszlást definiál az adott pont kifokain. $T(v|u)$ az (u, v) él valószínűségét jelenti.
 - * A START és az END pontok kivételével minden pontra az e függvény egy eloszlást definiál a karaktereken. $e(\sigma, u)$ azt jelenti, hogy u -ban σ karakter kibocsátásának mennyi a valószínűsége.
- Egy HMM-ben egy kibocsátási út egy random séta a START állapotból az END állapotig T és e valószínűségei alapján.
- Minden olyan Rejtett Markov modell, amelyben nem vezet él a START állapotból az END állapotba, ekvivalens egy sztochasztikus reguláris nyelvtannal, abban az értelemben, hogy létezik egy valószínűségtartó szűrjekció a Rejtett Markov modell kibocsátási útjairól a sztochasztikus reguláris nyelvtan levezetéseire.
- A tétel visszafelé nem igaz.

2 TÉTEL

• Sztochasztikus reguláris nyelvtanok algoritmusai

Egy sztochasztikus reguláris nyelvtan egy $\{T, N, S, R, P\}$ ötös, melyből $\{T, N, S, R\}$ egy reguláris nyelvtan, P pedig $R \rightarrow \mathbb{R}^+$ -ba képező függvény, melyre igaz, hogy minden $W \in N$ -re

$$\sum_{a \in T} \sum_{W' \in N} \mathbb{P}(W \rightarrow aW') + \sum_{a \in T} \mathbb{P}(W \rightarrow a) = 1$$

P -t az átírás valószínűségének nevezzük. Egy levezetés valószínűsége az átírási szabályok valószínűségeinek a szorzata. Egy A szekvencia teljes valószínűsége ($P(A)$) a lehetséges levezetési valószínűségek összege. $r \sim a_k$ -val jelöljük ha A szekvenciának a_k karakterét egy $r \in R$ szabály vezeti le.

– **Viterbi**

Az algoritmus minden $W \in N$ -re és k -ra megkérdezi az $a_1 \dots a_k W$ közti string legvalószínűbb levezetésének valószínűségét. Ennek a jele $v(W, k)$. A kezdeti feltételek: $v(S, 0) = 1, v(W, 0) = 0$ ha $W \neq S$. Ekkor dinamikusan programozva

$$v(W, k) = \max_{W' | W' \rightarrow a_k W} \{v(W', k-1) \mathbb{P}(W' \rightarrow a_k W)\}$$

$$\text{így } P_{\max}(A) = \max_{W | W \rightarrow a_n} \{v(W, n-1) \mathbb{P}(W \rightarrow a_n)\}.$$

– **Forward**

A teljes valószínűséget számolja ki az algoritmus. Jelölje $f(W, k)$ az $a_1 \dots a_k W$ közti string teljes levezetési valószínűségét. Ekkor ugyancsak dinamikus programozással és $f(S, 0) = 1, f(W, 0) = 0$ ha $W \neq S$ kezdőértékekkel a rekurzió a következő:

$$f(W, k) = \sum_{W' | W' \rightarrow a_k W} v(W', k-1) \mathbb{P}(W' \rightarrow a_k W)$$

$$\text{s így } P(A) = \sum_{W | W \rightarrow a_n} v(W, n-1) \mathbb{P}(W \rightarrow a_n)$$

– **Backward**

$P(r \sim a_k | A)$ -ra ad választ az algoritmus. Minden $W \in N$ -re és A^k suffixre (A szekvenciából kivesszük A^k prefixet) kiszámolja, hogy W -ből indulva milyen valószínűséggel vezeti le az adott suffixet. Az algoritmus inicializálása hátulról történik: $b(W, n-1) = \mathbb{P}(W \rightarrow a_n)$. Így a rekurzió

$$b(W, k) = \sum_{W' | W \rightarrow a_{k+1} W'} b(W', k+1) \mathbb{P}(W \rightarrow a_{k+1} W')$$

amiből $P(A) = b(S, 0)$. Ezeket felhasználva ha $r = W \rightarrow a_k W'$, akkor

$$\mathbb{P}(r \sim a_k | A) = \frac{f(W, k-1) \mathbb{P}(W \rightarrow a_k W') b(W', k)}{P(A)}$$

Ezt poszterior valószínűségnek hívjuk.

• **HMM-ek algoritmusai**

– **Viterbi**

Az algoritmusban $v(u, k)$ jelöli annak az útnak a valószínűségét, amely az A^k prefixet bocsájtotta ki, jelenleg az u állapotban van, és itt már bocsájtott már ki karaktert. Az algoritmus inicializációja: $v(START, 0) = 1$ és $v(u, 0) = 0$ ha $u \neq START$. Így a rekurzió

$$v(u, k) = \max_w \{v(w, k-1) T(u|w) e(a_k|u)\}$$

amiből $P_{max}(A) = \max_u \{v(u, n) T(END|u)\}$ következik.

– **Forward**

Az algoritmusban $f(u, k)$ jelöli azon utak valószínűségeinek az összegét, amelyek az A^k prefixet bocsájtották ki, jelenleg az u állapotban vannak, s itt már bocsájtott ki karaktert. Az algoritmus inicializációja: $f(START, 0) = 1$ és $f(u, 0) = 0$ ha $u \neq START$. Így a rekurzió

$$f(u, k) = \sum_w f(w, k-1) T(u|w) e(a_k|u)$$

amiből adódik a $P(A) = \sum_u f(u, n) T(END|u)$

– **Backward**

Az algoritmusban $b(u, k)$ jelöli azon utak valószínűségeinek az összegét, amelyek az A^k suffixet bocsájtják ki az u állapotból indulva, s itt már bocsájtott ki karaktert. Az algoritmus inicializációja: $b(u, n) = T(END|u)$. A rekurzió

$$b(u, k) = \sum_w b(w, k+1) T(w|u) e(a_{k+1}|w)$$

s innen a termináció $P(A) = b(START, 0)$ adódik. Ezeket felhasználva

$$\mathbb{P}(v \sim a_k | A) = \frac{f(v, k)b(v, k)}{P(A)}$$

ahol $v \sim a_k$ az, hogy v bocsájtotta ki a_k -t.

3 TÉTEL

• Expectation Maximization

Ha a levezetési valószínűségek, illetve ugrási és kibocsájtási valószínűségek nincsenek meghatározva, ezeket megbecsülhetjük a rendelkezésre álló adatokból. Azon paramétereket szeretnénk meghatározni, amelyek maximalizálják a likelihoodot. Az Expectation Maximization metódus egy rekurzív módszer, melyre igaz, hogy $\mathbb{P}(D|\theta_i) \leq \mathbb{P}(D|\theta_{i+1})$.

