BTRY 4840 / 6840 / CS 4775
Computational Genetics and Genomics

Problem Set 3

Problem 1: Hidden Markov Models [50 pts]

Consider a two-state hidden Markov model for identifying regions of high G+C content, as shown in Figure 1. (This could be considered a crude gene-predictor, since genes tend to have higher G+C content than the surrounding noncoding DNA.) The state-transition parameters of this model are defined by one parameter: μ, the probability of switching states. The probabilities of the self-transitions are $1-\mu$, as needed for proper conditional probability distributions, and the probability of beginning with each state is simply one half. The emission probabilities for each base $x_i \in \{A, C, G, T\}$ given state $z_i \in \{h, l\}$ are:

$$P(x_i|z_i = h) = \begin{cases} 0.13 & x_i = A \\ 0.37 & x_i = C \\ 0.37 & x_i = G \\ 0.13 & x_i = T \end{cases} \qquad P(x_i|z_i = l) = \begin{cases} 0.32 & x_i = A \\ 0.18 & x_i = C \\ 0.18 & x_i = G \\ 0.32 & x_i = T \end{cases}$$

We will use this HMM to analyze a DNA sequence $\mathbf{x}$ of length 1,000, which can be downloaded from the course website. In parts (a) and (b) we will use a fixed value of $\mu = 0.01$.

(a) [20 pts] Implement the Viterbi algorithm in Python or R and use it to parse the sequence $\mathbf{x}$ into G+C-rich and -poor regions. Have your program output a list of nonoverlapping intervals (u, v) for the predicted G+C-rich regions, where each u indicates the first base of a G+C-rich interval and each v indicates the last base ($1 \leq u \leq v \leq 1000$). Turn in this list as a compact description of the Viterbi path. You should implement the algorithm using log probabilities, as discussed in lecture.

(b) [25 pts] Implement the forward and backward algorithms and compute the marginal posterior probability of state h at every position in the sequence. Graph these probabilities versus sequence position, with position i on the horizontal axis and $P(z_i = h|\mathbf{x})$ on the vertical axis. Indicate the intervals predicted by the Viterbi algorithm on this graph. You should again use logs of probabilities in your implementation, but in this case you'll need to address the "log of sums" problem as discussed in lecture. Be sure to leave μ as a parameter in your program.

Note: compare $P(\mathbf{x})$ as computed by the forward and backward algorithms to check for correctness. The two quantities may differ slightly because of numerical error but should be close. When computing the posterior probabilities for each position i, first compute $P(\mathbf{x})$ as $P(\mathbf{x}) = \sum_k f_k(i)b_k(i)$ and check that this value is close to $P(\mathbf{x})$ as computed by the termination steps of the forward and backward algorithms. Then use *this local estimate of $P(\mathbf{x})$ as the denominator in computing the posterior probabilities for position i to ensure that they sum to one.*

(c) [5 pts] We have discussed parameter estimation (learning) for HMMs, but to begin, examine the likelihood function of the HMM by trial and error. In particular, consider the (log) likelihood as a function of the parameter μ, leaving all other parameters fixed. Compute $\ln P(\mathbf{x}|\mu)$ for ten to twenty values of μ ranging from its minimum of 0 to its maximum of 1, simply by running the forward (or backward) algorithm separately for each value of μ. (You may want to make the spacing denser near 0.) Now plot the log likelihood as a function of μ. What is the approximate MLE of μ? How strongly peaked is the likelihood function? What do you think would happen to the Viterbi path or the posterior probabilities if you used a much smaller or a much larger value of μ? (You can try it and see!)

Problem 2: Supervised Learning [20 pts]

As it happens, the sequence that you have analyzed was *generated* by an HMM, so the "true" state path is known in this case. It can be summarized by the set of G+C-rich intervals {(66,417), (468,509), (527,728), (946,1000)}, where an interval (i, j) implies that the state variable $z_k = h$ for $i \leq k \leq j$, and $z_k = l$ for all k not in any listed interval. (Indexing starts with one—i.e., the state path is given by the variables $z_1, \ldots, z_{1000}$.) We will now use this information to estimate the free parameters of the model.

