CS284A: Introduction to Computational Biology **Winter 2008**

Lecture 10 — February 27

Scribe: Sidharth Shekhar *Lecturer: Xiaohui S. Xie*

10.1 Markov Chains and Hidden Markov Models

10.1.1 Markov Chains

Consider a set of states $X_1, X_2, ..., X_n$. The joint probability is given by

$$P(X_1, X_2, ..., X_n) = P(X_n|X_{n-1}, ..., X_1)P(X_{n-1}|X_{n-2}, ..., X_1)....P(X_1) \tag{10.1}$$

The Markov property states that each state is dependent only on its previous state. i.e.

$$P(X_i|X_{i-1}, ..., X_1) = P(X_i|X_{i-1}) \tag{10.2}$$

Therefore the joint probability can be written as

$$P(X_1, X_2, ..., X_n) = P(X_n|X_{n-1})P(X_{n-1}|X_{n-2})....P(X_2|X_1)P(X_1) \tag{10.3}$$

Suppose $X_i \in \{1, 2, ..., k\}$ i.e. X can take one of k possible states. Then we define

$$\begin{aligned} \text{Transition probability: } a_{kl} &= P(X_i = l|X_{i-1} = k) && (10.4)\\ \text{Initial probability: } & \quad P(X_1 = k) && (10.5) \end{aligned}$$

CpG islands

In the human genome, there are certain short stretches of the genome where the dinucleotide CG, (often written as C_pG to distinguish it from the C-G base pair across two strands), has higher frequency than elsewhere. Such regions are called C_pG islands. Let us consider the following problem. Given a short stretch of genomic sequence $X_1, X_2, ..., X_L$, we need to decide if it comes from a C_pG island or not. We can model this problem as a Markov chain by assuming that each nucleotide in the sequence is dependent only on the previous one. Now, assume that we are given the following probabilities.

For a C_pG island	Initial probability $P_1^+(k)$	Transition probabilities a_{kl}^+
For a non C_pG island	Initial probability $P_1^-(k)$	Transition probabilities a_{kl}^-

The simple way is to calculate the likelihood of the sequence $X_1, X_2, ..., X_L$ being generated by the C_pG model and by the non C_pG model and then take the log ratio of the two.

i.e.

$$\begin{aligned} P(X_1, X_2, ..., X_L | C_pG \ model) &= P_1^+(X_1) a^+_{X_1X_2} a^+_{X_2X_3} .... a^+_{X_{L-1}X_L} && (10.6) \\ P(X_1, X_2, ..., X_L | non \ C_pG \ model) &= P_1^-(X_1) a^-_{X_1X_2} a^-_{X_2X_3} .... a^-_{X_{L-1}X_L} && (10.7) \\ log \frac{P(X_1, X_2, ..., X_L | C_pG)}{P(X_1, X_2, ..., X_L | non \ C_pG)} &= log \frac{P_1^+(X_1)}{P_1^-(X_1)} + \sum_{i=2}^{L} log \frac{a^+_{X_{i-1}X_i}}{a^-_{X_{i-1}X_i}} && (10.8) \end{aligned}$$

Therefore, if this log likelihood ratio is positive, then it is more likely that the sequence is from a C_pG island, otherwise it is more likely that it is not from a C_pG island.

Now consider another problem. Given a long sequence, we need to identify the C_pG islands in that sequence. To go about solving this problem, we first need to introduce the concept of Hidden Markov Models.

10.1.2 Hidden Markov Models

To understand the basic concepts of Hidden Markov Models we look to a simple problem from Vegas. Consider a casino dealer who has 2 dice with him. One is a fair die i.e. each of the six numbers has an equal probability of turning up on a roll of the die. The other die is biased towards the number 6. The dealer can start with either one of the two dice and generally uses the same die most of the time. But sometimes he will switch the dice and once switched he again uses the die for most of the time. Now all we have is a sequence of observations, i.e. values that die in play turned up in each roll. Given these observations our task is to best predict which die was in use in each of the rolls. Let us now formalize this problem.

