Markov and Hidden Markov Models

Raphael Hoffmann

CSE 527 Lecture 17 Notes, 11/24/03

Reference: Richard Durbin et al. "Biological sequence analysis: probabilistic models of proteins and nucleic acids", Cambridge University Press 1998

1 CpG islands

CpG stands for two adjacent C and G nucleotides along a DNA sequence. (The p in the middle represents the phosphate group in the DNA backbone to distinguish CpG from the Watson/Crick CG pair.) DNA methylation is a chemical modification, addition of a methyl group, to DNA, catalyzed by methyl transferase proteins. DNA methylation has been implicated in control of a variety of cellular processes including transcription, DNA repair, and devlopmental regulation. In higher organisms, DNA methylation, particularly the methylation of the cytosine in CpG dinucleotides, is widespread. Furthermore, methyl-C in a CpG is prone to mutate into a T (i.e. $CpG \rightarrow TpG$). Consequently, CpG is less common than expected in the genome, i.e.

$$frequency(CpG) < frequency(C) \cdot frequency(G).$$

But in gene promoter and other regions which are important for regulation, CpG is usually unmethylated, hence the mutational drift to TpG does not affect these regions. These regions are called CpG islands, because they contain more CpG patterns than other regions. The typical lengths of CpG islands are a few hundred to a few thousand bases. The WMM (weight matrix model) is not useful in this context, because it assumes independence between different positions. However, in this case there is no independence between adjacent positions. A better model is the Markov Chain.

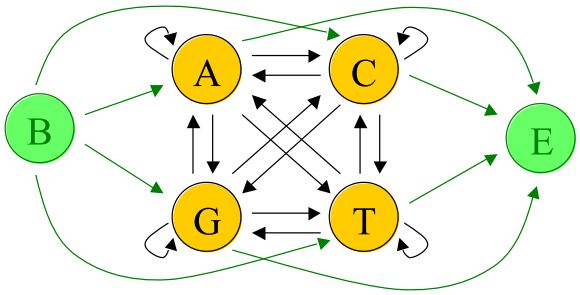
2 Markov Chains

Def. A sequence of random variables $x_1 x_2 ... x_n$ is a k-th order Markov Chain if $\forall i Pr(x_i | x_1 x_2 ... x_{i-1}) = Pr(x_i | x_{i-k} x_{i-k+1} ... x_{i-1})$ i.e. the i-th value is independent of all but the previous k values.

Example 1 A uniform random sequence AACTAG... is a 0-th order Markov Model, because each letter is independent from the rest.

Example 2 Our weight matrix model (WMM) is a 0-th order Markov Model, because it assumes independence.

Example 3 Let's have sequences of A,C,G,T, but where the Pr(G follows C) is lower than the product of Pr(G) and Pr(C). This can be represented by a 1-st order Markov Model. We can visualize the Markov Model with the following graph.



(To simplify formulas, a begin state and an end state have been added. Now, we don't have to worry about boundary positions anymore.) Our model consists of:

States:	A,C,G,T
Emissions:	corresponding letter
Transitions:	$a_{st} = Pr(x_i = t \mid x_{i-1} = s)$

Assuming that we have a sequence $x = x_1 x_2 .. x_n$, then the probability of emitting that sequence is

$$\begin{aligned} Pr(x) &= Pr(x_1, x_2, ..., x_n) \\ &= Pr(x_n | x_{n-1} x_{n-2} ... x_1) \cdot Pr(x_{n-1} | x_{n-2} ... x_1) ... Pr(x_1) \\ &= Pr(x_n | x_{n-1}) \cdot Pr(x_{n-1} | x_{n-2}) ... Pr(x_2 | x_1) \cdot Pr(x_1) \\ &= Pr(x_1) \prod_{i=1}^{n-1} a_{x_i, x_{i+1}} \end{aligned}$$

3 Training - Learning transition probabilites a_{st}

The Maximum Likelihood Estimators (MLE) for the transition probabilities are the frequencies of the transitions observed in training data. For instance, if we are given a sequence ACGTCGCA we can count the number of AC pairs c_{AC}. The transition probability is then defined as

$$a^+_{st} = \frac{c^+_{st}}{\sum_{t'} c^+_{st'}}$$

The +-sign indicates that we are counting in the CpG islands. The transition probabilities in the CpG islands (+) and the transition probabilities in the remainder of the sequence (-) can be written in two tables (where the bases on the left indicate previous bases and bases on top are next bases).

+	A	C	G	T
A	...	...	...	...
C	...	...	0.274	...
G	...	...	...	...
T	...	...	...	...

-	A	C	G	T
A	...	...	...	...
C	...	...	0.078	...
G	...	...	...	...
T	...	...	...	...

In our example, the probability of a CpG pair in + regions is significantly higher than in - regions.

