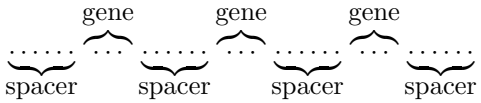
1

1 Hidden Markov Models

Genes

- A genome is divided into chromosomes, which are long chains of DNA.
- Only sections of a chromosome get transcribed.



2

Where are the Genes?

- The problem is that it is not obvious from looking at DNA where these genes are located along the chromosome.
- Both spacer and genes just **look like** random words over `A, C, T, G`.
- If we did know, it would go a long way to determining what proteins could be generated!

3

CpG-Islands

- Each individual nucleotide should occur by chance every 4 nucleotides.
- Each pair of nucleotides, called a *dinucleotide* should occur by change every $1/(4^2) = 1/16$ dinucleotides.
- The least frequent dinucleotide in many genomes is `CG`.
- In the human genome, it only occurs at 20% of the frequency that should have occurred by chance.

4

CpG-Islands

- The `C` of `CG` is easily methylated, which adds a methyl group to the 5' carbon without altering base pairing properties.
- This methylated `C` tends to mutate into `T`.
- The methylation is often suppressed around genes in areas called *CpG-Islands*. In these areas, `CG` appears at a normal rate.
- These Islands often span from position -1500 to $+500$ of human genes.

5

CpG-Islands

- We would like to be able to predict which areas are `CpG`-Islands. Unfortunately, there are not any obvious markers which indicate where they start and finish.
- Thus, it is valuable to find `CpG`-Islands in order to find genes.
- A popular technique for solving this problem is the use of *Hidden Markov Models*.

6

Hidden Markov Models

- Hidden Markov Models (HMMs) are a popular machine learning tool.
- They have been used extensively for such tasks as natural language processing.
- HMMs are frequently used to look for patterns in biological sequences.
- Like other methods of machine learning, we have a training set, which trains the HMM, and at that point, we apply it to a test sample.

7

Hidden Markov Models

We can draw them as graphs, with some adjustments:

1. we call the vertices *states* and we call the edges *transitions*,
2. we will attach probabilities to it somehow.

8

Probability Theory

We would like to be able to talk about the probabilities of events happening.

Definition 1. If a and b are real numbers, then $[a, b]$ is the set of all real numbers between a and b (inclusive).

Definition 2. A *finite sample space* is a finite set S and a function $P : S \mapsto [0, 1]$ such that $\sum_{a \in S} P(a) = 1$.

Intuitively, $P(a)$ represents the probability of a occurring. If we add up all the possibilities, we should get 1 since one of the possibilities must happen.

9

Probability Theory

Definition 3. An *event* E is any subset of the sample space S. We extend P to sets by $P(E) = \sum_{a \in E} P(a)$.

Example 4. If we wanted to represent the tossing of a fair coin once, then we could use a sample space $S = \{H, T\}$, representing heads and tails, and assign $P(H) = .5$ and $P(T) = .5$. Then $P(S) = 1$.

10

Probabity Theory

Example 5.
- Say we perform an experiment whereby we flip a coin 3 times. The sample space of the experiment is $S = \{HHH, HHT, HTH, HTT, THH, THT, TTH, TTT\}$.
- If all outcomes in the sample space are equally likely, then $P(a) = 1/8$, for every $a \in S$.
- If $E = \{HHH, TTT\}$ is an event, then $P(E) = 2/8 = 1/4$.

Definition 6. If A and B are two events, then $P(A \mid B)$ is the probability of A given B (in math $P(A \cap B)/P(B)$).

Casino!

Finding `CpG`-Islands is very similar to the following example.

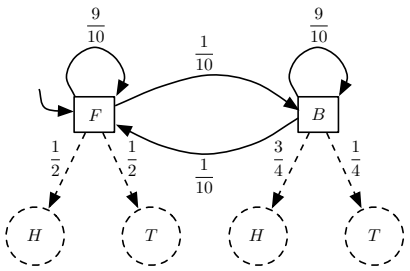
- Let's say there is a casino, and the dealer has two coins.
- The first coin is a fair one in which heads appears with probability .5 and tails appears with probability .5.
- The second coin is a biased coin which gives heads with probability .75 and tails with probability .25.
- The dealer only changes coins with a probability of .1. Thus, he keeps the same coin with a probability of .9.

