CS313 Exercise 10 Cover Page

Name(s): ________________________________

In the *TIME* column, please estimate the time you spent on the parts of this exercise. Please try to be as accurate as possible; this information will help us to design future exercises.

PART	**TIME**	**SCORE**
Exercise		

Task 1: Computing the Optimal Annotation of a Sequence Using a HMM

Suppose we have a HMM with the following model parameters:

$b_1(A) = 0.28$
$b_1(E) = 0.04$
$b_1(H) = 0.04$
$b_1(I) = 0.28$
$b_1(R) = 0.04$
$b_1(S) = 0.28$
$b_1(V) = 0.04$

$b_2(VER) = 0.90$
$b_2(AVE) = 0.10$
$b_3(ER) = 0.50$
$b_3(AV) = 0.50$

$a_{11} = 0.40$
$a_{12} = 0.59$
$a_{13} = 0.01$
$a_{21} = 0.27$
$a_{22} = 0.03$
$a_{23} = 0.70$

$a_{31} = 0.48$
$a_{32} = 0.04$
$a_{33} = 0.48$

Using the Viterbi algorithm for the observation sequence **HAVERAVERIVERS**, complete the two tables (dynamic programming and backtracking) below to compute both (a) the natural logarithm of the probability of the optimal annotation of the observation sequence and (b) the optimal annotation of the observation sequence. You should assume the optimal state sequence begins with state #1.

Dynamic Programming Table

-3.2	**-5.4**	**-9.5**	**-12.5**	**-10.6**	**-8.6**	**-12.8**	**-11.0**						
-∞	**-∞**	**-∞**	**-6.1**	**-6.0**	**-∞**	**-∞**	**-11.9**						
-∞	**-∞**	**-8.5**	**-∞**	**-9.9**	**-∞**	**-7.1**	**-∞**						

Backtracking Table

-1	**1**	**1**	**3**	**2**	**2**	**1**	**3**						
-1	**-1**	**-1**	**1**	**1**	**1**	**2**	**2**						
-1	**-1**	**1**	**1**	**3**	**2**	**2**	**1**						

What is the natural logarithm of the probability of the optimal annotation of the observation sequence **HAVERAVERIVERS**?

What is the optimal annotation of the observation sequence **HAVERAVERIVERS**?

Task 2: Computing the Optimal Annotation of a Sequence Using a GMM

Suppose we have a general Markov model (GMM) with the following model parameters:

$b_1(H) = 0.9$	$a_{11} = 0.90$	$a_{31} = 0.30$	$c_1(1) = 1.0$	$c_3(1) = 0.1$
$b_1(T) = 0.1$	$a_{12} = 0.05$	$a_{32} = 0.30$		$c_3(2) = 0.2$
$b_2(H) = 0.5$	$a_{13} = 0.05$	$a_{33} = 0.40$	$c_2(2) = 0.5$	$c_3(3) = 0.4$
$b_2(T) = 0.5$	$a_{21} = 0.02$		$c_2(3) = 0.5$	$c_3(4) = 0.2$
$b_3(H) = 0.2$	$a_{22} = 0.70$			$c_3(5) = 0.1$
$b_3(T) = 0.8$	$a_{23} = 0.28$			

Using the Viterbi algorithm for the observation sequence **HTTT**, complete the dynamic programming table below to compute the natural logarithm of the probability of the optimal annotation of the observation sequence. You should assume the optimal state sequence begins with state #1.

Dynamic Programming Table

-0.1	**-2.5**	**-4.9**	
-∞	**-∞**	**-5.2**	
-∞	**-5.6**	**-5.2**	

What is the natural logarithm of the probability of the optimal annotation of the observation sequence **HTTT**?

Task 3: Runtime of Viterbi Algorithm

Suppose we have a HMM with N states and an observation sequence of length T.

Consider the scenario when, for each state, the characters emitted by the state all have the same length. For example, all characters emitted by state Q^1 have the same length $|b_1|$, all characters emitted by state Q^2 have the same length $|b_2|$, etc. What is the running time in big-Oh notation of the Viterbi dynamic programming algorithm in this scenario?

Consider the scenario when, for each state, the characters emitted by the state can have any (differing) length. For example, state Q^1 might emit different characters ranging in length from 1 to T, state Q^2 might emit different characters ranging in length from 1 to T, etc. What is the running time in big-Oh notation of the Viterbi dynamic programming algorithm in this scenario?

Task 4: Runtime of Forward Algorithm

Suppose we have a HMM with *N* states and an observation sequence of length *T*.

The Viterbi algorithm enables computation of the probability of the *optimal* state sequence corresponding to the observation sequence. However, there are other values we may be interested in computing besides the probability of the *optimal* state sequence.