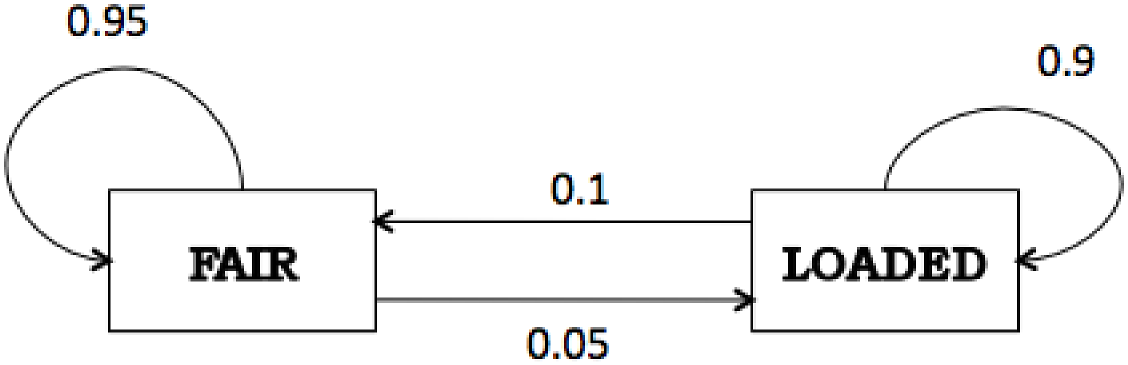

Figure 1

As shown in Figure 1, we can model the environment as a finite state machine. For the game, at any time it can be in one of only two possible states i.e. the fair die is in play or the loaded die is in play. Therefore, our states

$$Q = \{Fair, Loaded\} \tag{10.9}$$

Each die can take only one of 6 possible numbers. Therefore, our alphabet

$$\Sigma = \{1, 2, 3, 4, 5, 6\} \tag{10.10}$$

The transition probabilities can be given by the following matrix

$$A = \begin{bmatrix} a_{FF} & a_{FL} \\ a_{LF} & a_{LL} \end{bmatrix} \tag{10.11}$$

For our example we can consider these transition probabilities to be as follows

$$A = \begin{bmatrix} 0.95 & 0.05 \\ 0.1 & 0.9 \end{bmatrix} \tag{10.12}$$

Once in a state, the probability of the die turning up a particular value is called the emission probability. In our problem we can consider them to be as follows

$$e_{fair}(i) = \frac{1}{6}, \, i = 1, 2, ..., 6 \tag{10.13}$$

$$e_{loaded}(i) = \begin{cases} \frac{1}{2} & i = 6 \\ \frac{1}{10} & i = 1, 2, ..., 5 \end{cases} \tag{10.14}$$

Therefore, our Hidden Markov Model is completely specified as

$$M = (Q, \Sigma, A, e) \tag{10.15}$$

Now let us consider the actual problem. We are provided with this model M. We also have a set of observations $X_1, X_2, ..., X_L$, where X_i stands for the value of the die on the ith roll. We have to find $Z_1, Z_2, ..., Z_L$ where Z_i stands for the state of the die in the ith roll. Let us represent this pictorially.

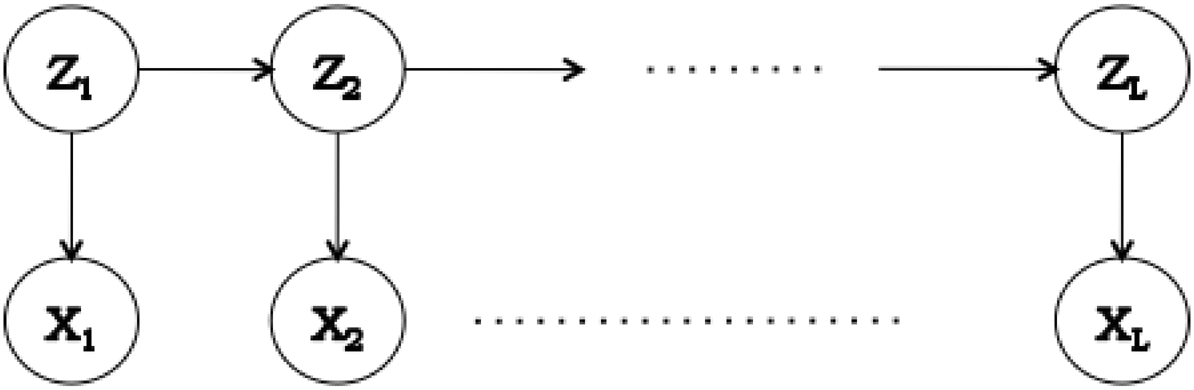

Figure 2

Because of the way our model has been defined, we can clearly see that each state Z_i is only dependent on its previous state Z_{i-1} (i.e. the Markov Property), and the value of the die X_i is only dependent on the state Z_i of the die. A possible question that could come to mind here is that why do we call it Hidden Markov Model? The reason is that the states $Z_1, Z_2, ..., Z_L$ are not directly observable. Only the dealer knows what die is in play. The only information that we have is the model and the observations. So now our task has become to find the values $Z_1^*, Z_2^*, ..., Z_L^*$ that best explain the observations $X_1, X_2, ..., X_L$. In other words