– Sztochasztikus reguláris nyelvtan

- * Egy nemterminális használatának a várható értéke, feltéve, hogy a nyelvtan az adott szekvenciát generálta. Jelölje $n(W, d)$ azt, hogy a d levezetésben hányszor használtuk a W nemterminált. Ekkor a várhatóérték definíció szerint:

$$\mathbb{E}(W|A) = \frac{\sum_{d|d \sim A} P(d)n(W, d)}{P(A)}$$

- * Egy r átírási szabály használatának a várható értéke, feltéve, hogy a nyelvtan az A szekvenciát vezeti le. Jelölje $n(r, d)$ azt, hogy egy d levezetésben az r szabályt hányszor használtuk. Ekkor a várható érték definíció szerint:

$$\mathbb{E}(r|A) = \frac{\sum_{d|d \sim A} P(d)n(r, d)}{P(A)}$$

– HMM

- * Egy adott állapot használatának a várható értéke, feltéve, hogy az A szekvenciát bocsájtja ki a modell. Jelölje $n(W, p)$ W használatának a számát. Ekkor a feltételes várható érték definíció szerint:

$$\mathbb{E}(W|A) = \frac{\sum_{p|p \sim A} P(p)n(W, p)}{P(A)}$$

- * Egy adott átmenet használatának a várható értéke, feltéve, hogy az A szekvenciát bocsájtja ki a modell. Jelölje $n(W_1 \rightarrow W_2, p)$ a W_1 -ből W_2 -be ugrások számát. Ekkor a feltételes várható érték definíció szerint:

$$\mathbb{E}(W_1 \rightarrow W_2|A) = \frac{\sum_{p|p \sim A} P(p)n(W_1 \rightarrow W_2, p)}{P(A)}$$

- * Egy adott karakternek egy adott állapot általi kibocsájtásának a várható értéke, feltéve, hogy az A szekvenciát bocsájtja ki a modell. Jelölje $(W \sim a, p)$ azt, hogy W hányszor bocsájtott ki egy a karaktert a p útban. Ekkor a feltételes várható érték definíció szerint:

$$\mathbb{E}(W \sim a|A) = \frac{\sum_{p|p \sim A} P(p)n(W \sim a, p)}{P(A)}$$

- Ekkor a becslések a következőképpen néznek ki:

$$\hat{P}(W_1 \rightarrow aW_2) = \frac{\mathbb{E}(r|A)}{\mathbb{E}(W_1|A)} \quad \hat{T}(W_2|W_1) = \frac{\mathbb{E}(W_1 \rightarrow W_2|A)}{\mathbb{E}(W_1|A)} \quad \hat{e}(a|W) = \frac{\mathbb{E}(W \sim a|A)}{\mathbb{E}(W|A)}$$

- Legyen $\{T, N, S, R, P\}/\{G(E, V), \Sigma, T, e\}$ egy sztochasztikus reguláris nyelvtan/HMM, melyben a levezetési/ ugrási&kibocsátási valószínűségek halmazát jelölje θ_1 . Jelölje θ_2 azon levezetési/ ugrási&kibocsátási valószínűségek halmazát, amelyeket az előző képlettel kapunk meg. Ekkor $P(A|\theta_1) \leq P(A|\theta_2)$, és egyenlőség csak $\theta_1 = \theta_2$ esetén van.

4 TÉTEL

- A várhatóértékek kiszámolása a Viterbi, Forward és Backward algoritmusok segítségével:

$$\mathbb{E}(W|A) = \frac{\sum_{k=0}^{|A|} f(W, k)b(W, k)}{P(A)} \quad \mathbb{E}(W \sim a|A) = \frac{\sum_{k=1|a_k=a}^{|A|} f(W, k)b(W, k)}{P(A)}$$

$$\mathbb{E}(W_1 \rightarrow aW_2|A) = \frac{\sum_{k=1}^{|A|} f(W_1, k-1)P(W_1 \rightarrow aW_2)b(W_2, k)}{P(A)}$$

$$\mathbb{E}(W_1 \rightarrow W_2|A) = \frac{\sum_{k=0}^{|A|-1} f(W_1, k)T(W_2|W_1)e(a_{k+1}|W_2)b(W_2, k+1)}{P(A)}$$

- **Posterior decoding**

Jelölje $W \sim (i, j)$ azt az eseményt, hogy a W nemterminális az i -edik pozíciótól a j -edik pozícióig terjedő részstringet vezeti le. Ekkor

$$P(W \sim (i, j)|A) = \frac{I(W, i, j)O(W, i, j)}{P(A)}$$

- **Expectation Maximization**

$$\hat{P}(W \rightarrow a) = \frac{\sum_i P(W \rightarrow a)O(W, i, j)}{\sum_{i \leq j} I(W, i, j)O(W, i, j)}$$

$$\hat{P}(W \rightarrow W_1W_2) = \frac{\sum_{k=i}^{j-1} O(W, i, j)P(W \rightarrow W_1W_2)I(W_1, i, k)I(W_2, k+1, j)}{\sum_{i \leq j} I(W, i, j)O(W, i, j)}$$

5 TÉTEL

- **Sztochasztikus környezetfüggetlen nyelvtanok**

Egy sztochasztikus környezetfüggetlen nyelvtan egy $\{T, N, S, R, P\}$ ötös, melyből $\{T, N, S, R\}$ egy környezetfüggetlen nyelvtan, $P: R \rightarrow \mathbb{R}^+$, melyre igaz, hogy $\forall W \in N$ esetén $\sum_{\beta|W \rightarrow \beta} P(W \rightarrow \beta) = 1$.

– A környezetfüggetlen nyelvtanok a regulárisak felett helyezkednek el.

- **Levezetési fa**

Mivel nem akarunk megkülönböztetni két levezetést, amely ugyanazokat a levezetési lépéseket tartalmazzák, csak más sorrendben, ezért bevezetjük a levezetési fát. A gyökér a kezdő nemterminalissal van felcímkézve, a levelek terminális karakterekkel, és minden belső pont fokszáma a hozzá tartozó levezetésben szereplő β string hossza. β minden egyes karakterére van egy kimenő él, a végén levő pont az adott karakterrel van felcímkézve. Egy levezetési fa valószínűsége a benne szereplő levezetések valószínűségének a szorzatai, egy A szekvencia levezetésének a valószínűsége az összes olyan levezetési fa valószínűségének az összege, amely A -t vezeti le.

- **Chomsky normálalak**

Egy környezetfüggetlen nyelvtan Chomsky normál formában van, ha minden átírási szabálya $W \rightarrow W_1W_2|a$ formában írható fel.

– Minden $NY = \{T, N, S, R, P\}$ sztochasztikus környezetfüggetlen nyelvtan ekvivalens egy Chomsky normál formában levő $NY' = \{T, N', S, R', P'\}$ sztochasztikus környezetfüggetlen nyelvtannal abban az értelemben, hogy létezik egy valószínűségtartó szűrjekció NY lehetséges levezetési fáiról NY' lehetséges levezetési fáira.

