Markov Models

Section 7

Y. ‘woodie’ Zhao

November 4th-6th, 2003

Outline

- •Overview
- •Simple Markov Chains
- •Hidden Markov Models

Definitions

- **Markov chain—**a collection of variables {X_i} (where the index *i* runs through 0, 1, ...) with a probability for transition between each two adjacent variables a_{kl}, which is the probability that the current value is $X_i=l$ given that the previous value $X_{i-1}=k$.

$$\begin{array}{c}k \xrightarrow[a_{kl}]{} l\\ X_{i-1} \quad X_i\end{array}$$

- **Hidden Markov Model**— a variant of a Markov chain having a set of states, a finite set of outputs, transition probabilities, output probabilities, and initial state probabilities. The current state is not observable. Instead, each state produces an output with a certain probability.

Biological Applications

- Determining if a certain sequence (protein or nucleotide) is a member of a particular family.
- Secondary structure prediction.
- Pair-wise and multiple alignment of sequences.
- Determining if parts of a given sequence belong to certain regions/structures/motifs (states): exons, introns, CpG islands, etc. This is the one we will study at length.

Joint Probability

The general definition of the joint probability Pr(A,B) is as follows:

$$\boldsymbol{P(A,B) = P(A|B)P(B)}$$

Where:

•P(B) = probability of B occurring.

•P(A|B) = probability of A occurring given B – *conditional probability.* $P(A|B) = \frac{P(A \cap B)}{P(B)}$

•P(A,B) = probability of having B and A occurring together – *joint probability.*

Joint Probability Example

<u>Application</u>: What is the probability of observing a given DNA sequence? Say: “ACGTC.”

- The probability of seeing a given sequence of length *L* can be determined from the joint probability of having each nucleotide at its specific position:

$$Pr(seq) = P(b_L ,b_{L-1} ,b_{L-2} ,\ldots ,b_1)$$

- Using our example “ACGTC”, the joint probability of this sequence is written as:

$$Pr(ACGTC) = P(C_5,T_4,G_3,C_2,A_1)$$

- Using the equation **P(A,B) = P(A|B)P(B)**,

$$Pr(seq) = Pr(b_L \mid b_{L-1} ,b_{L-2} ,\ldots ,b_1)\ Pr(b_{L-1} ,b_{L-2} ,\ldots ,b_1)$$

or in the case of the example,

$$Pr(ACGTC) = P(C_5|T_4,G_3,C_2,A_1)\ P(T_4,G_3,C_2,A_1)$$

Markov Chain 1

Fundamental Simplifying Assumption of Markov Models: The Pr= $Pr(b_L \mid b_{L-1})\ Pr(b_{L-1} \mid b_{L-2})\ Pr(b_{L-2} \mid b_{L-3}) \ldots\ Pr(b_1)$ probability of obtaining a certain state at X_i depends **only** on the state at X_{i-1}, i.e. the previous position of the Markov chain.

$Pr(seq) = P(b_L, b_{L-1}, b_{L-2}, \ldots, b_1)$
$= Pr(b_L \mid b_{L-1})\ Pr(b_{L-1}, b_{L-2}, \ldots, b_1)$
$= Pr(b_L \mid b_{L-1})\ Pr(b_{L-1} \mid b_{L-2})\ Pr(b_{L-2} \mid b_{L-3}) \ldots Pr(b_1)$ Here

Pr(b1) is different from the rest – it's the probability of starting the sequence with base b1.

For our example:

$P(ACGTC) = P(C_5, T_4, G_3, C_2, A_1)$
$= P(C_5 \mid T_4, G_3, C_2, A_1)\ P(T_4, G_3, C_2, A_1)$
$= P(C_5 \mid T_4, G_3, C_2, A_1)\ P(T_4 \mid G_3, C_2, A_1)\ P(G_3 \mid C_2, A_1)\ P(C_2, A_1)$
$= P(C_5 \mid T_4)\ P(T_4 \mid G_3)\ P(G_3 \mid C_2)\ P(C_2 \mid A_1)\ P(A_1)$