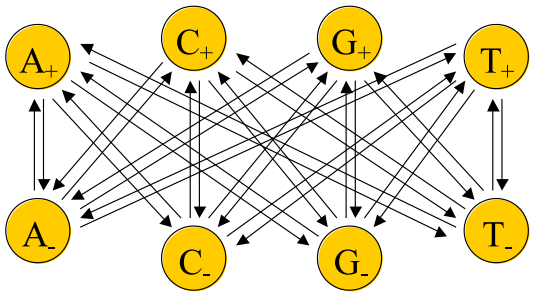
4 Discrimination / Classification - Finding the CpG islands

Question Given a short sequence, is it more likely to be from the feature model or the background model?
We can calculate the log likelihood ratio for CpG model versus background model by

$$S(x) = \log \frac{P(x|+model)}{P(x|-model)} = \sum_{i=1}^{n} \log \frac{a^{+}_{x_{i-1},x_i}}{a^{-}_{x_{i-1},x_i}}$$

As usual, a positive score indicates that the sequence is more probable under the CpG model; negative scores favor the background model.

Question Given a long sequence, where are the features in it?
There are two approaches to this question. One is to use a sliding window and calculate the scores in each step. This method is simple, but it is not clear how to define the window size, particularly when features may vary considerably in length. A better approach is probably to combine the + and - models and create a new model which could look like the following figure. Here, we can transition naturally from our CpG island model to the background model and vice versa. This leads us to Hidden Markov Models.



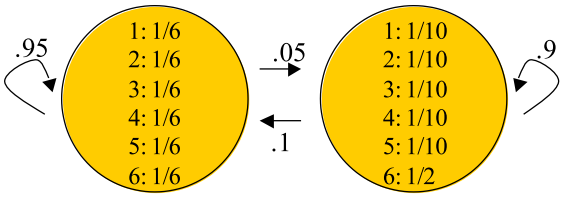
5 Hidden Markov Models

Hidden Markov Models (HMM) consist of the following:

States:	1,2, ...
Paths:	sequences of states $\pi = (\pi_1, \pi_2, ..., \pi_n)$
Transitions:	$a_{kl} = Pr(\pi_i = l \mid \pi_{i-1} = k)$
Emissions:	$e_k(b) = Pr(x_i = b \mid \pi_i = k)$

If we assume that our data was generated by this model, our observed data would be only the emission sequence. Unlike in our previous example, where we identified each state with our current nucleotide (A, C, G or T), we now don't know exactly in what state we are in anymore. Thus, the state/transition sequence can be regarded as hidden data.

Example Let's regard the sequence of rolling a die in a casino. Our opponent cheats from time to time and exchanges the fair die with a loaded die. Our task is to determine the fair die/loaded die sequence by only looking at the sequence of die rolls. Our model could look like this.



Our (observed) emission sequence and (hidden) transition sequence:

Rolls	651166453132651245636666
Die	LLLLLLFFFFFFFFFFFFLLLLL

In Computational Biology we are interested if a sequence, e.g. CGCG, came from $C_+G_+C_+G_+$ or $C_-G_-C_-G_-$ or $C_+G_-C_+G_-$ or ... We don't know the transition sequence π (hidden data). However, we can calculate the joint probability of a given path π and an emission sequence x:

$$Pr(x, \pi) = a_{0\pi_i} \prod_{i=1}^{n} e_{\pi_i}(x_i) a_{\pi_i \pi_{i+1}}$$

Alternative questions that arise in this context are:

- What is the most probable (single) path in our model, if we are given an emission seq. x? $\pi^* = argmax_\pi Pr(x, \pi)$
- What is the sequence of most probable states, if we are given an emission sequence x? $\hat{\pi}_i = argmax_k Pr(\pi_i = k|x)$

We can solve the first question by applying the Viterbi algorithm.

6 The Viterbi Algorithm

The Viterbi algorithm computes $\pi^* = argmax_\pi Pr(x, \pi)$. Hence, it is useful if one path dominates all the others. However, if this is not the case, i.e. many good paths are almost equally likely, then $Pr(x, \pi^*)$ could be very low and other approaches may be preferable.

One key problem to finding π^* is that there are exponentially many paths π. However, the Viterbi algorithm is a dynamic programming approach and is computationally efficient.

Let $v_l(i)$ denote the probability of the most probable path ending in state l, after emitting $x_1 x_2 ... x_i$. We can define v_l recursively using the following formulas:

$$v_l(0) = 1, \text{ if } l \text{ is start state, 0 otherwise}$$
$$v_l(i+1) = e_l(x_{i+1}) max_k(v_k(i) a_{kl})$$

The Viterbi algorithm sweeps through the data in forward direction, calculating $v_l(i)$ for every state l in every step i. We then get π^* by tracing backwards.

There is one important issue when implementing this algorithm: The emission probabilities e and transition probabilities a are typically less than 1 and in most cases less than $\frac{1}{2}$. Thus we end up multiplying thousands of fractions which are below 1, which often causes floating point underflows. One solution to this problem is to use logarithms.