- **Sztochasztikus környezetfüggetlen nyelvtanok algoritmusai**

- **CYK**

Az algoritmus egy Chomsky normál formában levő környezetfüggetlen nyelvtanra és egy szekvenciára megadja annak legvalószínűbb levezetési fáját. Jelölje $c(W, i, j)$ a legvalószínűbb rész-levezetési fának a valószínűségét, amelyik a W nemterminálisból vezeti le az i -től j -edik pozícióig terjedő részstringet. Az inicializáció $c(W, i, i) = P(W \rightarrow a_i)$, így a rekurzió

$$c(W, i, j) = \max_{W_1} \max_{W_2} \max_{i \leq k < j} \{P(W \rightarrow W_1W_2)c(W_1, i, k)c(W_2, k+1, j)\}$$

amit felhasználva a termináció: $P_{max}(A) = c(S, 1, n)$.

- **Inside**

Az algoritmus a teljes valószínűséget adja megegy szekvenciának. Jelölje $I(W, i, j)$ a lehetséges rész-levezetési fák valószínűségeinek az összegét, amelyek a W nemterminálisból vezetik le az i -től j -edik pozícióig terjedő részstringet. Az inicializáció $I(W, i, i) = P(W \rightarrow a_i)$, így a rekurzió

$$\sum_{W_1} \sum_{W_2} \sum_{k=i}^{j-1} P(W \rightarrow W_1W_2)I(W_1, i, k)I(W_2, k+1, j)$$

s ebből adódik, hogy $P(A) = I(S, 1, n)$.

– **Outside**

Az algoritmus kívülről befelé haladva számolja ki a teljes levezetési valószínűséget. Jelölje $O(W, i, j)$ az $a_1 \dots a_{i-1} W a_{j+1} \dots a_n$ közti string összes lehetséges levezetési fája valószínűségeinek az összegét. Az inicializáció $O(S, 1, n) = 1$, illetve $O(W, 1, n) = 0$ minden $W \neq S$ esetén. Így a rekurzió

$$O(W, i, j) = \sum_{W_1} \sum_{W_2} \sum_{k=1}^{i-1} P(W_1 \rightarrow W_2 W) I(W_2, k, i-1) O(W_1, k, j) + \sum_{W_1} \sum_{W_2} \sum_{k=j+1}^n P(W_1 \rightarrow W W_2) I(W_2, j+1, k) O(W_1, i, k)$$

6 TÉTEL

• **Szubsztitúciók modellezése**

- A szubsztitúciók modellezésére leggyakrabban a folytonos idejű Markov modelleket szokták használni. A Markov modell feltételezi, hogy egy a karakter b -re való cseréjének a valószínűsége csak a -tól és b -től függ, és attól nem, hogy az adott pozícióban az a karakter előtt milyen karakter volt. Ez a valóságban persze nem így van, azonban, a Markov tulajdonság esetén a modellekkel való számolás kevésbé számolásigényes, és ez mindig döntő jelentőségű a bioinformatikai modellekben.
- Mivel a biológiai szekvenciák esetében véges sok karakter van, a Markov modell véges állapotú. Ezt egy lineáris differenciálegyenlet-rendszerrel lehet megadni, mátrix-vektor formában: $\frac{dx}{dt} = Qx$, ahol Q a modellt leíró rátamátrix, és x tartalmazza az egyes állapotok valószínűségeit. Kezdeti érték feltételnek az i -edik egységvektort választva a differenciálegyenlet-rendszer megoldása megadja annak a valószínűségét, hogy az i -edik állapotból kiindulva t idő elteltével melyik állapotnak mekkora a valószínűsége. A differenciálegyenlet-rendszer általános megoldása:

$$x(t) = e^{Qt} x(0) = \sum_{n=0}^{\infty} \frac{(Qt)^n}{n!} x(0)$$

- Egy folytonos idejű Markov model reverzibilis, ha letezik egy π eloszlásfüggvény, hogy minden i -re és j -re teljesül a részletes egyensúly, azaz $\pi(i)q_{j,i} = \pi(j)q_{i,j}$.
- Mivel Q sorainak az összege 0, így a rangja kisebb, mint k , így a hozzá hasonló diagonális mátrixnak is kisebb a rangja, mint k , amiből következik, hogy legalább az egyik sajátérték 0. Ha ennek a sajátértéknek a multiplicitása 1, akkor $\sum_1^k x_i(0) = 1$ feltétel mellett, ahol minden $x(0) \geq 0$, létezik egyetlen olyan egyensúlyi állapot, mely globálisan stabil. Az i -dik állapot egyensúlyi valószínűségét jelölje π_i .
- Egyensúlyi állapotban egységnyi idő alatt átlagosan egy mutáció történik. Általános jelenség, hogy az egyes folyamatok rátáját és az evolúciós időt nem lehet külön-külön becsülni, csupán csak a szorzatukat. Ekkor st -re adnak Maximum Likelihood becslést ($Q = Q_0 s$).

• **Felsenstein algoritmus**

- A szubsztitúciós modell kiterjesztése kettőről több szekvenciára a következőképpen tehető meg: csak azokat a pozíciókat tekintjük, ahol az illesztésben nincs beszúrás/törlés. Feltesszük, hogy az egyes pozíciók egymástól függetlenül evolválódtak, így egy evolúciós folyamat valószínűsége az egyes pontokon történt események valószínűségeinek a szorzata. Legyen adva egy gyökeres bináris fa, amely reprezentálja a szekvenciák leszármazási sorrendjét. Feltesszük azt is, hogy a fa egyes élein a karakterek szubsztitúciós folyamata független a többi élen történő folyamatoktól, és minden élre egy adott (általában ugyan az) szubsztitúciós modell szerint megy. A fa gyökerében a folyamat minden pozícióra az egyensúlyi állapotból indul. A teljes szekvenciák valószínűségeinek az észlelése az egyes pozíciókra vett észlelési valószínűségek szorzata.
- Elég azt megmutatni, hogy hogyan kell egy pozícióra a karakterek észleléseinek a valószínűségeit kiszámolni. Jelölje $f(c, v)$ annak a valószínűségét, hogy a v pont alatti részfa levelein az adott karaktereket észleljük, feltéve, hogy a folyamat a c karakterből indult a v pontban. Az algoritmus inicializálása az a karaktert tartalmazó v levélre: $f(c, v) = 1$ ha $c = a$, egyébként 0. Ekkor a rekurzió egy belső v pontra, melynek u_1, u_2 a gyermekei, illetve az ide futó éleken eltelt idő t_1, t_2 :

$$f(c, v) = \sum_{c_1} P(c_1|c, t_1) f(c_1, u_1) \times \sum_{c_2} P(c_2|c, t_2) f(c_2, u_2)$$

ahol $P(c_i|c, t_i)$ jelöli annak a valószínűségét, hogy a c karakterből indulva a Markov folyamat t_i idő elteltével c_i karakterben van. Ebből a teljes valószínűség: $P = \sum_c f(c, root) \pi(c)$.

