FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

BIOINFORMATIKA

Određivanje poravnanja parova sljedova korištenjem HMM

Autori
Tomislav Božurić
Martin Pisačić
Krešimir Topolovec

Zadatak
doc. dr. sc. Mirjana Domazet-Lošo

Siječanj, 2019

1 Opis algoritma i vizualizacija

U ovome radu korišten je modificirani Viterbijem algoritam pomoću kojeg korištnjem dinamičkog programiranja možemo pronaći najvjerojatniju sekvencu skrivenih stanja koja ujedno predstavlja optimalno poravnanje. Da bismo klasičan Viterbijev algoritam tranformirali u HMM, moramo napraviti nekoliko izmjena. Prvo moramo odrediti vjerojatnosti za emitiranje simbola iz stanja. Npr. stanje M (match) ima vjerojatnosnu distribuciju p_{ab} za emitaranje para simbola ab. Stanja X i Y imaju vjerojatnost emitiranja p_a simbola a umjesto praznine. Također potrebno je definirati vjerojatnosti prijelaza između stanja tako da je suma vjerojatnosti odlaska iz pojedinog stanja jednaka 1. Vjerojatnost prijelaza iz stanja M u stanje X i Y opisujemo oznakom δ, a vjerojatnost u ostanka u stanju X ili Y sa ϵ. Takva definicija ne definira kompletni model koji omogučava vjerojatnosnu distribuju po svim mogučim sekvecama. Za kompletiranje modela dodajemo početno i krajnje stanje *Poetak* i *kraj*. Definiramo vjerojatnost τ koja je jednaka za sve prijelaze iz stanaj M, X i Y u stanje End, te za stanje Begin u stanja M, X i Y.

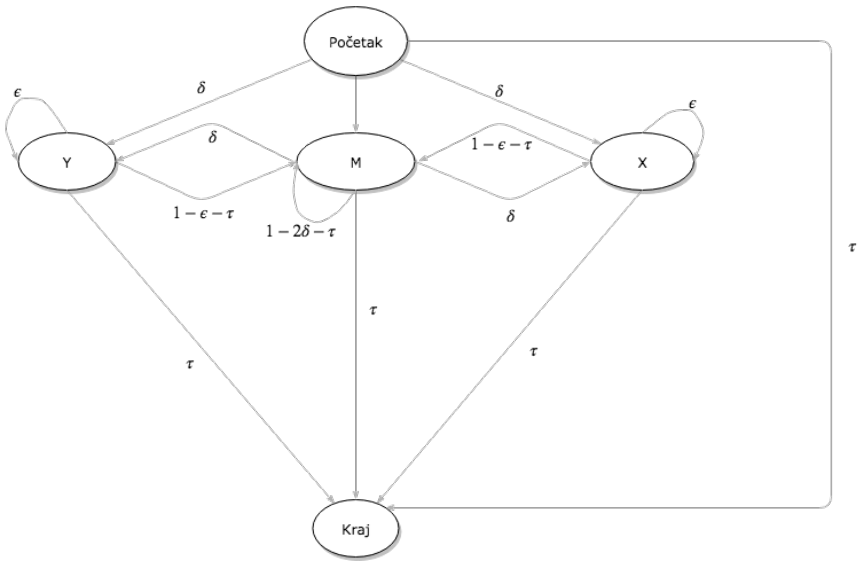

Figure 1: Model HMM-a korištenog za poravnanje parova sekvenci

Algoritam: Viterbijev algoritam za HMM [2]
Inicijalizacija:

$$v^M(0,0) = 1$$
$$v^\bullet(i,0) = v^\bullet(0,j) = 0$$

Korak:

za svaki $i = 1, ..., n$, $j = 1, ..., m$

$$v^M(i,j) = p_{x_i y_i} \max \begin{cases} (1-2\delta-\tau)v^M(i-1,j-1) \\ (1-\epsilon-\tau)v^X(i-1,j-1) \\ (1-\epsilon-\tau)v^Y(i-1,j-1) \end{cases} \quad (1)$$

$$v^X(i,j) = q_{x_i} \max \begin{cases} \delta v^M(i-1,j) \\ \epsilon v^X(i-1,j) \end{cases} \tag{2}$$

$$v^Y(i,j) = q_{y_j} \max \begin{cases} \delta v^M(i,j-1) \\ \epsilon v^Y(i,j-1) \end{cases} \tag{3}$$

Uvjet zaustavljanja: $v^E = \max(v^M(n,m), v^X(n,m), v^Y(n,m))$

Zbog činjenice da su vjerjoatnosti brojevi u intervalu $[0,1]$, gore navedeni algoritam nije upotrebiv za implementaciju na računalu zbog velikog broja množenja brojeva bliskih nuli, pa se u praksi koristi logaritamska inačica tog algoritma koja je navedena u nastavku.

