Assignment 4

N Kapil Chandra (MM14B022)

November 10, 2017

Images for Barabasi Albert algorithm

Figure 1: Initial Graph

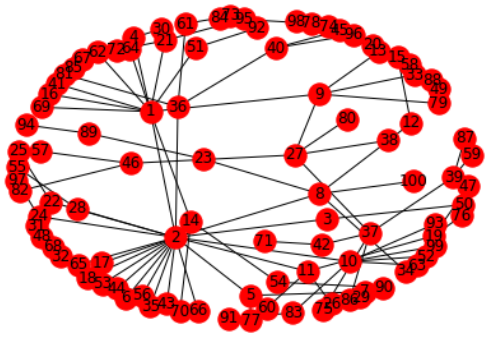

Figure 2: Final Graph with 100 nodes

Motivation

A couple of biological problems can be tackled using the algorithm going to discussed in this article. One of them is that if I was given a family of proteins and a database and was asked to search from the database for more proteins which can belong to the family of proteins, I can use regular expressions to solve it. But using something called 'Hidden Markov Model' turns out to much better than just using regular expressions and we are going to see why in the third section.[1]

Hidden Markov Model (HMM) can also be used for gene finding. A gene consists of two part which are repeated, namely, exons and introns, there are a few rules by which the sequence has to be made. A sequence cannot end with an intron and no two exons can be together. For transition of a gene to protein only the exons are used, introns are present for the stability of the gene. In gene finding several predictions of exons and introns are made which have to conform to the rules mentioned above. Gene finding is one of the most important steps in understanding genome of a species [3]. Earlier gene finding was done by conducting multiple painstaking experiments on living organism and results from the experiments were combined to create something called a genetic map which specifies the rough location of a gene relative to each other [3]. But with emergence of computers and algorithms this has become a computational problem. [3]

Another problem where it is used is in the location of 'CG-Islands'[2]. CG is very rarely occurring dinucleotide in a genome sequence. The reason being that C in this dinucleotide is easily methylated and methylated C has a tendency to mutate into T(cite). However, methylation of C is suppressed in CG-Islands, where CG appears relatively frequently. Locating these CG islands in a long genomic text is an important problem [2]. We are breifly going to discuss about Hidden Markov Models in this regard in the next section.

Hidden Markov Models (HMM)

A Markov process is a stochastic process that satisfies the property that the future state can be predicted only by using the present state without having any knowledge of the past. This is also sometimes called 'memorylessness' [5]. A hidden Markov Model is just a different version of Markov process and we will shortly see why it is 'hidden'.

I am taking a example from Wikipedia page of Hidden Markov Model to make the concept clear. This is called 'Urn problem' [5]. So the problem goes like this; there is an invisible genie sitting inside a room who has $X1$,$X2$,$X3$... urn which contain a known mix of balls $y1$,$y2$,$y3$... . The genie draws a ball with replacement from any urn of his choice and sends them out of the room though a conveyor belt. We can only see the ball but we don't know from which urn the ball came from. The choice of the urn is a random and depends on the choice of $(n-1)^{th}$ urn, this is why it is called a Markov process, as seen earlier.

The Markov process cannot be seen in this case and only the outcome which is the ball coming out can be seen, hence this is called a hidden Markov process. Figure 3 below shows how a model is constructed,

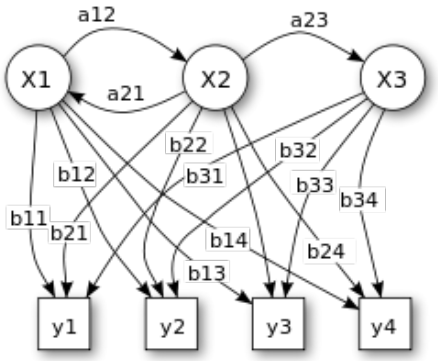

Figure 3: Markov Model [5]

In the above figure X is the state which in this case is urn and y is the outcome, that is, the ball. a is the state transition probability which in this case means the likelihood of genie changing from one urn to the other. b is the output probability which gives the probability of getting an outcome given the state. a and b are usually represented using matrices.

In this problem even if the person knows the composition of each urn it is not possible for him to say for sure which urn the genie used to draw the ball, he can say the likelihood of a particular ball coming from a particular urn though.

CG-Island problem

To understand HMMs better in the context of CG-islands problem let us first look at a generic scenario of using HMMs. We will look at the case of 'Fair bet casino' problem [2].

In this case there is a dealer from casino who flips a coin and I as a player must bet on the outcome. The dealer has two coins, one is biased and the other is fair. I don't know what coin he will be using and I know for a fact that for security reasons the dealer doesn't like to change coins so often. Given a sequence of outcomes from coin tosses I have to be able to guess what coin he is using because this will help me to win money. Assuming the biased coin is more biased towards getting heads on coin toss, if were to see a long sequence of heads in a row then it is more likely that he is using the biased coin and if I were to see a sequence containing almost equal number of heads and tails then

it is more likely that he is using a fair coin. But I can never be certain about it because getting a row of heads is also possible in a fair coin (why not?).