– **Emelő elv**

Ha a szubsztitúciót leíró modell reverzibilis, akkor teljesül az emelő elv. Ez azt mondja ki, hogy ha a fa gyökeréből kiinduló két él közül az egyik hosszát lecsökkentjük, a másikat pedig megnöveljük, akkor a fa likelihood-ja nem változik meg. Az emelő elv folyamánya, hogy egy adott fára az észlelési valószínűség független attól, hogy hol gyökereztetjük le, és egy leggyökereztetett fán az észlelési valószínűség megegyezik a gyökerezetlen fán vett észlelési valószínűséggel, ahol a folyamatot bármelyik belső pontból indíthatjuk az egyensúlyi eloszlásból kiindulva.

7 TÉTEL

• **RNS**

- Az RNS szekvenciák a DNS-sel ellentétben egyszálú nukleinsav szekvenciák az $\{A, C, G, U\}$ ABC felett, és az RNS szekvencia visszakanyarodva önmagával képezhet nukleinsavpárokat.
- A térszerkezet azt írja le, hogy melyik pozícióban levő nukleinsav melyik pozícióban levő nukleinsavval párosodik.
- A bioinformatikában az RNS-ek álcsumómentes másodlagos térszerkezeteit le lehet írni környezetfüggetlen nyelvtanokkal.
- Egy RNS térszerkezet egy N pozitív egész szám és pozitív egészek párjaiból álló rendezetlen pároknak a véges halmaza, amelyben minden egész szám legfeljebb egyszer szerepel, és minden szám kisebb vagy egyenlő N -nél.
- Egy RNS térszerkezet álcsumó-mentes, ha bármely két pár által kifesztett intervallumok nem lehetnek valódi metszők.
- Minden álcsumómentes RNS térszerkezet egyértelműen leírható a $\{., (,)\}$ feletti ABC N hosszú szekvenciájával, amelyikben minden (i, j) párra az i -edik pozícióban egy $($ áll, a j -edik pozícióban $)$, és minden olyan pozícióban, és amelyik pozíció nem vesz részt egyik párban sem, $.$ áll.

• **Knudsen-Hein nyelv**

- Az álcsumómentes RNS térszerkezetek leírhatóak környezetfüggetlen nyelvtanokkal. KH által generált nyelv pontosan azokból az álcsumómentes RNS térszerkezetekből áll, amelyben minden (i, j) párra $j - i \geq 3$. A nyelvtan: $S \rightarrow L|LS, L \rightarrow \cdot|(F), F \rightarrow (F)|LS$.
- Ez a nyelvtan kiterjeszthető úgy, hogy egy többszörös szekvenciaillesztést generáljon, amelyben a szekvenciaillesztés oszlopainak levezetési valószínűségei a nyelvtanban a megfelelő levezetési valószínűség és egy szubsztitúciós modellben való észlelési valószínűségek szorzata.

• **Összehasonlító bioinformatika**

- Alapelve az, hogy a struktúra mindig konzervatívabb, mint a szekvencia.
- Az összehasonlító szerkezetpredikcióra két metódus létezik.
 - * Két olyan szekvenciát hasonlítunk össze, amelyekből az egyiknek a szerkezete ismert, és ezt próbáljuk a másik szekvenciára rávetíteni.
 - * Egyik szekvenciának a szerkezete sem ismert, de feltesszük, hogy mindkettő ugyanaz.

• **Goldman-Thorne-Jones**

- Első modell, amelyben összekapcsolták a transzformációs nyelvtanokat az időfolytonos Markov modellekkel.
- Felhívta a figyelmet arra, hogy evolúciós információkat felhasználva javítani lehet a struktúrára vonatkozó predikciókat.
- A modell egy olyan Markov folyamat, mely három emittáló állapota van, α, β , és hurok, melyek a három alapvető másodlagos térszerkezeti kategóriát reprezentálják.
- A szekvenciák egy evolúciós fa levelein vannak, és a rejtett Markov modellben az egyes emissziók valószínűsége a karaktereknek az evolúciós fa levelein való észlelési valószínűségeként van definiálva.

• **Génkeresés páros HMM-el**

- Minden faj genomjában vannak olyan szakaszok, amelyeknek a szerepe tisztázatlan, illetve nem tartalmaznak érdemi információt. A maradék részben vannak a gének, amelyek a faj által termelt fehérjéket kódolják.

- A páros HMM olyan HMM, amelyek két szekvenciát bocsájtanak ki. Bizonyos állapotok csak az egyik, más állapotok csak a másik, míg olyan állapotok is vannak, melyek mindkettő féle szekvenciába bocsájtanak ki karaktereket. A szemlélő csak a kibocsájtott szekvenciákat látja, az együttes kibocsájtási mintázatot nem.
- Páros HMM-kel lehet megadni beszúrák és törlések egyszerű sztochasztikus modelljeit is.
- Irmtraud Meyer és Richard Durbin adott meg egy olyan páros HMM-t, amelynek a segítségével lehet génszerkezetet prediktálni.

8 TÉTEL

• TKF91

- A beszúrák és törlések leíró folytonos idejű Markov modell, melyet Thorne-Kishino-Felsenstein 1991-es publikációja után neveztek el.
- Két szekvencia kapcsoltsági valószínűsége: $P_t(A, B) = \sum_C P_\infty(C) P_t(A|C) P_t(B|C)$ ahol $P_t(A|C)$ annak a valószínűsége, hogy a C szekvencia t idő alatt A -vá evolválódott, $P_\infty(C)$ pedig a C szekvencia egyensúlyi valószínűsége.
- Egy tetszőleges tranzíció reprezentálható egy α illesztéssel. Az illesztés valószínűsége a tranzíció valószínűsége szorozva az ősi szekvencia valószínűségével.
- A beszúrák és törlés folyamatát képzeletbeli linkek segítségével írjuk le. A szekvencia minden egyes karaktere asszociálva van egy halandó linkkel, mely a karakter jobb oldalán található. Ezenkívül a szekvencia rendelkezik egy halhatatlan linkkel is, amely a szekvencia bal végén található. A linkek függetlenek. Minden link képes új link szülésére, melyek az eredeti link jobboldalán foglalnak helyet, s össze vannak kapcsolva egy karakter születésével, mely az újszülött link bal oldalán helyezkedik el. A karakter típusát a szubsztitúciós model egyensúlyi eloszlása szabja meg. Ilyen módon történő választása nem zavarja meg az egyensúlyt.
- A születési rátát λ -val, a halálozási rátát pedig μ -vel jelöljük. Többszörös beszúrák és törlések nem lehetségesek ebben a modellben. Egy n hosszúságú szekvencia növekedési rátája $(n+1)\lambda$, s a csökkenési rátája pedig $n\mu$.
- Nem világos, hogy a hagyományos értelemben az illesztések különböznek egymástól, de az itt bemutatott modell alapján a bármely két illesztés világosan különbözik.
- Figyelemmel kísérve az egyes linkek sorsát, három fajta valószínűségi függvényt kell számításba venni: $p_n(t)$ fogja jelölni a valószínűségét annak, hogy egy halandó link t idő elteltével túlél, és saját magát is beleszámítva n utódja van a t -edik időpillanatban. $p'_n(t)$ jelöli a valószínűségét annak, hogy egy link meghal t idő alatt, de n utódot hagy maga után, és végül $p''_n(t)$ fogja jelölni annak a valószínűségét, hogy a halhatatlan linknek a t -edik időpillanatban n utódja van, önmagát is beleértve. Definíció szerint $p_0(t) = p'_0(t) = 0$. A többi függvényt ez a differenciálegyenlet rendszer írja le:

