CS234 Homework 3 solution

Problem 1: Computing products of probabilities (50 points)

A typical subroutine in several of the methods discussed in class require the computation of the product of the probabilities of the symbols in a string. Assume that we are dealing with a Bernoulli source, and therefore that symbols are independently generated. Suppose we are given a text $x$ of size $n$ over the alphabet $\Sigma = \{A, T, C, G\}$ and that we know $p_A, p_T, p_C, p_G$ in advance.

Q: Design an algorithm to compute 

$$p(y) = \prod_{i=1}^{\vert y \vert} p_{y[i]}$$

in $O(1)$ for any of the substrings $y$ of the text $x$. A substring $y$ is identified by a pair $(i, j)$ where $i$ is the start of the substring, and $j$ is the end of the substring. You are allowed to spend $O(n)$ time in a preprocessing phase, just once. After that, your algorithm should be able to return $p(y)$ in constant time for any substring $y$ in the text.

ANSWER: In the preprocessing phase, we compute $P_i$’s for all the $i$ prefixes of text $x$, i.e. $P_i = \prod_{k=1}^{i} p_{x[k]}$. The details of the algorithm is as follows:

Preprocessing $(x)$:
- **Input:** String $x$ of length $n$
- **Output:** $P_i$’s, the probabilities of all the prefixes of $x$
- $P_0 \leftarrow 1$;
- **For** $(i \leftarrow 1$ to $n)$
  - $P_i \leftarrow P_{i-1} * p_{x[i]}$

Clearly, the above algorithm runs in $O(n)$ time.

To calculate the probability for any substring $y = x[i:j] = x[i] x[i+1] ... x[j]$, we observe that $P(y) = \prod_{k=i}^{j} p_{x[k]} = \frac{P_j}{P_{i-1}}$, which can be done in $O(1)$ time.

Q: How would you avoid cancellation problems induced by multiplying long sequences of very small real numbers?

ANSWER: Take the logarithms. After taking the log, production becomes summation, which is much more numerically stable since the operands are all of the same sign (they are all negative in this case).

$$\log_2(P(y)) = \sum_{i=1}^{\vert y \vert} \log_2(p_{y[i]})$$

Q (optional - 20 extra points): Discuss how to extend your algorithm for a Markov model of order $h > 0$. What is the time complexity of your algorithm? Is it possible to obtain the same time
complexity as for the Bernoulli case, that is $O(n)$ preprocessing once and then constant time for any substring $y$ of the text $x$?

**ANSWER:** Let $\alpha \in \Sigma$ be any symbol, and $s$ be any string of length $h$, i.e., $s \in \Sigma^h$, we denote by $p_t(\alpha|s)$ the transitional probability of seeing next symbol being $\alpha$ given the previous $h$ symbols are as in string $s$. Let’s assume that the stationary probability $p_s: \Sigma^h \to \mathcal{R}(0,1)$ are also known. Then for any string $x, |x| = n \geq h$, the probability of seeing $x$ under the markov model can be computed as follows:

$$P(x) = p_s(x_{[1,h]}) \prod_{i=h+1}^{n} p_t(x_{[i-1,i-1]}|x_{[i-h,i-1]})$$

As we did in the Bernoulli model, we preprocess the text $x$ to calculate the probability $P_i$ for all $i$ prefixes of $x$, where $i \geq h$. The details of the algorithm is as follows:

**Preprocessing $(x)$:**

**Input:** String $x$ of length $n$

**Output:** $P_i$'s, the probabilities of all the $i$ prefixes of $x$, where $i \geq h$

$P_h \leftarrow p_s(x_{[1,h]})$

For ($i \leftarrow h + 1$ to $n$)

$P_i \leftarrow P_{i-1} \times p_t(x_{[i-1,i-1]}|x_{[i-h,i-1]})$

Clearly, the above algorithm also runs in $O(n)$ time.

We notice that for any substring $y = x_{[i,j]}, |y| \geq h$, $P(y) = p_s(x_{[i,j-1+h-1]}) \prod_{k=h}^{j} p_t(x_{[k-1,k-1]}|x_{[k-h,k-1]}) = p_s(x_{[i,i+h-1]}) \times \frac{P_j}{P_{i+h-1}}$. The above calculation can be done in $O(1)$ time.

Therefore we can extend our algorithm to Markov model and obtain the same time complexity as for the Bernoulli case, that is $O(n)$ preprocessing and $O(1)$ constant time for any substring $y$ of text $x$.