To understand better let us first consider a basic model[2] where the dealer does not change the coin. If the coin is fair $P^+(0) = P^-(1) = \frac{1}{2}$ and if the coin is biased $P^-(0) = \frac{1}{4}; P^-(1) = \frac{3}{4}$. If the sequence x is generated after the experiment then the following can be established,

$$P(x|faircoin) = \prod_{i=1}^{n} p^+(x_i) = \frac{1}{2^n}$$

$$P(x|biasedcoin) = \prod_{i=1}^{n} p^-(x_i) = \frac{3^k}{4^n}$$

where k is the number of heads obtained. If $P(x|faircoin) > P(x|biasedcoin)$ then the dealer most likely used a fair coin and vice-versa. Based on the formulation above we define something called a *log-odds ratio* whose importance will soon be realized. It is defined as follows:

$$log_2 \frac{P(x|faircoin)}{P(x|biasedcoin)} = \sum_{i=1}^{k} log_2 \frac{P^+(x_i)}{P^-(x_i)} = n - klog_2 3$$

So to find out what coin the dealer is using we just have to use this formula and determining. One näive way to solve this problem of CG-islands is to calculate log-odds ratio for sub sequences of particular length and whichever sub sequence gets positive ratio value will be a potential CG island. But the disadvantage is that we don't know how long a sub sequence we have to consider and due to overlapping we might classify the nucleotide differently.

To model this problem in a better way we use Hidden Markov Model. Here just like in the urn example discussed previously we don't know what coin the dealer is using but we know the outcome which is the coin side. For the 'Fair bet casino' problem the possible outcomes are H or T, the states are Fair (F) and Biased (B). The transition probabilities are, $a_{FF} = a_{BB} = 0.9$ and $a_{FB} = a_{BF} = 0.1$. The outcome probabilities are $e_F(0) = e_F(1) = \frac{1}{2}$ and $e_B(0) = \frac{1}{4}; e_B(1) = \frac{3}{4}$. Figure 4 shows the HMM for 'Fair bet casino' problem,

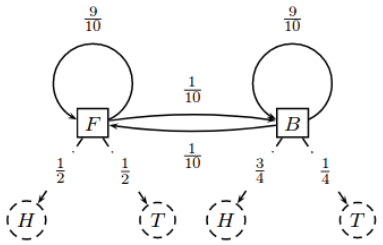

Figure 4: Hidden Markov Model for fair bet casino problem

For getting the total probability of a sequence we take probability of that state occurring from the previous state, multiply it with the probability of the outcome given the state and for multiply all of these for each part of the sequence. We only have the outcome at our disposal but we don't know the states. So given a sequence of outcomes we decide if a particular set of states is probable or not by calculating the probability as described above.

Why use HMM for searching protein sequences from database?

In this section we will discuss the advantage of using HMM for searching protein sequence from a database to fit in the family of proteins given [1]. Consider a DNA motif like in the figure 5

```
A C A - - - A T G
T C A A C T A T C
A C A C - - A G C
A G A - - - A T C
A C C G - - A T C
```

Figure 5: Illustration of sequences [1]

We could build a regular expression to match the sequences given in the figure 5 but the problem with regular expression is that it is equally likely to get 'A C A C - - A T C' and 'T G C T - - A G G' which should not be the case given the set of sequences given initially. To get the sequence right, we say that the A sits in the first position with a probability 0.8 and T sits there with a probability 0.2. Figure 6 keeps track of these kinds of probabilities and this is the model for this problem.

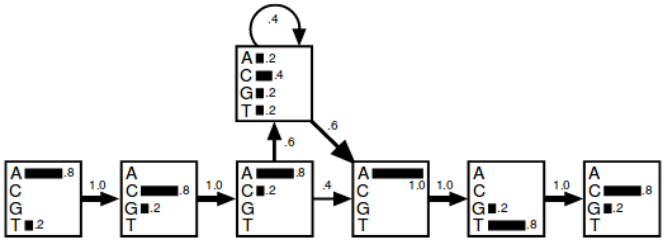

Figure 6: HMM for sequence [1]

Probabilities are calculated by just counting the number of occurrences of each letter in that position in the given set of sequences. Where there is an insertion, where the probability of each letter is found by counting all the occurrences of letters in this region of alignment. Probability of a sequence, just as before, is calculated by multiplying the probabilities of occurrence of each letter at its position. After calculating, we see that the probability of obtaining 'A C A C A T C' is 4.7×10^{-2} where as to get the sequence 'T G C T A G G' IS 2.3×10^{-5} which is 2000 times smaller than the former. In this way we can get a score for each sequence which tells how well the sequence fits in a family of proteins. Hence using HMM is far better than just using regular expressions for the finding proteins from a database belonging to a family of proteins.

Similar advantages are seen in the case of gene finding as well where there are exons and introns. The prediction should follow the rules of the sequences which can be incorporated using probabilities in the Hidden Markov Model. The sequences that don't follow a rule have very low probability as compared to sequences following the rules which makes the prediction of correct sequences more probable.

References

[1] An Introduction to Hidden Markov Models for Biological Sequences, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.85.7972&rep=rep1&type=pdf#page=3&zoom=auto,-82,772

[2] An Introduction to Bioinformatics Algorithms, http://bioinformaticsinstitute.ru/sites/default/files/an_introduction_to_bioinformatics_algorithms_-_jones_pevzner.pdf

[3] https://en.wikipedia.org/wiki/Gene_prediction

[4] https://en.wikipedia.org/wiki/Hidden_Markov_model

[5] https://en.wikipedia.org/wiki/Markov_chain