$$\frac{dp_n(t)}{dt} = \lambda(n-1)p_{n-1}(t) - (\lambda + \mu)np_n(t) + \mu np_{n+1}(t) \text{ ahol } n > 0$$

$$\frac{dp'_n(t)}{dt} = \lambda(n-1)p'_{n-1}(t) - (\lambda + \mu)np'_n(t) + \mu(n+1)p'_{n+1}(t) + \mu p_{n+1}(t) \text{ ahol } n > 0$$

$$\frac{dp''_n(t)}{dt} = \lambda(n-1)p''_{n-1}(t) - (\lambda n + \mu(n-1))p''_n(t) + \mu np''_{n+1}(t) \text{ ahol } n > 0$$

illetve $\frac{dp'_0(t)}{dt} = \mu p'_1(t) + \mu p_1(t)$, továbbá a kezdeti értéktételek:

$$p_1(0) = p'_1(0) = 1 \quad p_n(0) = p''_n(0) = 0 \text{ ha } n = 2, 3, \dots \quad p'_n(0) = 0 \text{ ha } n = 0, 1, \dots$$

Ez egy végtelen dimenziós rendszer, de átfogalmazva sorbanállási rendszerek nyelvére a problémát könnyebb megoldani.

• Sorbanállási rendszerek

- Adott egy rendszer, amelyben egy vagy több kiszolgáló egység található. A rendszerbe vevők érkeznek, akiket ki kell szolgálni. Egy kiszolgáló egység csak egyetlen vevőt tud egyszerre kiszolgálni. Ha minden kiszolgáló egység foglalt, akkor az újonnan érkezett vevőknek várakozniuk kell, ha van szabad, akkor a vevő kiszolgálása a beérkezéssel egy időben elkezdődik. Mind a kiszolgálást, mind a vevők beérkezését Poisson folyamatként képzeljük el. A rendszer állapotát a rendszerben tartózkodó vevők számával lehet leírni. A kiszolgálási időt és az érkezési időt leíró valószínűségi változók függhetnek a rendszer állapotától.

- Ha a TKF modellt átfogalmazzuk a sorban állási rendszerek nyelvére, akkor a halandó linkek lesznek a vevők. Egy vevő kiszolgálása egy link halála. Egy vevő kiszolgálási idejét egy λ parameterű exponenciális eloszlású valószínűségi változó írja le, és minden egyes újonnan beérkező vevő kiszolgálása azonnal elkezdődik, függetlenül a rendszerben tartózkodó vevők számától, azaz a rendszer végtelen sok kiszolgálóegységgel van felszerelve. Egy link születése a sorban állási rendszerekben egy új vevő érkezését jelenti. Egy link születési rátája egyenesen arányos a szekvenciában meglevő összes linkek számával azaz a modellben felbátorodó vevőkről van szó. A TKF modellben illeszteni kell a szekvenciákat, ezért a sorban állási rendszerben kíváncsiak vagyunk a részállapotok valószínűségeire is. Ezt úgy lehet leírni, hogy minden vevőhöz λ parameterű Poisson folyamattal csatlakozik egy ismerős. Ezenkívül van egy 'bolti csalogató', aki szinten λ parameterű exponenciális várakozási idővel csalogatja be egymás után a vevőket. Ez a 'csalogató' a hallhatatlan link.

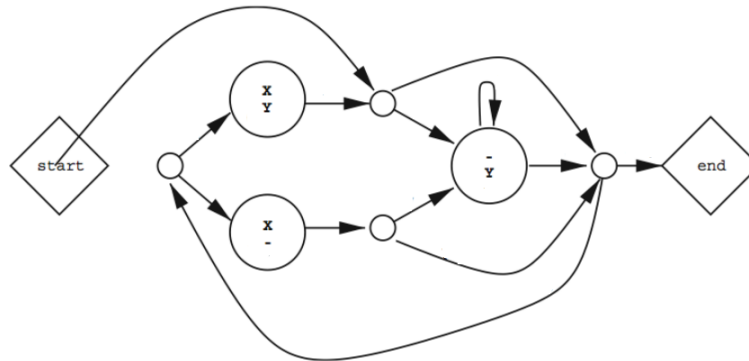
• Megoldás

$$p_n(t) = e^{\mu t}(1-\lambda\beta(t))(\lambda\beta(t))^{n-1}, p'_n(t) = (1-\mu\beta(t)-e^{\mu t})(1-\lambda\beta(t))(\lambda\beta(t))^{n-1}, p''_n(t) = (1-\lambda\beta(t))(\lambda\beta(t))^{n-1}$$

9 TÉTEL

• TKF92

- Az eredeti modell egy javított változatát egy évvel később publikálták, amely megenged többszörös beszúrásokat és törléseket. Ez lehetőséget ad arra, hogy egy link ne egy, hanem tetszőleges számú karakterrel legyen asszociálva. Ekkor azonban a szekvencia széttörhetetlen fragmentumokból fog állni: ha egy link a születésekor több karakterrel asszociálódott, akkor nincs lehetőség arra, hogy ezek a karakterek külön-külön törölődjenek. Bár a model biológus szemmel nézve nem realiztikus, mégis jobb, mint az elődje.
- A fragmentumok hossza r paraméterű geometriai eloszlású.
- A szekvenciák hosszának egyensúlyi eloszlása: $\gamma_0 = 1 - \frac{\lambda}{\mu}, \gamma_n = (1 - \frac{\lambda}{\mu})\frac{\lambda}{\mu}(1-r)(\frac{\lambda}{\mu}(1-r) + r)^{n-1}$
- Két szekvencia kapcsoltsági valószínűségének a kiszámolásához összegezni kell az összes lehetséges fragmentálódáson és az összes lehetséges tranzíción.
- Σ ABC feletti szekvenciákat kibocsájtó páros HMM ekvivalens a TKF91/TKF92 modellel, abban az értelemben, hogy létezik a páros HMM kibocsájtási útjai és a Σ ABC feletti szekvenciák illesztései között egy valószínűségtartó bijekció.