Algoritam: optimalno poravnanje logaritamskih kvota [2]
Inicijalizacija:

$$V^M(0,0) = -2log(\eta), V^X(0,0) = V^Y(0,0) = -\infty$$
$$V^\bullet(i,-1) = V^\bullet(-1,j) = -\infty$$

Korak:

za svaki $i = 0,...,n$, $j = 0,...,m$ osim (0,0):

$$V^M(i,j) = s(x_i,y_j) + \max \begin{cases} V^M(i-1,j-1) \\ V^X(i-1,j-1) \\ V^Y(i-1,j-1) \end{cases} \tag{4}$$

$$V^X(i,j) = \max \begin{cases} V^M(i-1,j) - d \\ V^X(i-1,j) - e \end{cases} \tag{5}$$

$$V^Y(i,j) = \max \begin{cases} V^M(i,j-1) - d \\ V^Y(i,j-1) - e \end{cases} \tag{6}$$

Uvjet zaustavljanja: $V = \max(V^M(n,m), V^X(n,m) + c, V^Y(n,m) + c)$
Pri čemu su:

$s(a,b) = \log \frac{p_{ab}}{q_a q_b} + \log\left(\frac{1-2\delta-\tau}{(1-\eta)}\right)$
$d = -\log \frac{\delta(1-\epsilon-\tau)}{(1-\eta)(1-2\delta-\tau)}$
$e = -\log \frac{\epsilon}{1-\eta}$
$c = \log(1-2\delta-\tau) - \log(1-\epsilon-\tau)$

Table 1: Vizualizacija početnog stanja matrice V_M

indeks	-1	0	1	2	...	m
-1	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
0	$-\infty$	$-2log\eta$				
1	$-\infty$					
2	$-\infty$					
...	$-\infty$					
n	$-\infty$					

U tablici iznad možemo vidjet početno stanje matrice u kojoj pamtimo logaritamske kvote za stanje podudaranja. Iz modificiranog Viterbijevog algoritma koji koristiti logaritamske kvote iterativno korištenjem dinamičkog programiranja punimo tablicu i upisujemo novo izračunate vrijednosti. Analogno

radimo za tablice ispod, te kada dođemo do kraja iste te tablice koristimo za rekonstrukciju optimalnog poravnanja. Želimo li izračunati primjerice za vrijednosti $i = 1, j = 0$ izraz za V_M bi izgledao ovako:

$$V^M(1,0) = s(x_1, y_0) + \max \begin{cases} V^M(0,-1) = -\infty \\ V^X(0,-1) = -\infty \\ V^Y(0,-1) = -\infty \end{cases} \tag{7}$$

. Analogno popunjavamo preostali dio tablice. Ovdje možemo primjetiti kako za računanje vrijednosti u koraku i, j nam trebaju podatci iz koraka $i-1, j-1$, tako da je potrebno pararelno popunjavati sve tri prikazane matrice.

Table 2: Vizualizacija početnog stanja matrice V_X

indeks	-1	0	1	2	...	m
-1	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
0	$-\infty$	$-\infty$				
1	$-\infty$					
2	$-\infty$					
...	$-\infty$					
n	$-\infty$					

Table 3: Vizualizacija početnog stanja matrice V_Y

indeks	-1	0	1	2	...	m
-1	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
0	$-\infty$	$-\infty$				
1	$-\infty$					
2	$-\infty$					
...	$-\infty$					
n	$-\infty$					

Kada smo završili s forward dijelom algoritma, odnosno kada su nam sve strukture podatka popunjene spremni za pronalazak optimalnog poravnanja dvije sekvence. Kako nam je uvjet "zaustavljanja" $V = \max(V^M(n,m), V^X(n,m) + c, V^Y(n,m) + c)$, tražimo maksimum između navednih stanja. Zatim generiramo poravnanje te u povratku prema početku ponavljamo postupak tražeći maksimum između V_M, V_X, V_Y.Na idućem primjeru je ilustriran primjer pronalaska optimalnog poravnanja gdje se na poziciji (n,m) vrijednost računa kako je prethodno spomenuto.

Table 4: Vizualizacija pronalaska optimalnog poravnanja

indeks	0	1	2	3	...	m-1	m
0	$V^\bullet(0,0)$						
1		$V^\bullet(1,1)$					
2							
3							
...							
n-1				...		$V^\bullet(n-1,m-1)$	
n							$V^\bullet(n,m)$

2 Analiza točnosti, vremena izvođenja i utroška memorije

3 Testiranje

3.1 Testiranje na sintetskim podatcima

3.2 Testiranje na stvarnim podatcima

References

[1] Byung-Jun Yoon. *Hidden Markov Models and their Applications in Biological Sequence Analysis*. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2766791, US National Library of Medicine, National Institutes of Health, 2009.

[2] Jun Xie. *Pairwise alignment using HMM*. http://www.stat.purdue.edu/~junxie/topic4.pdf, Purdue University.