In this example, a `CpG`-island is similar to the fair coin, and an area other than a `CpG`-island is similar to the biased coin.

Hidden Markov Models

Definition - informal
Informally, an HMM is a graph, where we have states, a transition between each pair of states with a probability between 0 and 1 associated with it, and for every state, a probability of outputting some letter.

An Example



- The square symbols represent the possible states of the HMM. Either the dealer is using the fair coin or the biased coin.

- The circles represent the possible "output symbols".
- If the dealer is using the fair coin, the model should output heads and tails with equal frequency.
- If he or she is using the biased coin, it should output heads with probability .75 and tails with probability .25.

14

Hidden Markov Model

Definition 7. A HMM M consists of the following components:

- An alphabet Σ of symbols (such as $\{A, C, T, G\}$),
- A set of states Q,
- A $|Q| \times |Q|$ transition[1] matrix U where the entry at position (k, l) is the probability of changing from the k^{th} state to the l^{th} state,
- A $|Q| \times |\Sigma|$ matrix E where the entry at position (k, l) is the probability of outputting the l^{th} letter from the k^{th} state.
- A $|Q| \times 1$ matrix (or just a vector) I where the entry at position k represents the probability of starting in state k.

15

Hidden Markov Model

- Initially, we pick some starting state according to I.
- At each step, we output a letter with some probability according to E.
- Then we switch to some new state with some probability according to U.

16

Example

For this example our alphabet is $\{H, T\}$ and our states are $\{F, B\}$.

Our transition matrix is

	F	B
F	.9	.1
B	.1	.9

17

Our output matrix is

[1] $|Q|$ is the number of elements in Q.

	H	T
F	.5	.5
B	.75	.25

.

Let's say that the dealer is equally likely to start with either the fair or biased coin. Thus, our initial matrix is

F	.5
B	.5

.

18

- Let's say that the states we traverse are $\pi = FFBBBBBFFFFF$ and the output we see is $x = THHHTHHTTHHT$.
- Let π_i be the i^{th} character (state) of π and let x_i be the i^{th} character of x.

19

Consider the following:

$$\begin{array}{c} x \\ \pi \\ P(x_i \mid \pi_i) \\ P(\pi_i \mid \pi_{i-1}) \end{array} = \begin{pmatrix} T & H & H & H & T & H & H & T & T & H \\ F & F & B & B & B & B & B & F & F & F \\ .5 & .5 & .75 & .75 & .25 & .75 & .75 & .5 & .5 & .5 \\ .5 & .9 & .1 & .9 & .9 & .9 & .9 & .1 & .9 & .9 \end{pmatrix}$$

In column i, we get the i^{th} character of x, the i^{th} character of π, the probability of outputting x_i from state π_i and the probability of switching from state π_{i-1} to state π_i.

To calculate the probability of this happening, we multiply $(.5 \cdot .5)(.5 \cdot .9)(.75 \cdot .1)(.75 \cdot .9)(.25 \cdot .9)(.75 \cdot .9)(.75 \cdot .9)(.5 \cdot .1)(.5 \cdot .9)(.5 \cdot .9)$.

20

HMMs

In general, we get the following formula for the probability that a output sequence x of length n was generated by the path π:

$$P(\pi_1)P(x_1 \mid \pi_1) \cdot \prod_{i=2}^{n} P(x_i \mid \pi_i)P(\pi_i \mid \pi_{i-1}),$$

where $p(\pi_1)$ is the probability[2] of starting in state π_1.

21

[2]The term $\prod_{i=1}^{n} a_i = a_1 \cdot a_2 \cdot \cdots \cdot a_n$.

CpG-islands

- With the CpG-Island problem, we are given the genome sequence, and our job is to try to predict which areas are CpG-Islands and which areas are not.
- Analgously, with this problem, we are watching the dealer, and all we can see is the sequence of heads and tails that comes up.
- Our job is to try to predict which coins he has at what time.

22

CpG-Islands

- In the context of this problem, that means we are given the *output* of the model (the letters) and our job is to try to predict the *states* that the model was in.
- Thus, we know the output and we do not know the states.
- This is why we call it a *Hidden* Markov Model.
- Given an HMM M and a sequence x generated by M, we would like to find a path π that traverses the states of M and that has the *maximimum* probability of generating x.