The *evaluation* problem is the problem of computing the probability that an observation sequence was produced by the model. Said another way, we may wish to score how well a given model matches a given observation sequence. We can score how well a model matches an observation sequence using the Forward algorithm. Whereas the Viterbi algorithm computes the probability of the *optimal* state sequence, the Forward algorithm computes the probability of *all* state sequences. The Forward algorithm computes the probability that *all* state sequences (in sum) generated the observation sequence, rather than identifying the *single* state sequence that was most likely to generate the observation sequence. Below is a recurrence describing the Forward algorithm (notice the difference between the Viterbi algorithm and the Forward algorithm is that the "max" in the Viterbi algorithm has been replaced with a summation in the Forward algorithm):

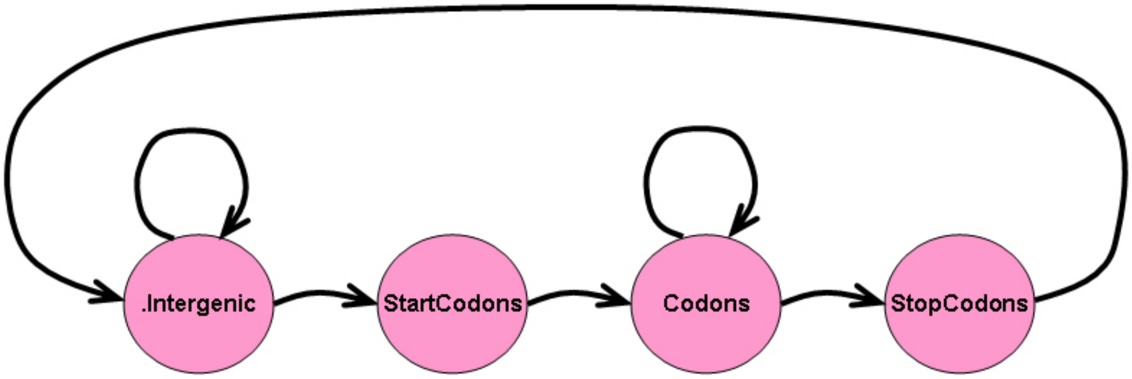
$$\alpha_t(j) = \begin{cases} b_1(O_1) & \text{if } t = 1, j = 1 \\ 0.0 & \text{if } t = 1, j \neq 1 \\ \sum\limits_{1 \leq i \leq N} \left(\alpha_{t-|b_j|}(i) * a_{ij} \right) * b_j(O_{t-|b_j|+1} \ldots O_t) & \text{if } 2 \leq t \leq T \end{cases}$$

The above recurrence represents the scenario when, for each state, the characters emitted by the state all have the same length. For example, all characters emitted by state Q^1 have the same length $|b_1|$, all characters emitted by state Q^2 have the same length $|b_2|$, etc. What is the running time in big-Oh notation of the Forward dynamic programming algorithm in this scenario?

Although the recurrence is not shown, consider the scenario when, for each state, the characters emitted by the state can have any (differing) length. For example, state Q^1 might emit different characters ranging in length from 1 to *T*, state Q^2 might emit different characters ranging in length from 1 to *T*, etc. What is the running time in big-Oh notation of the Forward dynamic programming algorithm in this scenario?

Task 5: Enhancing a Gene Finding HMM to Include Kozak Sequences

Consider a gene-finding HMM with the following architecture



specified by the following parameters:

$b_{\text{.Intergenic}}(A) = 0.28$
$b_{\text{.Intergenic}}(C) = 0.22$
$b_{\text{.Intergenic}}(G) = 0.22$
$b_{\text{.Intergenic}}(T) = 0.28$
$b_{\text{StartCodons}}(ATG) = 0.90$
$b_{\text{StartCodons}}(TTG) = 0.02$
$b_{\text{StartCodons}}(GTG) = 0.08$
$b_{\text{StopCodons}}(TAA) = 0.65$
$b_{\text{StopCodons}}(TGA) = 0.28$
$b_{\text{StopCodons}}(TAG) = 0.07$

$b_{\text{Codons}}(AAA) = 0.03$
$b_{\text{Codons}}(AAC) = 0.02$
$b_{\text{Codons}}(AAG) = 0.01$
$b_{\text{Codons}}(AAT) = 0.02$
$b_{\text{Codons}}(ACA) = 0.01$
$b_{\text{Codons}}(ACC) = 0.02$
...
$b_{\text{Codons}}(TTC) = 0.02$
$b_{\text{Codons}}(TTG) = 0.01$
$b_{\text{Codons}}(TTT) = 0.02$

$a_{\text{.Intergenic .Intergenic}} = 0.994$
$a_{\text{.Intergenic StartCodons}} = 0.006$
$a_{\text{.Intergenic Codons}} = 0.00$
$a_{\text{.Intergenic StopCodons}} = 0.00$
$a_{\text{StartCodons .Intergenic}} = 0.00$
$a_{\text{StartCodons StartCodons}} = 0.00$
$a_{\text{StartCodons Codons}} = 1.00$
$a_{\text{StartCodons StopCodons}} = 0.00$

$a_{\text{Codons .Intergenic}} = 0.00$
$a_{\text{Codons StartCodons}} = 0.00$
$a_{\text{Codons Codons}} = 0.997$
$a_{\text{Codons StopCodons}} = 0.003$
$a_{\text{StopCodons .Intergenic}} = 1.00$
$a_{\text{StopCodons StartCodons}} = 0.00$
$a_{\text{StopCodons Codons}} = 0.00$
$a_{\text{StopCodons StopCodons}} = 0.00$

In this task, your goal is to enhance the above HMM so that it models a form of Kozak sequences. Suppose that in a genome of interest, 90% of known genes have a Kozak-like sequence immediately upstream of their start codons and 10% of genes show no evidence of a Kozak sequence. For the 90% of known genes with a Kozak-like sequence immediately upstream of their start codons, 60% have the 4-mer `CACC` immediately preceding their start codon and 40% have the 4-mer `CGCC` immediately preceding their start codon.

Describe any necessary modifications to the HMM that would allow it to model Kozak-like sequences as described above. Modifications may relate to the number of states, the emission probabilities, or the transition probabilities. If the HMM architecture changes, please draw the new architecture.