$$(Z_1^*, Z_2^*, ..., Z_L^*) = \max_{Z_1, Z_2, ..., Z_L} P(X_1, X_2, ..., X_L, Z_1, Z_2, ..., Z_L) \tag{10.16}$$

But due to the Markov Property, we can write this as

$$(Z_1^*, ..., Z_L^*) = \max_{Z_1, ..., Z_L} P(Z_1)P(X_1|Z_1)P(Z_2|Z_1)P(X_2|Z_2)...P(Z_L|Z_{L-1})P(X_L|Z_L) \tag{10.17}$$

$$(Z_1^*, ..., Z_L^*) = \max_{Z_1, ..., Z_L} P(Z_1) \prod_{i=2}^{L} P(Z_i|Z_{i-1})P(X_i|Z_i) \tag{10.18}$$

A naive way of solving this maximization would be to try all 2^L possible combinations of Z_is and choose the maximum. But even in our simple problem where Z can take only 2 states, it is easy to see that this method quickly becomes infeasible. A more careful look at the equation suggests a better way of solving the problem. Consider the maximization over the value of Z_L. Most of the terms in the product do not depend on Z_L and hence can be taken out of the maximization to get

$$(Z_1^*, ..., Z_L^*) = \max_{Z_1, ..., Z_{L-1}} P(Z_1)P(X_1|Z_1)...P(Z_{L-1}|Z_{L-2})P(X_{L-1}|Z_{L-1}) \max_{Z_L} P(Z_L|Z_{L-1})P(X_L|Z_L) \tag{10.19}$$

Using the same technique for the other variables, we get

$$(Z_1^*, ..., Z_L^*) = \max_{Z_1} P(Z_1)P(X_1|Z_1) \max_{Z_2} P(Z_2|Z_1)P(X_2|Z_2).... \max_{Z_L} P(Z_L|Z_{L-1})P(X_L|Z_L) \tag{10.20}$$

Due to this, we can now write a recursive algorithm to solve the problem. The following algorithm is known as the Viterbi algorithm.

1. Initialize: $V(Z_{L-1}) = \max_{Z_L} P(Z_L|Z_{L-1})P(X_L|Z_L)$

2. Repeat for $i = L-2, L-3, ..., 1$

   $V(Z_i) = \max_{Z_{i+1}} P(Z_{i+1}|Z_i)P(X_{i+1}|Z_{i+1})$

3. Final Score = $\max_{Z_1} P(Z_1)P(X_1|Z_1)V(Z_1)$

Does this algorithm do better than the naive algorithm? Let us compare the time complexities. For the naive algorithm, we try 2^L possible values and for each value we will have to do $O(L)$ calculations. Therefore, time complexity is $O(L2^L)$. Viterbi algorithm needs to calculate L maximizations. Therefore, that is $O(L)$ calculations. For each of these L maximizations, we have $2^2 = 4$ possible values that we need to check. So in our example, time complexity of Viterbi algorithm is just $O(L)$. In general, we can have more than 2 states, so we denote the number of states in our model M as $|Q|$. Therefore time complexity now becomes, $O(L|Q|^2)$.

A question that arises is that we developed the Viterbi algorithm in a backward manner, i.e. we first maximize over Z_L, then Z_{L-1}, and so on. Can we do it in a forward pass as well? The answer is yes we can. We just have to redefine our functions a little bit. The Viterbi algorithm using the forward technique is as follows.