23

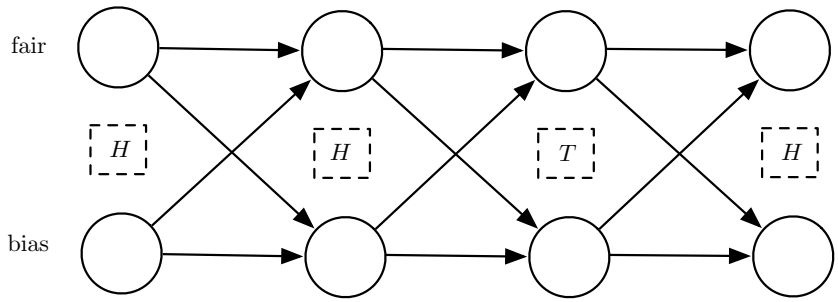
Decoding Algorithm

- We will study an algorithm that solves this problem!
- Given an HMM M and a sequence x generated by M, this algorithm will find the maximum probability of some sequence that traverses the states of M generating x.
- Moreover, we can then trace through this algorithm to determine a path that achieves this maximum probability.
- This can then be used to find the sections with the highest probability of being CpG-Islands; e.g. when we are most probably in "fair" state rather than "biased" state.

24

An Example

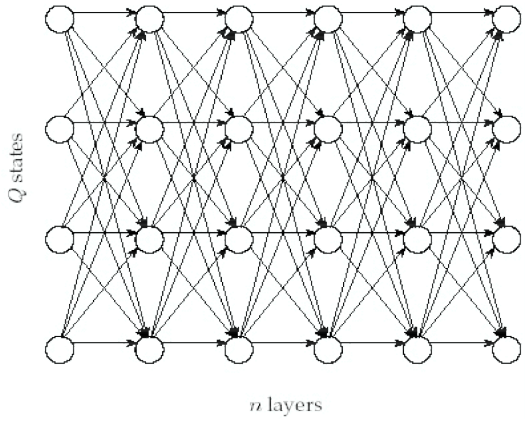
- Let's say we want to find the path that traverses the states of our HMM above and that maximizes the generation of the sequence $HHTH$.
- We first need to start out by making a graph that looks like an array of $|Q| \times n$ vertices, where n is the length of the HMM output. In this case it looks like 2×4 array.
- Each vertex has an edge from every vertex in the previous column.



25

An Example

In general, if the sequence of letters which is generated by the HMM is $x_1 \cdots x_n$, $x_i \in \Sigma, 1 \leq i \leq n$, then we start out with this more general diagram.



26

Example Continued

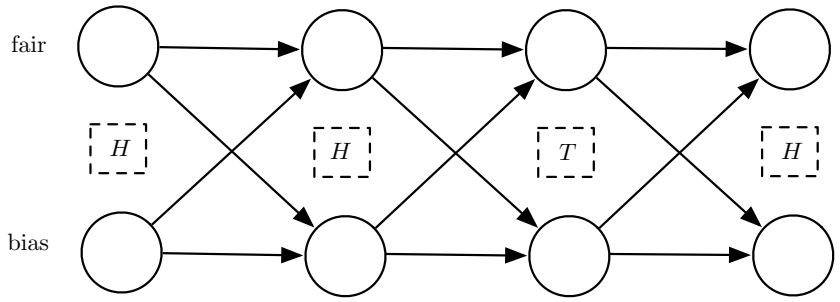
- What we are going to do is, for each vertex in our graph, calculate a weight.
- Assume the output is $x_1 \cdots x_n$.
- If the vertex is at position (i, j), the weight we calculate will represent the maximum probability of any path that reads $x_1 \cdots x_j$ and ends up in the i^{th} state.

27

picture taken from www.bioalgorithms.info

Example Continued

For example, the weight we calculate for the vertex at position $(1, 2)$ will be the maximum probability of any path that reads HH and ends up in the fair state.



