Bioinformatics Practice Final Exam Problems

1 A Recipe for Inference

From the beginning of the course we have emphasized a consistent, systematic approach to solving inference problems. Based on Bayes Law, list each of the basic questions that we must answer to solve any inference problem, and briefly explain its meaning.

2 A Simple Protein Coding Model

A gene sequence (a string of A, T, G, C nucleotide letters) encodes a *protein* sequence in the form of a series of *codons*, each three nucleotide letters long. Each three-letter codon encodes one letter in the protein sequence (which has a twenty letter "alphabet").

Let's design a simple Markov model of a codon sequence, based on the following rules:

- A codon sequence *must* begin with the codon ATG.
- The codons TAA, TGA and TAG are *STOP codons* that terminate the protein.
- Between the start and stop codons we can have any number of regular codons.
- For simplicity, let's assume only two kinds of regular codons are allowed: GTx (where x means any of A, T, G or C), or TCx.

Draw a Markov chain representing this model, assuming:

- Each node emits one nucleotide letter at a time;
- Label each node with the nucleotide letter(s) it can emit;
- Show the direction of connectivity of nodes by drawing arrows;
- Include explicit "start" and "end" nodes indicating where we can enter or exit the model.
- No need to indicate any transition probabilities (edge weights) on this model.

3 Hidden Markov Models

1. Define a Markov chain.
2. Define a hidden markov model, explaining each of its basic elements.
3. Derive an expression for the posterior probability of a specific state s_i given the observations, i.e. $p(\theta_t = s_i|\vec{O}_1^n)$.
4. Explain how this can be calculated efficiently using the forward-backward algorithm (include the specific recurrence relations for performing this calculation).
5. Analyze the computational complexity of the forward-backward algorithm.

4 HMM Profiles

1. Assuming that you are given an alignment of a set of related sequences representing a protein homology family, briefly outline a procedure for aligning a new sequence $\vec{X}$ to this family, taking into account the details of its position-specific letter probabilities (e.g. at a crucial "catalytic" site, only one specific letter might be observed).

   (4)

2. Draw a fragment of the HMM structure needed to allow both insertion and deletion during the emission of the new sequence $\vec{X}$ relative to the family model.

   (4)

3. Assuming that you are given a series of such HMMs $F_1, F_2, ...F_N$ representing different protein homology families, and the prior probability $p(F_k)$ that a randomly chosen protein sequence will be a member of each family, derive an expression for the posterior probability that sequence $\vec{X}$ is a member of family F_k, i.e. $p(F_k|\vec{X})$. Explain how you would actually compute this.

   (4)

4. Assuming that $\vec{X}$ is from a specific family HMM model, use Bayes Law to derive an expression for the posterior probability that sequence letter X_t was actually emitted from HMM state $\Theta_t = s_i$ from this model, i.e. $p(\Theta_t = s_i|\vec{X})$. Use an information graph structure diagram to justify your choice of how to split up the total observation likelihood.

   (4)

5. Derive recursion rules for actually calculating the probability component(s) necessary to compute the posterior probability $p(\Theta_t = s_i|\vec{X})$.

   (4)

5 Poly-pyrimidine tracts

Consider the following 2-state model for detecting "poly-pyrimidine tracts" (regions with a high density of C or T):

- State α: emits one of the four nucleotides (A, T, G, C) with equal probability (1/4 each).
- State β: preferentially emits C or T (with probability 1/3 each), or A or G with probability 1/6 each.
- The transition probabilities are $p(\beta|\alpha) = 1/5$, $p(\alpha|\beta) = 1/10$.

1. Calculate the likelihood $p(CTA|\alpha\alpha\alpha)$
2. Calculate the joint probability $p(CTA, \beta\beta\beta)$
3. Calculate the odds ratio $\frac{p(\beta\beta\beta|CTA)}{p(\alpha\alpha\alpha|CTA)}$
4. Say you are given a sequence O and want to calculate the posterior probability that the HMM was in state β for both letter i and $i + 1$, i.e. $p(S_i = \beta, S_{i+1} = \beta|\vec{O}^n)$. Derive an expression for calculating this posterior probability; express your answer in terms of the standard forward-backward values.

6 Affine Gap Alignment

1. Briefly state the simple gap scoring model and affine gap scoring model.
2. Describe exactly how to change the dynamic programming alignment algorithm from the simple gap model to the affine gap model.
3. Describe a basic HMM model of pairwise sequence alignment, and relate each of its components to the specific details of your answer in (b) for how to implement affine gap alignment by dynamic programming.

	A	B	C	D	E	F
B	3					
C	7.5	6.5				
D	6.5	5.5	3			
E	4	3	4.5	3.5		
F	3.5	2.5	5	4	1.5	

7 Simple Sequence Assembly

Genome sequencing produces many short sequences that represent fragments of the complete genome sequence. These short sequences must be *assembled* into a continuous sequence representing the complete genome sequence, by aligning sequences that have matching, overlapping ends.

Outline an algorithm to solve this problem for a pair of sequences, i.e. to find whether the head of one sequence matches the tail of the other sequence (with possible mismatches due to sequencing error).

8 Local vs. Global Alignment

1. Briefly define the *local* vs. *global* alignment problems.
2. Briefly describe the *local* vs. *global* alignment algorithms.
3. Is one of these alignment problems more general than the other? i.e. are the path constraints of one a subcase of the other?
4. Based on your answer to (c), is there any possible way to make one of these alignment algorithms actually perform the other kind of alignment? If so, briefly explain how. If not, explain why not.

9 Ultrametric Test?

You have been given a set of pairwise distance metrics for a set of sequences A - F (above). Do they appear suitable for re-constructing the phylogenetic tree using the UPGMA (hierarchical clustering) algorithm? Explain.

10 Tree Construction

1. Explain the neighbor joining principle and how it can be used to reconstruct a phylogenetic tree.
2. For a set of additive distances, how can we identify a pair of nodes that are neighbors in the tree?
3. Outline the neighbor joining tree construction algorithm in pseudocode.
4. Analyze the computational complexity of the algorithm.
5. How would you change the algorithm to build a tree from a set of ultrametric distances?

11 Tree Construction

1. Explain the difference between a rooted vs. unrooted tree, and derive the number of rooted trees associated with a given unrooted tree with n leaf nodes.

   (4)

2. Explain the assumptions of an "ultrametric distance" metric, and outline a test for evaluating whether the distances D_{ij} between a set of modern sequences i, j obey these assumptions.

   (4)

3. Derive an inductive rule for constructing the correct phylogenetic tree given a set of D_{ij} ultrametric distances. Provide a brief proof that this rule guarantees correctly connecting pairs of nodes that are adjacent in the tree.

   (4)

4. Explain how an additive distance metric differs from an ultrametric distance metric, and give an example of when it is more appropriate.

   (3)

5. Explain how the tree construction algorithm in (c) must be modified to work with an additive distance metric.

   (4)

12 Ancestral State Inference

1. Consider the following tree of three species A, B, C. Assuming that you know the states of a given character (e.g. nucleotide sequence) for each of these three species, explain the intuitive principle(s) for infering the state of the most recent common ancestor of A, B.

   (4)

2. Derive the recursion rule for infering the most likely ancestral states using the Viterbi algorithm, given a tree topology, and the observed characters X_u of leaf nodes u of the tree.

   (4)

3. Explain the detailed traceback rule for infering the most likely ancestral states using the Viterbi algorithm.

   (2)

4. Explain briefly how your Viterbi algorithm satisfies the intuitive principle(s) you stated in part (a).

   (2)