Markov Chain 2

Application: we can use Markov models to figure out whether a DNA sequence came from an intron or an exon. From our example,

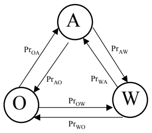
$P(ACGTC) = P(C_5 \mid T_4)\ P(T_4 \mid G_3)\ P(G_3 \mid C_2)\ P(C_2 \mid A_1)\ P(A_1)$
$= P(beg.\ w/A) * a_{AC} * a_{CG} * a_{GT} * a_{TC}$

- The conditional probabilities above are different in introns and exons. Certain dinucleotides are more likely found in introns than exons, and others vice versa. So plug in these different conditional probabilities derived from introns and exons, and whichever model produces a higher probability is more likely where the sequence comes from.
- We can use a training set of sequences where it's known which regions (intron/exon) they came from. Use the frequencies of each dinucleotide and starting nucleotide within the training set to calculate probabilities for each dinucleotide both in exons and in introns.

Markov Chain 3

Another example: the honest/dishonest slot machine.

	Honest	Dishonest
WIN	1/3	1/10
Apple	1/3	4/10
Orange	1/3	5/10



State Diagram

Markov chain: Individual slot machines are either honest or dishonest, we can find out which by looking at the frequencies and confirm by the equations established in the previous section:

- WWAOAWWOAOAWOAWOOAAWAOW – probably honest
- WAOAOAOOOOAAOOWAAOOOOWOOO–probably dishonest

HMM 1

Hidden Markov Models

In the case of hidden Markov models, the observed sequence can come from a number of underlying "states". We know the resulting sequence, but not the states that produced them. Let's go back to our casino example:

- Owner can switch machine between honest and dishonest states at will without the player knowing. The goal is to decide when to play by finding times when it's honest.
- Six possible states: A_H, W_H, O_H, A_D, W_D, O_D. To construct a state diagram, you need to draw transitions between all of them.

Markov Chains vs. HMMs

- Hidden Markov models are an extension of simple Markov chains.
- In a hidden Markov model, the observations $x_1, x_2 \ldots x_n$, may arise from different underlying states $s_1, s_2 \ldots s_n$. We know the observations but not necessarily the states that produced them.
- In Markov chains you know the state sequence, in HMMs you do not. This is the most important distinguishing feature between the two.

HMM 2

Besides the transition probability:

$$a_{kl} = \Pr(X_i = l \mid X_{i-1} = k)$$

where *l* and *k* are the states at positions *i* and *i-1*,

we often add more detail by allowing states to "emit" different observations. The emission probability:

$$e_{kb} = \Pr(O_i = b \mid X_i = k)$$

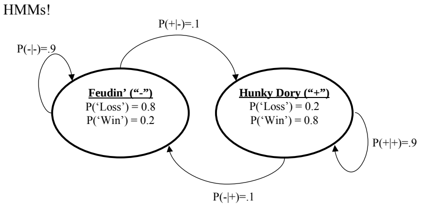
is the probability that the state *k* at position *i* emits observation *b* (X_i is the state at position *i*, and O_i is the observation at that position). Now we can write the probability of an individual path through a sequence of hidden states as:

$$P(\vec{O}, \vec{X}) = a_{oX_1} \prod_i e_{O_i X_i} a_{X_i X_{i+1}}$$

HMM 3

Another real world example:

The Los Angeles Lakers are a wonderful team that would win 80% of its games if everything were hunky dory. However, its two biggest stars, Shaq and Kobe, often have monstrous **private** feuds that disrupt the chemistry of the team such that the win/loss probability is flip-flopped. On any given game day, Shaq and Kobe have a 0.1 probability of feuding, and once a feud starts there's only a 0.1 probability that they work things out and bring the team back to the hunky dory state. The casual observer/fan only sees if the team wins or loses, but wants to know when Shaq and Kobe have been feuding. This can be done by HMMs!



