

Hidden Markov models

Claus Thorn Ekstrøm
ekstrom@dina.kvl.dk

Department of Mathematics and Physics, KVL

October 20, 2003

Literature:

Durbin et al (1998): *Biological Sequence Analysis*, Cambridge University Press, pp 46–133

Why?

- Derive a reasonable model for sequence data.
- Classification of sequences (i.e., junk and non-junk DNA)
- Given a sequence determine if it contains any interesting or coding sites and their locations

Overview

- Why?
- Markov models
 - Shakespeare, music, weather, sequences ...
 - Definition
 - Application to sequences
- Hidden Markov models
 - Definition
 - Application to sequences

Shakespeare, music, weather, ...

Imagine we want to write a drama in the style of Shakespeare

Cookbook:

- Collect all of Shakespeare's work
- Define a vocabulary of words used by Shakespeare, $V = \{X_1, \dots, X_N\}$
- Build a model for selecting words from the vocabulary, e.g., $P(X_k | X_1, \dots, X_{k-1})$
- Compose new Shakesperian drama by generating sentences from the model $P(X_k | X_1, \dots, X_{k-1})$.

Shakespeare, music, weather, ...

Letter model:

Zero-order approximation (marginal)

XFOML RXKXRJFFUJ ZLPWCFWKCYJ FFJEYVKCQSGHYD QPAAMKBZAACIBZLHJQD

First-order approximation

OCRO HLI RGWR NWIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH RBL

Second-order approximation

ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT
TEASONARE FUSO TIZIN ANDY TOBE SEACE CITSBE

Third-order approximation

IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE
REPTABIN IS REGOACTIONA OF CRE

Markov models and sequences

- Collect a set of sequences of interest
- Define a vocabulary $V = \{X_1, \dots, X_N\}$
 - DNA sequences: $N = 4$ and $V = \{A, T, C, G\}$
 - Protein sequences: $N = 20$ and $V = \{\text{amino acids}\}$
- Build a model $P(X) = P(X_k | X_1, \dots, X_{k-1})$ and estimate its parameters
- Use the model to
 - Generate typical sequences from the class of sequences examined
 - Compute the probability of an observed sequence, O begin from the model class

Shakespeare, music, weather, ...

Word model:

First-order approximation

REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NATURAL HERE HE
THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE T

Second-order approximation

THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS
POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD
THE PROBLEM FOR AT UNEXPECTED

Could do the same with :

- Music (compose a symphony like Beethoven)
- Predict weather (will it rain tomorrow?)
- Duration of successive eruptions of Old Faithful
- Predict function of gene by classification
- ...

Markov chains, definition and an example: DNA strand

Consider a DNA strand with N consecutive nucleotides $X_1, \dots, X_N$. An example with $N = 20$ is

$$X_1 \dots X_N = CAGCGAGGCAGCGACCTCGT \quad (1)$$

A model sometimes used to describe such a sequence is a Markov chain characterized by a set S of states, in the DNA strand example $S = \{A, C, G, T\}$, and the "Markov property" for transition probabilities:

$$\Pr(X_n = s | X_1, \dots, X_{n-1}) = \Pr(X_n = s | X_{n-1}) \quad (2)$$

Let us also assume that the transition probabilities do not depend on the position n , that is

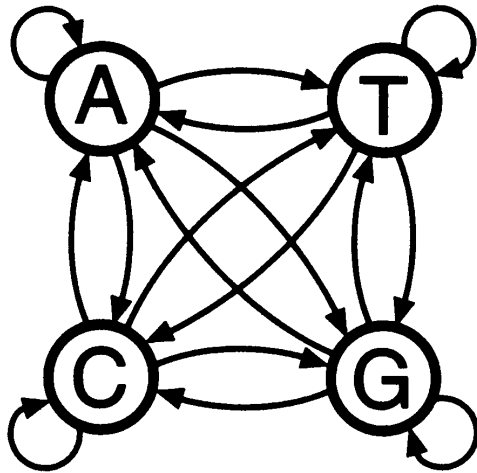
$$\Pr(X_n = s | X_{n-1} = r) = p_{rs} \quad (3)$$

with a matrix

$$P = [p_{rs}] \quad (4)$$

with rows summing to one.

Markov chain graphics



Markov chains, example: CpG islands

A specific pair of nucleotides, *C* followed by *G*, is often written *CpG*. For biological reasons such pairs are relatively infrequent, except in certain regions around the promoters of genes (Bird, 1987). The following estimated probability matrices $\hat{P}^+$ and $\hat{P}^-$ are given in Durbin et al. (1999), p 50, for transitions inside and outside *CpG* islands, respectively. As before the order of the states is *A, C, G, T*.

$$\hat{P}^+ = \begin{bmatrix} 0.180 & 0.274 & 0.426 & 0.120 \\ 0.171 & 0.368 & 0.274 & 0.188 \\ 0.161 & 0.339 & 0.375 & 0.125 \\ 0.079 & 0.355 & 0.384 & 0.182 \end{bmatrix}, \quad \hat{P}^- = \begin{bmatrix} 0.300 & 0.205 & 0.285 & 0.210 \\ 0.322 & 0.298 & 0.078 & 0.302 \\ 0.248 & 0.246 & 0.298 & 0.208 \\ 0.177 & 0.239 & 0.292 & 0.292 \end{bmatrix}. \quad (7)$$

The sequence (1) was actually generated by random sampling from the transition probability matrix $\hat{P}^+$ above.