1. Initialize: $V(Z_2) = \max_{Z_1} P(Z_1)P(X_1|Z_1)P(Z_2|Z_1)$
2. Repeat for $i = 3, 4, ..., L$

   $V(Z_i) = \max_{Z_{i-1}} P(Z_i|Z_{i-1})P(X_{i-1}|Z_{i-1})V(Z_{i-1})$
3. Final Score = $\max_{Z_L} P(Z_L|Z_{L-1})P(X_L|Z_L)V(Z_L)$

Inference Problem

The problem that we just solved is known as the decoding problem. Another problem we have is known as the Inference problem. Here, we want to know, what is the probability of observing the sequence that we have, i.e. we want to find $P(X_1, X_2, ..., X_L)$. But

$$P(X_1, X_2, ..., X_L) = \sum_{Z_1, Z_2, ..., Z_L} P(X_1, X_2, ..., X_L, Z_1, Z_2, ..., Z_L) \tag{10.21}$$

Equation (10.21) is very similar to equation equation (10.16), except that the maximization operator has been replaced by the summation operator. Therefore, we can follow the exact same methodology that we developed for the decoding problem and use it to solve this problem. The steps are exactly the same, except that at each stage instead of taking $\max_{Z_i}$ we will take $\sum_{Z_i}$.

Forward Algorithm for Inference

Let us define the function $f_k(i)$ as follows

$$f_k(i) = P(X_1, X_2, ..., X_i, Z_i = k) \tag{10.22}$$

Hence, we can write

$$\begin{aligned}
f_l(i+1) &= P(X_1, X_2, ..., X_{i+1}, Z_{i+1} = l) && (10.23)\\
f_l(i+1) &= \sum_k P(X_1, ..., X_i, X_{i+1}, Z_i = k, Z_{i+1} = l) && (10.24)\\
f_l(i+1) &= \sum_k P(X_1, ..., X_i, Z_i = k) P(Z_{i+1} = l | Z_i = k) P(X_{i+1} | Z_{i+1} = l) && (10.25)\\
f_l(i+1) &= \sum_k f_k(i) P(Z_{i+1} = l | Z_i = k) P(X_{i+1} | Z_{i+1} = l) && (10.26)
\end{aligned}$$

Therefore, the final likelihood can be written as

$$P(X_1, X_2, ..., X_L) = \sum_k f_k(L) \qquad (10.27)$$

Backward Algorithm for Inference

Let us define the function $b_k(i)$ as follows

$$b_k(i) = P(X_{i+1}, X_{i+2}, ..., X_L | Z_i = k) \qquad (10.28)$$

Hence we can write

$$\begin{aligned}
b_l(i-1) &= P(X_i, ..., X_L | Z_{i-1} = l) && (10.29)\\
b_l(i-1) &= \sum_k P(X_i, X_{i+1}, ..., X_L, Z_i = k | Z_{i-1} = l) && (10.30)\\
b_l(i-1) &= \sum_k P(X_i | Z_i = k) P(Z_i = k | Z_{i-1} = l) b_k(i) && (10.31)
\end{aligned}$$

Therefore, the final likelihood can be written as

$$P(X_1, X_2, ..., X_L) = \sum_k P(X_1 | Z_1 = k) P(Z_1 = k) b_k(1) \qquad (10.32)$$

Posterior Inference

We now look at the problem, wherein given that we have observed a particular sequence $\{X_1, X_2, ..., X_L\}$, we want to find the probability that a particular state Z_i takes on the value k. i.e. we want to find $P(Z_i = k | X_1, X_2, ..., X_L)$. We can write this posterior probability in terms of the joint probability as follows

$$P(Z_i = k | X_1, X_2, ..., X_L) = \frac{P(X_1, X_2, ..., X_L, Z_i = k)}{P(X_1, X_2, ..., X_L)} \qquad (10.33)$$

We already know how to calculate the denominator. We need to find a simple way to calculate the numerator, which can be done as follows

$$\begin{aligned} P(X_1,...,X_L,Z_i=k) &= P(X_1,X_2,...,X_i,Z_i=k,X_{i+1},....,X_L) && (10.34)\\ P(X_1,...,X_L,Z_i=k) &= P(X_1,..,X_i,Z_i=k)P(X_{i+1},..,X_L|X_1,..,X_i,Z_i=k) && (10.35)\\ P(X_1,...,X_L,Z_i=k) &= f_k(i)P(X_{i+1},...,X_L|Z_i=k) && (10.36)\\ P(X_1,...,X_L,Z_i=k) &= f_k(i)b_k(i) && (10.37) \end{aligned}$$

We can go from (10.35) to (10.36) because of the Markov Property. Thus, the final posterior can be calculated using

$$P(Z_i=k|X_1,X_2,.....,X_L) = \frac{f_k(i)b_k(i)}{P(X_1,X_2,.....,X_L)} \qquad (10.38)$$