HMM 4

Biological example: CpG islands

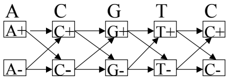
We have an observed sequence where we do not know whether or not a given nucleotide is inside a CpG island. The state (*inside* or *outside*) is "hidden." We can denote the hidden states with + (inside an island) and – (outside an island.) For example A+ denotes an adenine that is inside an island and A– an adenine that is outside an island.

Now, let's examine the same observed sequence, ACGTC using the above hidden Markov model and recalculate the possibility of observing this sequence. In this case the observations could have been the result of many (actually $2^5 = 32$) different sequences of hidden states A+C+G+T+C+, A+C–G–T+C+, A+C+G–T–C+ etc. To find the probability of observing this sequence, we need to find the sum of the probabilities of all 32 paths producing these observations, or the total probability.

HMM 5

In the simple Markov chain, there was only one possible path that produced the observations. Now, in the hidden Markov model there are many. We must consider all possible paths that could produce the observations. The total probability is given by the sum of the probabilities of each of these possible paths.

In our model, where we have two possible hidden states at each step in the sequence, there are $2^5 = 32$ paths. And the number of paths increases exponentially with the length of the sequence. Fortunately, there is a computationally efficient way to calculate the total probability of a sequence using dynamic programming.

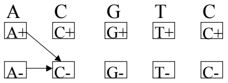


HMM 6

The dynamics programming trick that we use to make the sum computationally efficient uses partial probabilities.

A partial probability is the probability of the model being in state *k* at position *i*. Let's begin by looking at the first transition. The first partial probabilities that we compute are those for being in the states C- and C+ at position 2. We'll denote the first partial probability as:

$$\alpha_{\text{c-},2} = (a_{\text{A+, C}-})(a_{0,\text{ A+}}) + (a_{\text{A}-,\text{ C}-})(a_{0,\text{ A}-})$$

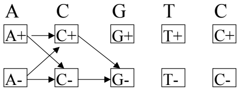


Visual representation

HMM 7

Continuing with the next position :

$$\alpha_{\text{G}-,3} = (a_{C+,\text{ G}-})(\alpha_{\text{C+},2}) + (a_{\text{C}-,\text{ G}-})(\alpha_{\text{C}-,2})$$



We can see how this algebraically expands into the sum of all of the paths through the model which end in G- by writing out explicitly $\alpha_{c+,2}$ and $\alpha_{c-,2}$:

$$\alpha_{\text{G}-,3} = (a_{C+,\text{ G}-})[(a_{A+,\text{ C+}})(a_{0,\ A+}) + (a_{A-,\text{ C+}})(a_{0,\ A-})] +$$

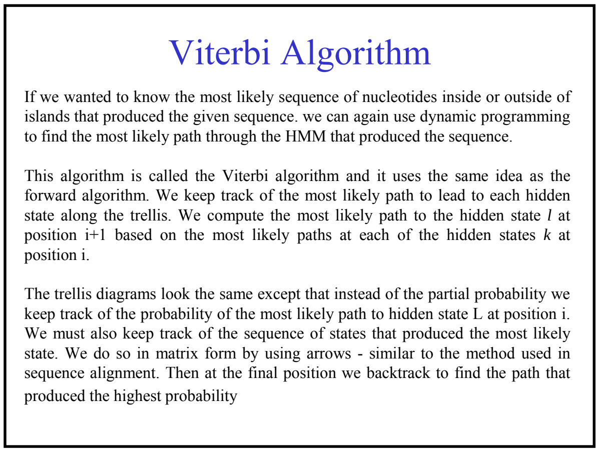
$$(a_{\text{C}-,\text{ G}-})[(a_{A+,\text{ C}-})(a_{0,\ A+}) + (a_{A-,\text{ C}-})(a_{0,\ A-})]$$

Forward Algorithm

This recursion eventually gives:

$$\alpha_{l,\text{i}} = \sum_{i} a_{kl}\alpha_{k,\text{i-1}}$$

The algorithm we've just used to calculate the probability of observing "ACGTC" using our HMM is called the "forward algorithm."