To reduce parameters, assume strand symmetry—i.e., assume the numbers of Gs and Cs are roughly equal, and the number of As and Ts are roughly equal, and simply describe the distribution over bases given state $j \in \{h, l\}$ by a single parameter θ_j, which can be interpreted as the G+C content ($0 \leq \theta_j \leq 1$). In other words, each emission is a Bernoulli trial, with "success" being a G or a C (we don't care which) and "failure" being an A or a T. Thus, there are two emission parameters to estimate, θ_h and θ_l. In addition, there is one transition parameter, μ.

(a) [10 pts] Derive an expression for the complete log likelihood of the model in terms of these three parameters (i.e., $\log P(\mathbf{x}, \mathbf{z}|\mu, \theta_h, \theta_l)$). Express this quantity in terms of six types of counts: the number of transitions between states (c_b), the number of self transitions (c_s), the number of G/C bases emitted from the h state (d_{hG}), the number of A/T bases emitted from the h state (d_{hA}), the number of G/C bases emitted from the l state (d_{lG}), and the number of A/T bases emitted from the l state (d_{lA}). These "marginal counts" are sufficient statistics for the model. (Note that for this relatively simple model, we can write down the log likelihood analytically.)

(b) [10 pts] Estimate the parameters by maximum likelihood from the data above. Show that the problems of estimating the emission parameters (θ_l and θ_h) and the transition parameter (μ) can be decomposed.

Problem 3: Sequence Generation and Sampling State Paths [30 pts]

HMMs are *generative* models which define $P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})$ and provide a straightforward methodology for generating random samples of $\mathbf{x}$ and $\mathbf{z}$: a sequence of observations and the corresponding hidden states for each position. This contrasts with *discriminative* models which provide probabilities for unobserved variables conditional on observed data (e.g., $P(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})$) but do not specify the joint probability of $\mathbf{x}$ and $\mathbf{z}$. The ability to simulate both observed data and hidden states enables several applications including comparison of data generated from a model with real data. Besides generating data, one can use HMMs to sample hidden states from $P(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})$, that is, the posterior distribution of $\mathbf{z}$ for the observed data $\mathbf{x}$.

(a) [15 pts] Implement random sequence generation from the HMM specified in problem 1. Produce 50 pairs of $\mathbf{x}$ and $\mathbf{z}$ of length 1,000 along with their log likelihoods. How variable are the log likelihoods? (Note that $\ln 10 = 2.3$, so a difference of only 2.3 corresponds to ten-fold difference in the [normal space] likelihood of the sequence.) Calculate the number of state transitions in each sample and calculate the mean and variance of this number across all the observations. The distribution of the number of state transitions should follow a binomial model with a mean equal to the expected number of transitions out of 999 possible positions. Compare the mean and variance you observe to that expected under a binomial model. Are they very different? If so, why might this happen?

(b) [15 pts] Using the sequence data and model specification from problem 1, implement random sampling of state paths from the posterior $P(\mathbf{z}|\mathbf{x}, \mu)$. Sample 50 state paths while calculating the log likelihood of the sampled path. How do each of these log likelihoods compare to the log likelihood of the Viterbi path you calculated in problem 1(a)? How different are the paths in terms of their locations of high and low G+C content regions? Plot counts of the numbers of sequences that denote each position (from 1 to 1,000) as high G+C content (use the horizontal axis for sequence position and the vertical axis to indicate counts). How does your plot compare to the posterior state probabilities plot from problem 1(b)?

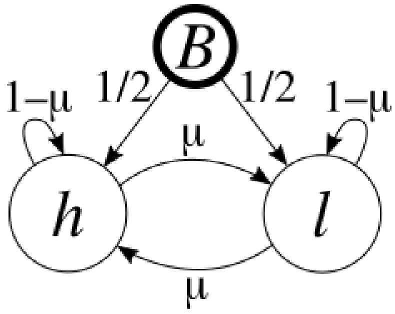

Figure 1: The HMM with states for high (h) and low (l) G+C content, and one parameter (μ) that defines state transitions. B denotes the "begin" state.