• Egy állapot rekurzio

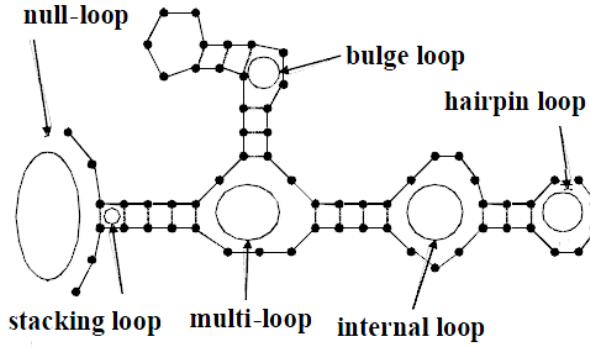
- Mivel mind a TKF91, mind a TKF92 modell ekvivalens egy-egy páros HMM-el, a kapcsoltsági valószínűségek kiszámolhatóak a Forward algoritmussal is, a legvalószínűbb illesztés pedig a Viterbi algoritmussal. A TKF91 modell esetében a Forward algoritmusnál egy picit hatékonyabb algoritmus is létezik.
- Az elnevezés arra utal, hogy a TKF91 modellben a kapcsoltsági valószínűség kiszámolható úgy is, hogy nem osztjuk a lehetséges illesztéseket 3 halmazra, illetve a páros HMM 3 állapotára nem kell külön-külön számolni. Az algoritmus: $P(0,0) = \left(1 - \frac{\lambda}{\mu}\right)(1 - \lambda\beta(t))$, ahol $P(i,j)$ az i és j hosszú prefixek kapcsoltsági valószínűségét adja meg.

$$P(i,j) = P(i-1,j)\lambda\beta(t)\pi(a_i) + P(i,j-1)\lambda\beta(t)\pi(b_j) + P(i-1,j-1)\pi(a_i)(e^{-\mu t}(1-\lambda\beta(t))T(b_j|a_i,t) - \lambda\beta(t)\mu\beta(t)\pi(b_j))$$

10 TÉTEL

• Zuker-Tinoco energiamodell

- Az energiamodell az álcsonomómentes RNS térszerkezetek szabadenergiáira ad meg egy értéket. Ebben a modellben minden érszerkezetet szétdarabolható körökre, és a térszerkezet szabadenergiája az egyes körökhöz rendelt szabadenergiák összege. Egy S térszerkezet szabadenergiáját $\Delta G(S)$ fogja jelölni, ami az S térszerkezet szabadenergiája és azon térszerkezet szabadenergiája közötti különbség, amelyben egyetlen bázispárosodás sincsen.
- Egy (i, j) bázispár kisebb egy (i', j') bázispárnál, ha az (i, j) intervallumot magába foglalja az (i', j') intervallum.
- **Loopok**



- * Egy álcsonomómentes térszerkezet null-loopja a maximális bázispárokból és azon pozíciókból áll, amelyek nincsenek benne egyetlen bázispár intervallumában.
- * Egy álcsonomómentes térszerkezetben minden $(i, i+1, j-1, j)$ négyes stackingloopot alkot, amelyben mind (i, j) , mind $(i+1, j-1)$ bázispár.
- * Egy álcsonomómentes térszerkezetben minden $(i, i+1, k, k+1, \dots, j-1, j)$ pozíciók, valamint minden $(i, i+1, \dots, k'-1, k', j-1, j)$ pozíciók bulge-loopot alkotnak, amelyre mind (i, j) , mind $(i+1, k)$ vagy $(k', j-1)$ bázispárt alkot, valamint k és j , illetve i és k' közötti pozíciók egyike sem szerepel bázispárban.
- * Egy álcsonomómentes térszerkezetben minden $(i, i+1, \dots, k-1, k, l, l+1, \dots, j-1, j)$ pozíciók egy internal-loopot alkotnak, amelyre mind (i, j) , mind (k, l) bázispárt alkot, valamint mind az i és k , mind az l és j közötti pozíciók egyike sem vesz részt bázispárban.
- * Egy álcsonomómentes térszerkezetben minden (i, j) bázispárra az i -től j -ig terjedő szakaszban azok a pozíciók, amelyek nem szerepelnek egyetlen, (i, j) -nél kisebb bázispárosodás intervallumában, valamint az (i, j) által fedett bázispárok (i, j) -vel együtt egy multi-loop-ot alkotnak, amennyiben az (i, j) által fedett bázispárok száma legalább 2.
- * Egy álcsonomómentes térszerkezetben minden (i, j) bázispárra az i -től j -ig terjedő szakasz az (i, j) bázispárral egy hairpin-loopot alkot, ha a kérdéses szakaszban egyik pozíció sem vesz részt bázispárban.
- A Zuker-Tinoco energiamodellnek egy a null-loop szabadenergiája 0.
- Egy L multi-loop-ra számolt szabadenergia tag lineáris közelítése: $\Delta G(L) = a + bl_s(L) + cl_d(L)$, ahol a, b, c konstansok, $l_s(L)$ a nem párosodó nukleinsavak száma, $l_d(L)$ a párosodó nukleinsav-párok száma.
- A többi kör energiája csak a benne szereplő bázispároktól függ.

• Zuker-Tinoco algoritmus

- Egy RNS szekvenciára megadja a Zuker-Tinoco energiamodellben minimális szabadenergiájú térszerkezetet.
- Minden térszerkezetet pontosan egyszer vesz figyelembe.
- A dinamikus programozás több táblázatot tölt ki, ezek közül a legmagasabb szinten levő az F^5 táblázat. $F^5(j)$ megadja az első pozíciótól a j -edikig terjedő résstring minimális szabadenergiájú térszerkezetének a szabadenergiáját. Az 5-ös felső index arra utal, hogy az RNS szekvenciát az 5' irányból kezdjük olvasni. A dinamikus programozási rekurzió: $F^5(j) = \min\{F^5(j-1), \min_{2 \leq l \leq j-m-1} \{F^5(l-1) + C(l, j)\}\}$ ahol $C(l, j)$ megadja az l -től j -ig terjedő részstring azon térszerkezetei közül a minimális szabadenergiájúnak a szabadenergiáját, amelyben (l, j) bázispárt alkot.

- **Boltzman eloszlás**

- Egy S struktúra előfordulási valószínűsége Boltzman eloszlásban: $P(S) \sim e^{-\frac{\Delta G(S)}{RT}}$
- Az arányossági tényező reciproka a partíciófüggvény: $Z = \sum_S e^{-\frac{\Delta G(S)}{RT}}$
- Ha ki tudnánk számolni a partíciófüggvényt, akkor a minimális szabadenergiájú térszerkezet valószínűségét a Boltzmann eloszlásban pontosan meg tudnánk mondani. Felmerülhet a kérdés, hogy szükséges-e újra implementálni a dinamikus programozási rekurziót.
- A dinamikus programozásban az egyes esetek közötti kapcsolat ugyanaz, csak a számolás más. Ha szét tudnánk választani az esetek közötti kapcsolatokat a konkrét számolástól, akkor a bonyolult részt csak egyszer kellene leprogramozni, és csak a számolásokra kellene külön függvényeket írni. Ezt a fajta programozási eljárást hívják algebrai dinamikus programozásnak.