Markov chains, estimation of transition probabilities

Suppose we have observed $X_1, \dots, X_N$. A suitable estimate (at least for large N) of the transition probability p_{rs} is

$$\hat{p}_{rs} = \frac{\#\{n : 2 \leq n \leq N, X_{n-1} = r \text{ and } X_n = s\}}{\#\{n : 2 \leq n \leq N \text{ and } X_{n-1} = r\}}, \quad (5)$$

where $\#A$ denotes the number of elements in a finite set A . For the observed sequence (1), that is,

$$X_1 \dots X_N = CAGCGAGGCAGCGACCTCGT$$

we find the following estimate of the transition probability matrix with the states in the order *A, C, G, T*

$$\hat{P} = \begin{bmatrix} 0/4 & 1/4 & 3/4 & 0/4 \\ 2/7 & 1/7 & 3/7 & 1/7 \\ 2/7 & 3/7 & 1/7 & 1/7 \\ 0/1 & 1/1 & 0/1 & 0/1 \end{bmatrix} \quad (6)$$

Comparing a sequence to two model classes

Which of two different model classes (transition matrices) do we believe that the current sequence belongs to?

Compare the likelihood of the sequence given one model to the likelihood of the sequence given the other model:

$$\log \frac{P(x | \text{model } +)}{P(x | \text{model } -)}$$

Use log-transform to obtain easy calculations:

$$S(x) = \sum_{i=1}^L \log p_{x_{i-1}x_i}^+ - \sum_{i=1}^L \log p_{x_{i-1}x_i}^- = \sum_{i=1}^L \log \left(\frac{p_{x_{i-1}x_i}^+}{p_{x_{i-1}x_i}^-} \right)$$

High value of S : the model in the numerator is much more likely.

Low value of S : the model in the denominator is much more likely.

Hidden Markov models

Question: How do we locate *CpG* islands in a long sequence?

Two possibilities:

- Examine pre-specified blocks of sequences, e.g., 100 nucleotides, and compare the two classes for each block.
- Extend the Markov model such that it includes both *CpG*-rich and *CpG*-poor regions.

Hidden Markov models: Like ordinary Markov models but some information about the current state is unknown (hidden).

The true Markov chain is only indirectly observable via relative frequencies for nucleotide shifts.

Hidden Markov model

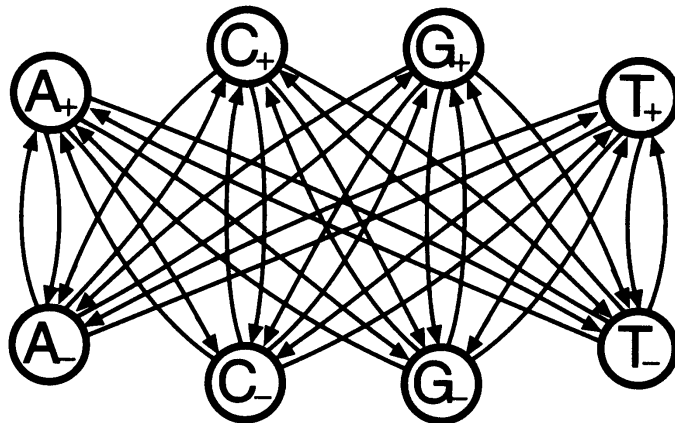
Definition:

- *Transition probabilities*: from state to state: $a_{kl} = P(\pi_i = l | \pi_{i-1} = k)$
- *Emission probabilities*: Once in state k what is the probability of observing b :
 $e_k(b) = P(x_i = b | \pi_i = k)$
- Joint probability of observed sequence and a state sequence π :

$$P(x, \pi) = a_{0\pi_1} \prod_{i=1}^L e_{\pi_i}(x_i) a_{\pi_i \pi_{i+1}}$$

Potential problem: we don't know the state sequence π .

Hidden Markov chain graphics



Find most probable state path

Used to locate *CpG* regions.

Want to estimate

$$\pi^* = \operatorname{argmax}_{\pi} P(x, \pi)$$

π^* the true state that makes the observed data most likely.

π^* can be found recursively by the Viterbi algorithm.

May be useful but can be rather imprecise — how more likely is the most probable state path compared to the next most probable state path?

Useful for shorter sequences.

Calculating $P(x)$ for HMM

Add the probabilities for all possible states:

$$P(x) = \sum_{\pi} P(x, \pi)$$

The number of states increases exponentially so the above sum may be (prohibitively) large.

Elegant solution — Forward algorithm:

- Initialisation

$$i = 0, P(x_0, \pi_0 = 0) = 1, P(x_0, \pi_0 = k) = 0 \text{ for } k > 0$$

- Recursion

$$P(x_1, \dots, x_i, \pi_i = l) = e_l(x_i) \sum_k P(x_1, \dots, x_{i-1}, \pi_{i-1} = k) a_{kl}$$

- Termination

$$P(x) = \sum_k P(x_1, \dots, x_L, \pi_L = k)$$

Posterior state

What is the probability that observation x_i came from state k ?

$$P(\pi_i = k | x) = \frac{P(\pi_i = k, x)}{P(x)}$$

Calculate

$$\begin{aligned} P(\pi_i = k, x) &= P(x_1, \dots, x_i, \pi_i = k) P(x_{i+1}, \dots, x_L | x_1, \dots, x_i, \pi_i = k) \\ &= P(x_1, \dots, x_i, \pi_i = k) \underbrace{P(x_{i+1}, \dots, x_L | \pi_i = k)}_{b_k(i)} \end{aligned}$$

Recursive algorithm (p. 59) can be used to calculate the $b_k(i)$'s recursively.

$$P(\pi_i = k | x) = \frac{P(x_1, \dots, x_i, \pi_i = k) b_k(i)}{P(x)}$$