Bi/BE/CS 183 2021-2022
Instructor: Lior Pachter
TAs: Tara Chari, Meichen Fang, Zitong (Jerry) Wang

Problem Set 8

Submit your solutions as a single PDF file via Canvas by **8am Tuesday March 7th**.

- If writing up problems by hand, please use a pen and not a pencil, as it is difficult to read scanned submission of pencil work. Typed solutions are preferred.
- For problems that require coding, Colab notebooks will be provided. Please copy and save the shared notebook and edit your own copy, which you should then submit by including a clickable link in your submitted homework. Prior to submission make sure that you code runs from beginning to end without any error reports.

A hidden Markov model (HMM) has n hidden states and m observed states. At any time t, the random variable X_t is the hidden state at time t and the random variable Y_t is the observation at time t. There is a $n \times n$ transition matrix $S = \{S_{ij}\}$, where $S_{ij} = \mathbb{P}(X_t = j | X_{t-1} = i)$ is the probability of transitioning from hidden state i to hidden state j. There is a $n \times m$ matrix $T = \{T_{ij}\}$, where $T_{ij} = \mathbb{P}(Y_t = j | X_t = i)$ is the probability of observing state j while in hidden state i.

Problem 1 (60 points)

Consider the HMM with $n = 2$, $m = 4$ (corresponding to A,C,T,G), and an equal of probability of starting in each of the two hidden states. Suppose that

$$S = \begin{pmatrix} 0.8 & 0.2 \\ 0.05 & 0.95 \end{pmatrix} \tag{1}$$

and

$$T = \begin{pmatrix} 0.2 & 0.5 & 0.1 & 0.2 \\ 0.1 & 0.25 & 0.25 & 0.4 \end{pmatrix}. \tag{2}$$

(a) (10 points) Give all sequences of hidden states with non-zero probability of generating the sequence ACG (please provide link to code if not done by hand).

A: All hidden states have non-zero initial probability and all four transitions between hidden states have non-zero probability, this mean that all sequences of hidden states are possible. For either of the two hidden states, the transmission probability for each of the four letters ACTG have non-zero probability. Together, this implies that all $2^3 = 8$ triplets of hidden states have non-zero probability of generating the sequence ACG, which consists of 000, 001, 010, 011, 100, 101, 110, 111.

(b) (20 points) Find the most probable sequence of hidden state to have produced the sequence ACG (please provide link to code if not done by hand)

A: The hidden states can only take $\Sigma = \{1, 2\}$ corresponding to $n = 2$. Denote the probability of observing the sequence ACG with hidden state ijk by p_{ijk}. Then

$$p_{ijk} = \sum_{ijk \in \Sigma} 0.5 \times S_{ij} S_{jk} T_{iA} T_{jC} T_{kG}$$

Compute it either by hand or by code, we can find that the most probable sequence of hidden state is 111.

(c) (20 points) Compute p_{ACG}, which is the probability of observing the sequence ACG given all possible paths of hidden states (please provide link to code if not done by hand).

A: Summing over all possible sequences of hidden state, we have $p_{ACG} = 0.0155$.

(d) (5 points) Compute p^*_{ACG}, the probability of observing the sequence ACG via the most probable sequence of hidden states.

A: Using our solution from part b), we can find that the the probability of observing the sequence ACG via the most probable sequence of hidden states, being 111, is $p^*_{ACG} = p_{111} = 0.0064$.

(e) (5 points) For a particular observed DNA sequence, one might consider approximating the probability of observing the sequence by the probability of observing the sequence via the most probable sequence of hidden states. In the case of the sequence ACG, would this be a good approximation?

A: In the case of the sequence ACG, $p^*_{ACG} = 0.0064$ and $p_{ACG} = 0.0155$. p_{ACG} is more than double that of p^*_{ACG} and so the latter is not a good approximation of the former.

Problem 2 (40 points)

In this problem you will implement the Needleman-Wunsch algorithm for global alignment of a pair of sequences. Your program will read in a FASTA file containing a pair of DNA sequences, and then run the Needleman-Wunsch algorithm to find their optimal alignment given parameters for matching, mismatching, and unaligned base-pairs. The algorithm constructs an optimal global alignment by dynamic programming.

The solution notebook is here.