28

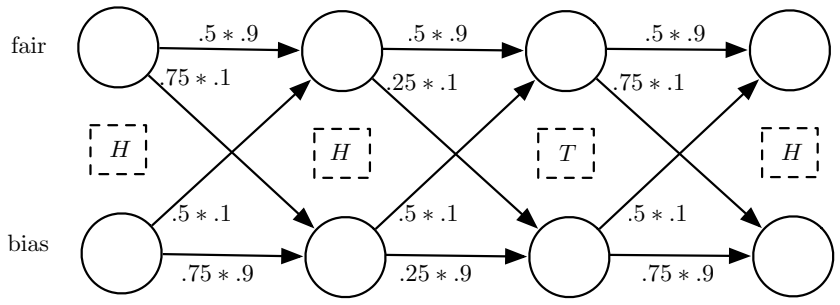
First Step

We will do this in steps:

- The first step is to associate a weight with each edge.
- The weight on edge (k, i) to $(l, i + 1)$ is equal to the probability of switching from the k^{th} state to the l^{th} state multiplied by the probability of outputting x_{i+1} from the l^{th} state.

29

For example, the edge from the vertex at position $(2, 2)$ to position $(1, 3)$ has weight which is the probability of switching from the bias to fair state $(.1)$ times the probability of outputting a T from the fair state $(.5)$.

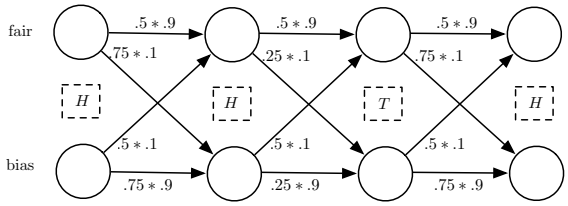


30

Algorithm Continued

More generally, as we have called our transition matrix U and our output matrix E, the weight on the edge from (k, i) to $(l, i + 1)$ is equal to $E(l, i + 1) \cdot U(k, l)$.

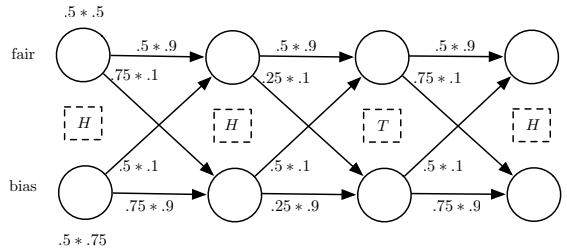
Next, we will calculate our desired weight on the vertices. We start with the first column, then the second column until the n^{th} column. For each column, will proceed from the top towards the bottom.



31

Example Continued

- The weights on the vertices in the first column are calculated differently from those of all other columns.
- If we are calculating the weight for the i^{th} state, we multiply the probability of starting in that state (according to the vector I) times the probability of outputting the first symbol from that state.
- For example, the weight on the vertex at position $(2, 1)$ is the probability of starting in the bias state (.5) times the probability of flipping heads from the bias state.



32

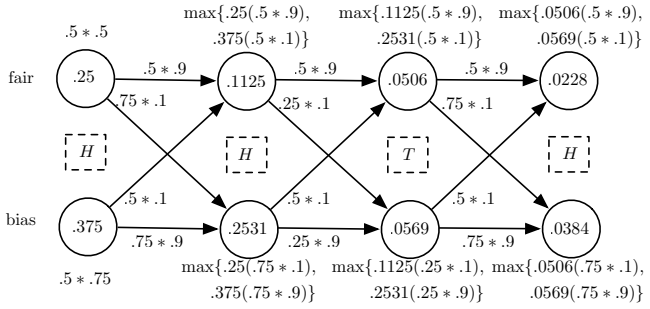
To calculate the weight on every other vertex (i, j), we use the following procedure:

1. we examine the set of all vertices that have an edge into (i, j).
2. For each of these, we multiply the weight on that vertex by the weight on the edge joining it to (i, j).
3. We then take the maximum of all these products.

This becomes the weight on (i, j).

That is, the weight on vertex (i, j) is equal to $\max_{k \in Q}\{\text{weight on vertex } (k, j-1) \cdot \text{weight on edge between}(k, j-1) \text{ and } (i, j)\}$.

33

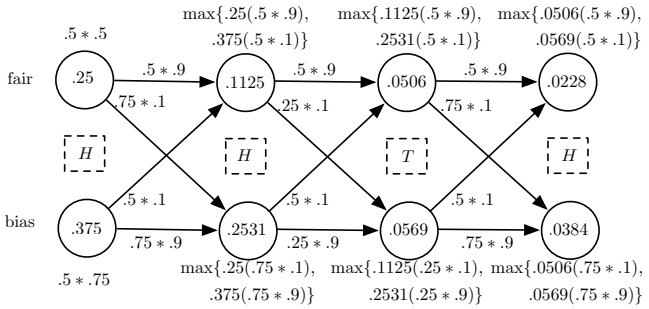


Example 8. For example, the weight on the vertex $(1, 2)$ is equal to the maximum the following two products:

1. the weight on the vertex $(1, 1)$ times the weight of the connecting edge,
2. the weight on the vertex $(2, 1)$ times the weight on the connecting edge.

