Assignment batch 2, Algorithmic Bioinformatics, Spring 2010

May 11, 2010

1 Implementation: EM-algorithm

Since some students have express a desire to implement the training algorithm for HMMs, you can choose between 1 and 2 below,

1. If you like, implement the EM-algorithm for training HMMs and show by applying it that it works.

2. Here one implementation assignment is described that is built on a pretty unnatural but simple probabilistic model. In this assigment you are supposed to implement an EM algorithm. First the probabilistic model is described. You will later be able to access data generated from this model, on the course page, so that you can test your implementations on this data and describe the performance.

   The sequences described below are circular and indices are counted modulo n, so $n = 0$. Consider the following probabilistic model with parameters $f_1, \ldots, f_n$ and $\lambda_1, \ldots, \lambda_n$, where f_i is a distribution over $\{1, \ldots, m\}$. A sequence $a_1, \ldots, a_n$ where $a_i \in \{1, \ldots, m\}$ is generated as follows: (1) a direction L or R is chosen for position i, the probability that L is chosen is λ_i and the probability for R is $1 - \lambda_i$ and (2) if i has direction L, then a_i is chosen according to f_{i-1} and otherwise according to f_i.

   The EM-algorithm: For the EM implementation, (1) there is an easy way to find an ML solution where λ_i is 0 or 1 for each i, such solutions are not accepted and (2) f_i is an arbitrary distribution over $[m]$. You should implement a proper EM-algorithm using the derivation of the Q term below. The given samples are denoted $X^1, \ldots, X^l$, i.e., there are l samples. The expected log-likelihood, the Q term, is

   $$Q(\Theta, \Theta_n) = \sum_{j=1}^{l} \sum_{Z \in \{L,R\}^n} \Pr\left[Z | X^j, \Theta_n\right] \log \Pr\left[Z, X^j | \Theta\right],$$

   where Θ_n are the current parameters and Θ are the new parameters (i.e., the parameters that we are seeking). That is $Z = Z_1, \ldots, Z_n$ where Z_i is the direction for a_i (L or R). Let

$$\delta(D, D') = \begin{cases} 1 & \text{if } D = D' \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

and notice

$$\begin{aligned}
\Pr[Z, X|\Theta] &= \prod_{i=1}^{n} \Pr[Z_i, X_i|\Theta] && (2)\\
\Pr[Z_i, X_i|\Theta] &= (\lambda_i f_{i-1}(X_i))^{\delta(Z_i,L)}((1-\lambda_i)f_i(X_i))^{\delta(Z_i,R)} && (3)\\
\log \Pr[Z_i, X_i|\Theta] &= \delta(Z_i, L)(\log \lambda_i + \log f_{i-1}(X_i)) && (4)\\
&\quad +\delta(Z_i, R)(\log(1-\lambda_i) + \log f_i(X_i)) && (5)\\
& && (6)
\end{aligned}$$

$$\begin{aligned}
&\sum_{j=1}^{l} \sum_{Z\in\{L,R\}^n} \Pr\left[Z|X^j, \Theta_n\right] \log \Pr\left[Z, X^j|\Theta\right] && (7)\\
&= \sum_{j=1}^{l} \sum_{Z\in\{L,R\}^n} \Pr\left[Z|X^j, \Theta_n\right] \sum_{i=1}^{n} \log \Pr\left[Z_i, X_i^j|\Theta\right] && (8)\\
&= \sum_{i=1}^{n} \sum_{j=1}^{l} \sum_{D\in\{L,R\}} \Pr\left[Z_i = D|X_i^j, \Theta_n\right] \log \Pr\left[Z_i, X_i^j|\Theta\right] && (9)\\
&= \sum_{i=1}^{n} \sum_{j=1}^{l} (\Pr\left[Z_i = L|X_i^j, \Theta_n\right] (\log \lambda_i + \log f_{i-1}(X_i^j)) && (10)\\
&\quad +\Pr\left[Z_i = R|X_i^j, \Theta_n\right] (\log(1-\lambda_i) + \log f_i(X_i^j)) && (11)
\end{aligned}$$

2 Problems

1. Let M be an HMM. Give an efficient algorithm that for a given sequence $X = x_1, \ldots, x_n$ generates sequences of states, “paths”, according to the distribution $\Pr[\pi_1, \ldots, \pi_n|X, M]$.

2. Let M be an HMM and let p_A, p_C, p_T, p_G be probabilities summing to 1. Let a random sequence of length n be a sequence X of n nucleotides drawn from the distribution induced by p_A, p_C, p_T, p_G (i.e., for any position i, the proability that $x_i = N$ is p_N). Give an efficient algorithm that for a given n computes the probability that a random sequenc of length n satisfies $M(X) \geq t$ (i.e., the probability that M generates X is at least t).