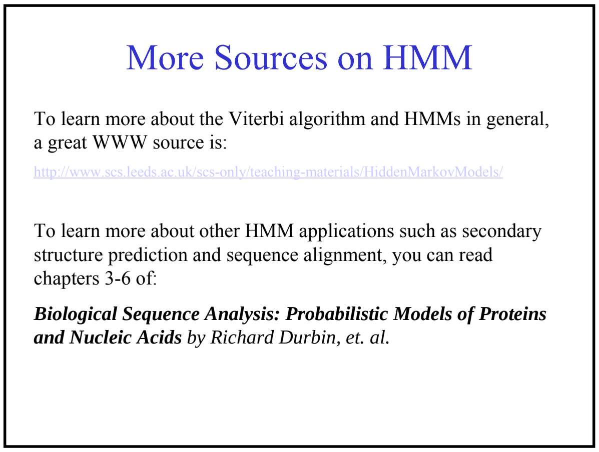


Viterbi Algorithm

If we wanted to know the most likely sequence of nucleotides inside or outside of islands that produced the given sequence. we can again use dynamic programming to find the most likely path through the HMM that produced the sequence.

This algorithm is called the Viterbi algorithm and it uses the same idea as the forward algorithm. We keep track of the most likely path to lead to each hidden state along the trellis. We compute the most likely path to the hidden state *l* at position i+1 based on the most likely paths at each of the hidden states *k* at position i.

The trellis diagrams look the same except that instead of the partial probability we keep track of the probability of the most likely path to hidden state L at position i. We must also keep track of the sequence of states that produced the most likely state. We do so in matrix form by using arrows - similar to the method used in sequence alignment. Then at the final position we backtrack to find the path that produced the highest probability



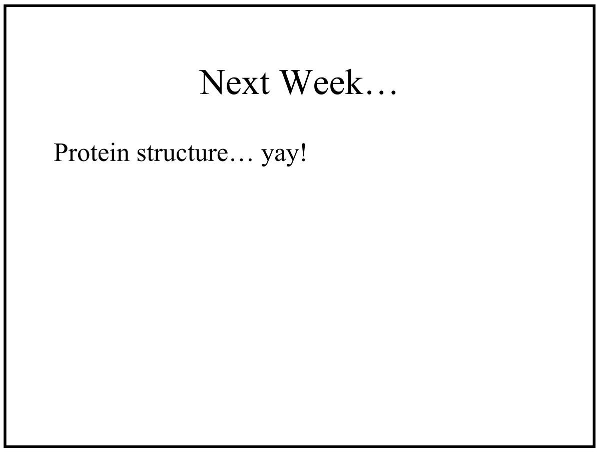
More Sources on HMM

To learn more about the Viterbi algorithm and HMMs in general, a great WWW source is:

http://www.scs.leeds.ac.uk/scs-only/teaching-materials/HiddenMarkovModels/

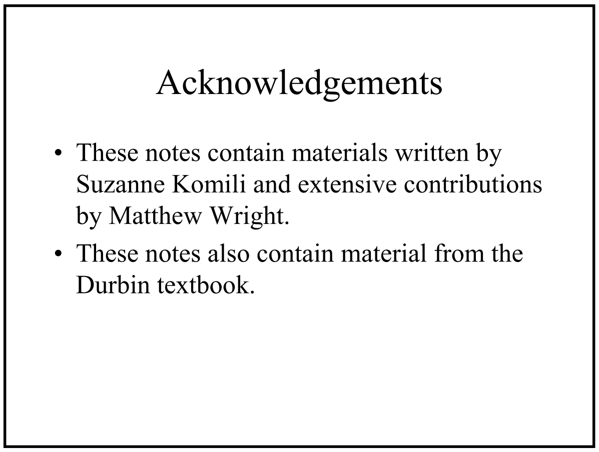
To learn more about other HMM applications such as secondary structure prediction and sequence alignment, you can read chapters 3-6 of:

Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids *by Richard Durbin, et. al.*



Next Week…

Protein structure… yay!



Acknowledgements

- These notes contain materials written by Suzanne Komili and extensive contributions by Matthew Wright.
- These notes also contain material from the Durbin textbook.