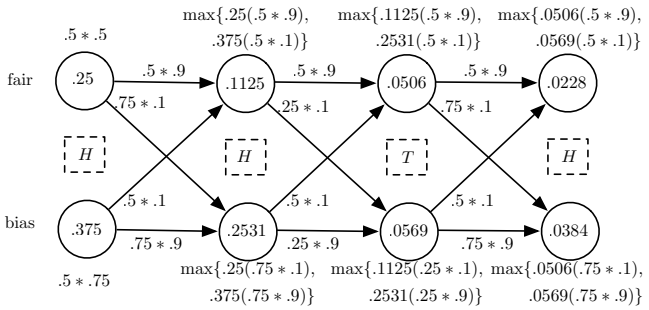
34

- Notice that when we calculate the maximum of the two products, we are taking the maximum of $(.5 \cdot .5)(.5 \cdot .9)$ and $(.5 \cdot .75)(.5 \cdot .1)$.
- The first corresponds to the path FF while the second corresponds to the path BF. We have indeed calculated the maximum probability of reading the first two symbols and ending in state F.



35

- When we are calculating the maximum at vertex $(1, 3)$, we do not need to try every possible path, since we have already calculated the maximum after reading the first two characters.
- We can simply multiply the weights in the second column by the edges connecting them to $(1, 3)$.



36

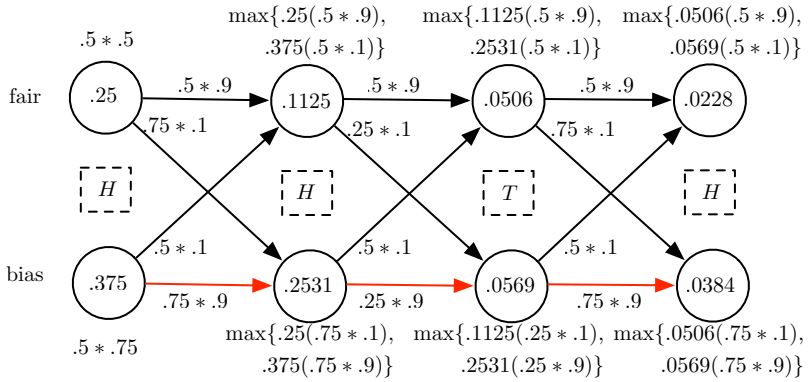
- Our maximum probability is the highest probability in the last column.
- In this case, it is .0384.
- There is a path which can output $HHTH$ with that probability.
- But what is that path?

37

- We can determine the magic path by starting at the highest probability in the last column and working our way leftwards in the graph until we hit the first row.
- If the vertex of highest weight in the last column is at position (i, j) of the graph, then the i^{th} state is the last vertex in the best path.
- We then continue to find the rest of the path by looking at which vertex, which connects to this vertex, achieved the maximum in the vertex weight calculation.
- That is the second last state of the path.

38

Here is our final graph, with the optimal path marked in red.



This means the path of the HMM that achieved the highest probability is $BBBB$.

39

- conclusion: if we see the sequence $HHTH$, it is most probable situation is that the dealer started with a biased coin, and used the biased coin for the subsequent three tosses

40

Probabilities?

- We never described how the probabilities were calculated!
- We need to determine the probabilities of switching from one state to the next and also the probabilities of outputting letters from given states.
- Like we said earlier, HMMs are a machine learning technique.
- We can learn the probabilities from a training set.

41

Probabilities?

- There are different heuristic ways of doing this.
- If we already know that a path $\pi_1 \cdots \pi_n$ corresponds to observed states $x_1 \cdots x_n$, then we can take this into account.
- If $sw(k, l)$ is the number of transitions from state k to l, then we can calculate the probability of switching from state k to l as
$$\frac{sw(k,l)}{\sum_{q \in Q} sw(k,q)}.$$
- We could calculate the probability of outputting a given letter from a given state in a similar fashion.

42