16/20

William Argus
A12802324
ECE 208

Homework 6 Writeup

Correct Recursive Formula:

Let sequence length be *l*, parent node be *u*, and children nodes be *v,* and *w.* The subtree likelihood matrices are given as follows:

$$L_u = \begin{array}{c} \\ A \\ G \\ C \\ T \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & \cdots & l \end{array} \\ \begin{bmatrix} L_u^1(A) & L_u^2(A) & & L_u^l(A) \\ L_u^1(G) & L_u^2(G) & & L_u^l(G) \\ L_u^1(C) & L_u^2(C) & & L_u^l(C) \\ L_u^1(T) & L_u^2(T) & & L_u^l(T) \end{bmatrix} \end{array}$$

$$L_v = \begin{array}{c} \\ A \\ G \\ C \\ T \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & \cdots & l \end{array} \\ \begin{bmatrix} L_v^1(A) & L_v^2(A) & & L_v^l(A) \\ L_v^1(G) & L_v^2(G) & & L_v^l(G) \\ L_v^1(C) & L_v^2(C) & & L_v^l(C) \\ L_v^1(T) & L_v^2(T) & & L_v^l(T) \end{bmatrix} \end{array}$$

$$L_w = \begin{array}{c} \\ A \\ G \\ C \\ T \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & \cdots & l \end{array} \\ \begin{bmatrix} L_w^1(A) & L_w^2(A) & & L_w^l(A) \\ L_w^1(G) & L_w^2(G) & & L_w^l(G) \\ L_w^1(C) & L_w^2(C) & & L_w^l(C) \\ L_w^1(T) & L_w^2(T) & & L_w^l(T) \end{bmatrix} \end{array}$$

Note that the goal of this recursion is to find the subtree likelihood matrix at the root node, which will in fact be the likelihood matrix of the entire tree, from which the likelihood of the

tree can be found. Hence the recursive formula starts at the root of the tree and recuses down into the tree until it reaches leaves. When the recursion is completed, it will give the root likelihood matrix.

$for\ i\ \in 1\ to\ l,\ \ for\ s\ \in (A,G,C,T)$:

$L_u^i(s) =$

$\left(\sum_{x\in A,G,C,T} P(\ D_v^i = x \middle| D_u^i = s, t_v, \theta\right) * L_v^i(x)) * \left(\sum_{x\in A,G,C,T} P(\ D_w^i = x \middle| D_u^i = s, t_w, \theta\right) * L_w^i(x))$

Where $P\left(D_v^i = x \middle| D_u^i = s, t_v, \theta\right) = e^{t_v R_{[s,x]}}$.

$t_v = branch\ length\ at\ node\ v$. R is defined below:

	A	G	C	T
A	$-(\alpha\pi_G+\beta\pi_C+\gamma\pi_T)$	$\alpha\pi_G$	$\beta\pi_C$	$\gamma\pi_T$
G	$\alpha\pi_A$	$-(\alpha\pi_A+\delta\pi_C+\varepsilon\pi_T)$	$\delta\pi_C$	$\varepsilon\pi_T$
C	$\beta\pi_A$	$\delta\pi_G$	$-(\beta\pi_A+\delta\pi_G+\eta\pi_T)$	$\eta\pi_T$
T	$\gamma\pi_A$	$\varepsilon\pi_G$	$\eta\pi_C$	$-(\gamma\pi_A+\varepsilon\pi_G+\eta\pi_C)$

Note that the matrix R is divided by the value d in order to normalize it.

$$d = \sum \pi_i v_i = \sum -\pi_i R_{i,i}$$

Note that once the likelihood matrix of the root node has been found, the following equation can be used to find the likelihood of the entire tree.

$$L(\theta) = \prod_{i\ \in 1\ to\ l} \sum_{x\in A,G,C,T} \pi_x L_{root}^i\ (x)$$

If the log-likelihood is desired, the following equation is used:

$$\log(L(\theta)) = \sum_{i \in 1 \text{ to } l} \log\left(\sum_{x \in A,G,C,T} \pi_x L^i_{root}(x)\right)$$

Correct Base Case:

The base case is when the current node, *u* is a leaf. Since it has no children, the formula above cannot be used on it, and furthermore it is not necessary to use the formula above on it because in this case the sequence at a leaf is known, so the likelihood matrix can be obtained directly from that sequence. Hence since the formula is not used on a leaf node and its likelihood matrix can be found as opposed to having to recurse deeper, this is the base case.

Describe how you compute L at the leaves -2 pts

Code Description and Asymptotic Running Time:

First a description of the function likelihood will be given and then its runtime will be analyzed. Here *n* is the number of nodes in the tree, and *l* is the length of the longest sequence.

First there is the function popLeafs, where when given a leaf node, initializes the likelihood matrix for the leaf node by setting, for each site (aka each column in the 4 by *l* matrix), the row that corresponds to the character given in seqeuce to 1 and the other 3 rows to zero. It does this iterating over all sites in the sequence.

Next there's a function, updateMatrix, that takes an internal node and fills in its likelihood matrix based on the recursive formula given above. To do so it must iterate through all sites,

and for each site, all 4 letters (A, G, C, T), in order to fill in the likelihood matrix of the parent node passed in.

Following that, there's definitions of important variables, and then the main loop in the likelihood function. This loops through all nodes in the tree, sending the leaf nodes to the function popleafs and the internal nodes to the function updateMatrix in order to get the likelihood matrix of all the nodes in the tree, including the root node.

Finally there's a loop that loops through 0 to *l*, and for the root likelihood matrix, uses the equation for $log\ (L(\theta))$, defined above, to find the log likelihood score of the tree.

With regards to runtime, the main loop goes through all *n* nodes. Each time it loops it either sends the current node to the function *popleafs* (if it is a leaf) or the function *updateMatrix* (if it is not a leaf). Within the function *popleafs*, the loop runs *l* times. Within the function *updateMatrix*, there is a nested for-loop, with the outer loop going through *l* times, the inner loop going through 4 times. The runtime of this main loop is then written as *O(nl)*, since the linear 4 in the *updateMatrix* runtime is not considered in big-O notation, making that function and *popleafs* have runtime *l*, with both inside a for loop that runs *n* times.

The final loop runs *l* times in order to find the likelihood of the tree, making its runtime *O(l)*.

Combined, this results in a runtime for the function of *O(nl) + O(l)*.

Accordingly, **the asymptotic runtime of the function likelihood is *O(nl),*** where *n* is the number of nodes in the tree and *l* is the length of the longest sequence.

O(nls^2) where s=4 for DNA and s=20 for amino acids
-2 pts