11 TÉTEL

- **NP, P**

- Egy döntési probléma NP-beli, ha létezik olyan nem-determinisztikus Turing gép, amely minden problémabeli feladatot meg tud oldani a feladat polinom függvényének idejében.
- Egy probléma NP-beli, ha létezik olyan tanu, amely polinom időben ellenőrizhető.
- Egy döntési probléma P-beli, ha létezik olyan determinisztikus Turing gép, amely minden problémabeli feladatot meg tud oldani a feladat polinom függvényének idejében.
- Mivel minden determinisztikus Turing gép egyben nem-determinisztikus Turing gép is, ezért minden P-beli probléma egyben NP-beli is.
- Egy A probléma polinomiálisan visszavezethető B problémára, ha minden A-beli feladat polinom időben megoldható B-beli feladatok megoldásainak segítségével.
- Egy A probléma NP-nehéz, ha minden NP-beli probléma polinomiálisan visszavezethető A-ra.
- Egy probléma NP-teljes, ha NP-beli és NP-nehéz.
- Létezik NP-teljes probléma. (pl Hamilton kör)

- **RP, BPP \$P, FP**

- Egy döntési probléma RP-beli, ha létezik olyan polinom időben futó random algoritmus, amelyre ha a helyes válasz nem, akkor az algoritmus is 1 valószínűséggel nem-et fog mondani, míg ha a válasz igen, akkor a helyes válasz több, mint 0.5 valószínűséggel igen.
- $P \subseteq RP \subseteq NP$, további sejtés: $RP = P = NP$
- Egy döntési probléma BPP-beli, ha létezik olyan random algoritmus, amelyik több, mint 0.5 valószínűséggel igen-t mond, ha a tényleges válasz igen és több, mint 0.5 valószínűséggel nem-et mond, ha tényleges válasz nem.
- $NP = BPP \Rightarrow NP = RP$
- A #P-beli leszámolási problémák az NP-beli leszámolási problémák tanuira kérdeznak rá.
- Egy #P-beli probléma FP-ben van, ha létezik olyan algoritmus, amely polinom időben kiszámolja a tanuk számát.
- Egy probléma #P-nehéz, ha minden #P-beli probléma polinom reducibilis rá. Egy probléma #P-teljes, ha #P-beli és #P-nehéz.
- Létezik #P-teljes probléma. (pl teljes párosítások száma)
- $FP = \#P \Rightarrow P = NP$

- **FPRAS, FRAUS, önhasonló**

- Egy #P-beli leszámolási probléma FPRAS-ban van, ha létezik rá olyan sztochasztikus algoritmus, amely minden problémabeli feladatra és $\delta > 0$ -ra megbecsüli a tanuk számát, úgy, hogy teljesüljön a $\mathbb{P}\left(f(1+\epsilon)^{-1} \leq \hat{f} \leq f(1+\epsilon)\right) \geq 1-\delta$ egyenlőtlenség, ahol f a tanuk tényleges száma, $\hat{f}$ a becslés, és az algoritmus futási ideje polinom függvénye mind a feladat méretének, mind $\frac{1}{\epsilon}$ -nak, mind $-\log(\delta)$ -nak.
- Ha bármely NP-teljes döntési probléma tanui számára létezik FPRAS algoritmus, akkor $RP = NP$.
- $\#P - teljes \cap FPRAS \neq \emptyset$ (pl teljes párosítások száma)

- Egy $\#P$ -beli probléma FPAUS-ban van, ha minden feladatára és $\delta > 0$ -ra létezik olyan algoritmus, amely egy random tanut ad meg a π eloszlásból, melyre teljesül, hogy $d_{TV}(\pi, U) \leq \delta$ ahol d_{TV} jelöli a variációs távolságot, U a tanuk egyenletes eloszlását, és az algoritmus futási ideje polinom függvénye mind feladat meretének, mind $-\log(\delta)$ -nak.
- Egy $\#P$ -beli probléma önhasonló leszámolási probléma, ha minden x feladatának minden w parciális megoldásához létezik egy x' feladat, amely leírása nem hosszabb x leírásánál, és a w parciális megoldások befejezései éppen x' megoldásai.
- Önhasonló leszámolási problémákra létezik FPAUS algoritmus akkor és csak akkor, ha létezik FPRAS algoritmus.
- Ha egy önhasonló leszámolási problémára létezik polinom faktorú determinisztikus approximáció, akkor létezik rá FPRAS is.

12 TÉTEL

• Elutasítás metódus

Adott egy π eloszlás, amelyre bármely x pontban $\pi(x)$ kiszámolható egy ismeretlen normalizációs konstans erejéig (azaz van olyan $f(x)$, hogy $\frac{f(x)}{\pi(x)}$ minden x re ugyan az), továbbá van egy p eloszlás, amelyből tudunk mintavételezni, egy g függvény, amely arányos p -vel és kiszámolható minden x pontban, és ismert egy c borítékoló konstans úgy, hogy minden x -re $f(x) \leq cg(x)$. Ekkor a metódus lépései:

- Vesszünk egy random x -et p -ből.
- Vesszünk egy random u -t az egyenletes eloszlásból a $[0, 1]$ intervallumon. Ha $ucg(x) \leq f(x)$ akkor elfogadjuk x -et, egyébként elutasítjuk, és újrakezdjük a procedúrát.

Az elutasítás metodussal generált random számok a π eloszlást követik.

• Metropolis-Hastings

Legyen adott egy irreducibilis, aperiodikus Markov lánc egy X halmazon $T(\cdot|\cdot)$ átmeneti valószínűségekkal, amelyre teljesül, hogy $\forall x, y : T(x|y) \neq 0 \Rightarrow T(y|x) \neq 0$, valamint minden x és y párra a $T(y|x)$ egy ismeretlen normalizációs konstans erejéig számolható. Legyen adott egy π eloszlás, amely sehol nem 0 X -en, és X bármely pontjára egy ismeretlen normalizációs konstans erejéig számolható. Ekkor a Metropolis-Hastings algoritmus a következő két lépésből áll:

- Ha az aktuális állapot x_t , vegyünk fel egy véletlen y -t a $T(\cdot|x_t)$ eloszlásból.
- Vegyünk fel egy véletlen u -t az egyenletes eloszlásból a $[0, 1]$ intervallumon. Legyen $x_{t+1} = y$, ha $u \leq \frac{\pi(y)T(x_t|y)}{\pi(x_t)T(y|x_t)}$, egyébként x_t .