**Problem 2: Maximum likelihood (30 points)**

We have seen in class (slide 31 of “Intro to Probability and Statistics”) that the maximum likelihood estimate for the probabilities of the Bernoulli model corresponds to the intuitive idea of computing the proportion of the occurrences in the text for each symbol of the alphabet.

Formally, the ML estimate for a Bernoulli model is

$$p^{ML}_A = \frac{f(A)}{f(A) + f(C) + f(G) + f(T)}$$

and similarly for $p^{ML}_C, p^{ML}_G$ and $p^{ML}_T$. Recall we denote by $f(y)$ the number of occurrence of $y$.

**Q:** Prove that $P(x|p^{ML}) > P(x|p)$ for any other choice of $p \neq p^{ML}$.

**Hint:** show that $\log(P(x|p^{ML})/P(x|p)) > 0$ and use the fact that the relative entropy is always positive.

**ANSWER:** Let $p = (p_A, p_C, p_G, p_T)$ be any choice of the probability other than $p^{ML}$, we prove that $P(x|p^{ML}) > P(x|p)$.
Since we are assuming a Bernoulli model, we have:

\[ P(x|p^{ML}) = p^{ML}_A f(A) * p^{ML}_C f(C) * p^{ML}_G f(G) * p^{ML}_T f(T) \]
\[ P(x|p) = p_A f(A) * p_C f(C) * p_G f(G) * p_T f(T) \]

\[ \log\left(\frac{P(x|p^{ML})}{P(x|p)}\right) = \log\left((\frac{p^{ML}_A}{p_A}) * (\frac{p^{ML}_C}{p_C}) * (\frac{p^{ML}_G}{p_G}) * (\frac{p^{ML}_T}{p_T})\right) \]
\[ = (f(A) * \log(\frac{p^{ML}_A}{p_A}) + f(C) * \log(\frac{p^{ML}_C}{p_C}) + f(G) * \log(\frac{p^{ML}_G}{p_G}) + f(T) * \log(\frac{p^{ML}_T}{p_T})) \]
\[ = (f(A) + f(C) + f(G) + f(T)) * H(p^{ML}|p) \] (according to the definition of relative entropy)

\[ > 0 \] (since relative entropy is always positive and \( f(A) + f(C) + f(G) + f(T) > 0 \))

Therefore: \( \frac{P(x|p^{ML})}{P(x|p)} > 1 \), and which implies \( P(x|p^{ML}) > P(x|p) \).

**Problem 3: Profile HMM (20 points)**

Build a profile HMM on the alignment below. Use one state for columns 1,2,3,7,8,9 and one insert state for columns 4,5,6.

123456789
1 ACAC---ACC
2 AGAAGTCTC
3 ACA---ATG
4 TCAC---ATC
5 ACCG---ATC

**ANSWER:**

State 1 (column 1) has emission probabilities: \( p_A = 0.8, p_C = 0.0, p_G = 0.0, p_T = 0.2 \).
Transition 1 \( \rightarrow \) 2 has probability 1.0
State 2 (column 2) has emission probabilities: \( p_A = 0.0, p_C = 0.8, p_G = 0.2, p_T = 0.0 \).
Transition 2 \( \rightarrow \) 3 has probability 1.0
State 3 (column 3) has emission probabilities: \( p_A = 0.8, p_C = 0.2, p_G = 0.0, p_T = 0.2 \).
Transition 3 \( \rightarrow \) 4 has probability 0.8
Transition 3 \( \rightarrow \) 5 has probability 0.2
Insert state 4 (columns 4,5,6) has emission probabilities: \( p_A = 0.333, p_C = 0.166, p_G = 0.333, p_T = 0.166 \).
Transition 4 \( \rightarrow \) 4 has probability 0.333
Transition 4 \( \rightarrow \) 5 has probability 0.666
State 5 (column 7) has emission probabilities: \( p_A = 0.8, p_C = 0.2, p_G = 0.0, p_T = 0.0 \).
Transition 5 \( \rightarrow \) 6 has probability 1.0
State 6 (column 8) has emission probabilities: \( p_A = 0.0, p_C = 0.2, p_G = 0.0, p_T = 0.8 \).
Transition $6 \rightarrow 7$ has probability 1.0
State 7 (column 9) has emission probabilities: $p_A = 0.0, p_C = 0.2, p_G = 0.2, p_T = 0.0$. 