An extension to this posterior inference is to calculate $P(Z_i=k,Z_{i+1}=l|X_1,X_2,.....,X_L)$. We can again do this using the joint probability

$$P(Z_i=k,Z_{i+1}=l|X_1,...,X_L) = \frac{P(X_1,...,X_L,Z_i=k,Z_{i+1}=l)}{P(X_1,...,X_L)} \qquad (10.39)$$

Now, the joint probability can be written as

$$\begin{aligned} P(X_1,...,X_L,Z_i,Z_{i+1}) &= P(X_1,..,X_i,Z_i=k)P(X_{i+1},..,X_L,Z_{i+1}=l|Z_i=k) && (10.40)\\ &= f_k(i)P(Z_{i+1}=l|Z_i=k)P(X_{i+1},..,X_L|Z_{i+1}=l) && (10.41)\\ &= f_k(i)P(Z_{i+1}=l|Z_i=k)P(X_{i+1}|Z_{i+1}=l)b_l(i+1) && (10.42) \end{aligned}$$

Thus, the required posterior can be calculated as

$$P(Z_i=k,Z_{i+1}=l|X_1,...,X_L) = \frac{f_k(i)P(Z_{i+1}=l|Z_i=k)P(X_{i+1}|Z_{i+1}=l)b_l(i+1)}{P(X_1,...,X_L)} \qquad (10.43)$$

10.1.3 Extensions by Sohail Jahid

CpG islands

They are called CpG islands or CpG-rich islands (CGIs) because they are found in a "sea" of DNA sequences low in CG content and the p denotes the phosphodiester bond between cytosine and guanine. CGIs are generally associated with promoters; genes, whose promoters are especially rich in CpG sequences, tend to be expressed in most tissues. CGIs have an average 60% GC content as compared to 40% in random DNA sequences and they extend over 100-1000 nucleotides. They are usually found just upstream of a promoter and extend downstream into the transcribed regions of a gene. Furthermore, genes that are constitutively expressed (on all the time) are surrounded by CGIs in the 5 region. Most CpGIs are found in the 5 region of genes. All of the housekeeping genes are constitutively expressed and have CGIs which makes up 50% of all the CGIs and the other 50% of CGIs are associated with promoter activity of tissue specific genes. It is been shown that half of all human genes are associated with CGIs.

Genes without TATA boxes or initiator elements contain CGIs in upstream of the start site (ATG). Usually transcription of genes with promoter containing a TATA box or initiator element begins at a well defined initiation site. However, some genes are controlled by the state of CGIs methylation through the addition of a methyl group at position 5 on the cytosine changing it to 5-methylcytosine. DNA methylation is an epigenetic (”on” genes, refers to all modifications to genes other than changes in the DNA sequence itself) modification that occurs in some eukaryotes whereby CpG dinucleotides are methylated at the C5 position of cytosine. The methylation of the 5 regulatory regions of genes results in gene silencing. CGI is important in gene expression control through methylation where nonmethylated islands are found near active genes. Promoters are molecular modules, which are controlled by their surrounding DNA sequence and state. Methylation of a CGI at the promoter of a gene renders that gene inactive. This is why nonmethylated animal cells in tissue culture (vitro) becomes methylated and stops expressing certain proteins.

In the laboratory CGIs are identified by their susceptibility to restriction enzymes (found in bacteria, cuts double strand DNA by recognizing specific DNA patterns) that recognize CG sequences. The enzyme HpaII cleaves the sequence CCGG, but if the second is methlyated, the enzyme can no longer recognize the site. There are other enzymes that would cleave CG irrespective of the methylation state of the CG. There are also methylated DNA immunoprecipitation along with Nimbelgen array technology to detect CGIs in the genome. These arrays allow you to determine methylation of the promoter regions as well as within the genome, also compare differential methylation between cells, tissues, and tumor samples.

CpG dinucleotides tend to change to TpG/CpA and this is why its believed that in the human genome there is about 5% less CGIs. Methylated cytosines tend to turn into thymines because of spontaneous deamination (loss of NH2 group) this accounts for the less frequency of the CGIs in the genome.

CpG island aberrant hypermethylation is associated with different cancer, where many important genome maintanence genes are silenced through CGI methylation. Normally, tumor suppressor genes may be silenced by deletion (reflected in loss of heterozygosity) or by point mutations, but there is increasing evidence for a third mechanism CGIs methylation.