A Metropolis-Hastings algoritmus által definiált Markov láncnak π globálisan stabilis stacionárius eloszlása.

• Markov lánc Monte Carlo

– Gibbs-féle mintavételezés

Akkor alkalmazható, ha minden pontra és i koordinátára lehetséges az adott i koordináta értékét mintavételezni, úgy, hogy a kérdéses értékek a többi koordinátával együtt a kívánt π eloszlást kövessék, felteve, hogy az i -edik koordináta kivételével az összes koordinátaérték rögzített. Ekkor a Gibbs-féle mintavételezés minden lépésben kiválaszt egy véletlen koordinátát tetszés szerinti, de az aktuális ponttól független eloszlásból, és a kiválasztott koordinátára egy új értéket választ, a $\pi(\cdot|x_{-i})$ eloszlást követve, ahol $x_{-i} = x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$.

- * Ha a Gibbs-fele mintavételezés irreducibilis Markov lánc, akkor π globálisan stabilis stacionárius eloszlása.

– Párhuzamos melegítés

A párhuzamos melegítés során egy adott állapottól önmagával vett k -szoros Descartes szorzatát vesszük, és ez lesz az új állapottól. Az új állapottól eloszlása koordinátánként definiáljuk, az egyes koordináták eloszlásai függetlenek egymástól, és minden koordináta egy Boltzmann eloszlás lesz. Az eredetileg elérni kívánt π eloszlásra úgy tekintünk, mint egy Boltzmann eloszlásra $T = 1$ hőmérsékleten, így minden eredeti x állapotra definiálni tudjuk az állapot szabadenergiáját: $\Delta G(x) = -\log(\pi(x))$.

- * Ha az egyes koordinátákra vett Metropolis-Hastings algoritmusok által definiált Markov láncok aperiodikusak és irreducibilisek, akkor a párhuzamos melegítéssel definiált Markov láncnak a koordinátánként vett Boltzmann eloszlások szorzata globálisan stabilis stacionárius eloszlása.
- * A párhuzamos melegítés akkor segítheti elő a konvergenciát, ha az eredeti állapottól kívánt π eloszlásának több lokális maximuma van, és a Markov lánc nem tud gyorsan keveredni a lokális maximumokat elválasztó kis valószínűségű állapotok miatt.

– **Metropolizált fontossági mintavételezés**

Minden fontossági mintavételezést fel tudunk használni egy Markov lánc Monte Carlohoz is, ha a fontossági mintavételezésre úgy tekintünk, mint egy nulladrendű Markov láncra (a következő állapot független az aktuális állapottól). Ha π a céeloszlás, p a fontossági eloszlás, akkor a Metropolis-Hastings hányados a fontossági súlyok hányadosa: $\frac{\pi(y)p(x)}{\pi(x)p(y)} = \frac{w(y)}{w(x)}$ ahol $w(\cdot) = \frac{\pi(\cdot)}{p(\cdot)}$.

* A Metropolizált fontossági mintavételezés Markov láncának második legnagyobb sajátértéke: $\rho = 1 - (\max_i \{w(x_i)\})^{-1}$, magyarul a spektrális rés reciproka a legnagyobb fontossági súly.

13 TÉTEL

• **Markov láncok**

- Egy Markov lánc gráfja egy irányított gráf, amelynek pontjai a Markov lánc lehetséges állapotai, és egy u csúsból akkor és csak akkor megy v -be él, ha az u -ból v -be való átmenet valószínűsége nem 0.
- Egy Markov lánc irreducibilis, ha a gráfjában bármely két pont között megy irányított út.
- Egy Markov lánc aperiodikus, ha a gráfjában az irányított körök hosszainak legnagyobb közös osztója 1.
- Ha egy Markov lánc irreducibilis, aperiodikus és reverzibilis, akkor a részletes egyensúlyban szereplő eloszlás globálisan stabilis lesz.

• **Konvergenciasebesség**

- Az abszolútértékben második legnagyobb sajátértéknek az 1-től való távolságát spektrális rés-nek hívják.
- Egy r állapotú Markov lánc második legnagyobb sajátérték modulusa: $\rho = \max_r \{\lambda_2, |\lambda_r|\}$
- A relaxációs idő: $\frac{1}{1-\lambda_2}$
- **Lusta markov lánc**
 - * Egy T átmeneti mátrixú Markov lánc lusta változata a $\frac{T+E}{2}$ átmeneti mátrixú Markov lánc, ahol E az identikus mátrix.
 - * Egy irreducibilis, reverzibilis Markov lánc lusta változata irreducibilis, aperiodikus és reverzibilis lesz, ugyanazzal a stacionárius eloszlással, ugyanazokkal a sajátvektorokkal, és a sajátértékek az eredeti sajátértékek és az 1 számtani átlaga lesznek.
 - * Lusta markov lánc minden sajátértéke > 0 , illetve teljesül, hogy $(1 - \rho')^{-1} = 2(1 - \lambda_2)^{-1}$
- Ha egy P-beli döntési problémához, amelyre a tanuk számának logaritmus polinomiális függvénye a feladat méretének, lehet egy olyan irreducibilis, reverzibilis Markov láncot készíteni, amely a tanuk egyenletes eloszlásához konvergál, a spektrális résének a reciproka polinomiálisan növekszik a feladat méretével, és egy lépés a Markov láncban a feladat méretének polinom idejében megtehető, akkor a P-beli problémához tartozó #P-beli feladatra létezik FPAUS.
- Egy markov lánc állapotterében egy S részhalmaz ergodikus folyama: $F(S) = \sum_{i \in S, j \in \bar{S}} \pi(i)T(j|i)$.
- Egy markov lánc konduktanciája: $\Phi = \min\{\frac{F(S)}{\pi(S)} | S \subset I, 0 < \pi(S) \leq 0.5\}$, ahol I az állapottér. Azaz a konduktancia a feltételes kifolyási sebesség minimuma az összes olyan részhalmazon, amelynek a valószínűsége legfeljebb 0.5.
- $1 - 2\Phi \leq \rho \leq 1 - \frac{\Phi^2}{2}$
- Poincaré koefficiens: Legyen adva egy Γ útvonalrendszer, amely a Markov lánc gráfjában minden (i, j) pár tartalmaz pontosan egy irányított utat i -ből j -be. Egy út Q -mértéket definiáljuk a kapacitások reciprokainak összegeként, azaz $|\gamma_{ij}|_Q = \sum_{e \in \gamma_{ij}} Q(e)^{-1}$, ahol $e = (x, y)$ él kapacitása $Q(e) = \pi(x)T(y|x)$. Ekkor Γ útvonalrendszer PK-ja: $\kappa = \max_e \sum_{\gamma_{ij} | e \in \gamma_{ij}} |\gamma_{ij}|_Q \pi(i)\pi(j)$.
- $\rho \leq 1 - \kappa^{-1}$