1 Introduction - HMM definition

In the last lecture we saw the definition of HMM:

- $S$ - set of states (start, $1..k$, end).
- $\Sigma$ - set of output chars.
- $\tau$ - transition probability - $\tau$ is the probability of moving from state $s$ to state $s'$.
- $\rho$ - emission probability - $\rho_{sx}$ is the probability of $x$ being emitted from state $s$.

We have previously described a profile-HMM. This is a state machine consisting of three series of nodes. Each node-trio corresponds to a single position in the alignment from which the profile was built. This machine defines probabilities of trajectories, by the multiplication of the probabilities of passing in the trajectory path. In this machine, the conditional probability of future states, given the present state and all past states, depends only on the current state. Let $H_i$ be a random variable that represents the state at time $i$.

Let $X_i$ be a random variable that represents the emission at time $i$.

**Definition 1.1** A process $H$ is a Markov Process if $\Pr(H_{i+1}|H_1, \ldots, H_i) = \Pr(H_{i+1}|H_i)$, i.e., the future is independent of the past given the present.

**Definition 1.2** A pair of processes $X, H$ constitutes a Hidden Markov Model (HMM) if:

- $H$ is a Markov Process.
- $\Pr(x_i|x_1, \ldots, x_n, H) = \Pr(x_i|H_i)$ i.e. $X$ is i.i.d. observation sequence.

**Definition 1.3** $H_0 = \text{start}$ and $H_{n+1} = \text{end}$.

**Example 1.4** The machine at Figure 1, has 2 states $A,B$ and a transition probability between all the states.

The emission probability of this machine is:

- $\rho_1(A) = 0.5$.
- $\rho_1(B) = 0.5$.
- $\rho_2(A) = 0.9$.
- $\rho_2(B) = 0.1$

An example of a run on this machine is:

<table>
<thead>
<tr>
<th>random variable \ Time</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H$</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>end</td>
</tr>
<tr>
<td>$X$</td>
<td></td>
<td></td>
<td>A</td>
<td>B</td>
<td>B</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>B</td>
<td>A</td>
</tr>
</tbody>
</table>


2 The Three Problems of HMM

There are three key problems of HMMs. These are:

1. Likelihood - Given an observed sequence and a HMM model, we would like to evaluate the probability that the observed sequence was generated by the model:

   \[ \Pr(x_1, \ldots, x_n) = \sum_{H_1} \ldots \sum_{H_n} \Pr(H_1, \ldots, H_n, x_1, \ldots, x_n) \]  

   (1)

2. Inference - Given a specific sequence \( X = x_1, \ldots, x_n \), we would like to evaluate the probability that at time \( i \) the machine was at state \( H_i \).

   \[ \Pr(H_i|x_1, \ldots, x_n) = \frac{Pr(H_i, x_1, \ldots, x_n)}{Pr(x_1, \ldots, x_n)} \]  

   (2)

3. Inference the most probable state sequence - Given a specific observation sequence \( X = x_1, \ldots, x_n \), we would like to find the series of states that maximizes the chance to get the observed sequence of \( X \)'s.

   \[ \arg \max_{H_1, \ldots, H_n} \Pr(x_1, \ldots, x_n, H_1, \ldots, H_n) \]  

   (3)

3 First problem - likelihood

Given an observation sequence and a HMM model, we would like to evaluate the probability that the observed sequence \( X = x_1, \ldots, x_n \) was generated by the model.

\[ \Pr(x_1, \ldots, x_n) = ? \]

We saw a naive way to solve the problem using the total probability theorem:

\[ \Pr(x_1, \ldots, x_n) = \sum_{H_1} \ldots \sum_{H_n} \Pr(H_1, \ldots, H_n, x_1, \ldots, x_n) \]

However, this computation is impossible in practice since its time complexity is \( O(n|S|^n) \) where \( |S| \) is the number of states. The problem can be overcome by a dynamic programming solution. We will see two different approaches that will give us the equation that we look for:

1. Algebraic manipulation.
2. Probability reasons.
3.1 Algebraic manipulation

Using HMM properties and some algebraic manipulations we get:

\[
\Pr(X, \bar{H}) = \prod_{1}^{n+1} \Pr(H_i|H_{i-1}) \prod_{1}^{n} \Pr(x_i|H_i) \Rightarrow \\
= \sum_{H_1} \cdots \sum_{H_n} \prod_{i=1}^{n+1} \Pr(H_i|H_{i-1}) \Pr(x_i|H_i) = \sum_{H_1} \cdots \sum_{H_n} \prod_{i=1}^{n+1} \Pr(H_i|H_{i-1}) \Pr(x_n|H_n)
\]

The last term is defined as:

\[
B_{n-1}(H_{n-1}) = \sum_{H_n} \prod_{i=n}^{n+1} \Pr(H_i|H_{i-1}) \Pr(x_n|H_n)
\]

In the same way we define \(B_{n-2}(H_{n-2})\):

\[
\sum_{H_1} \cdots \sum_{H_{n-1}} \prod_{i=1}^{n-1} \Pr(H_i|H_{i-1}) \Pr(x_i|H_i) B_{n-1}(H_{n-1}) = \sum_{H_1} \cdots \sum_{H_{n-2}} \prod_{i=1}^{n-2} \Pr(H_i|H_{i-1}) \Pr(x_i|H_i) \sum_{H_{n-1}} \prod_{i=n}^{n+1} \Pr(H_i|H_{i-1}) \Pr(x_n|H_n) B_{n-1}(H_{n-1})
\]

\[
B_{n-2}(H_{n-2}) = \sum_{H_{n-1}} \Pr(H_{n-1}|H_{n-2}) \Pr(x_{n-1}|H_{n-1}) B_{n-1}(H_{n-1})
\]

The recursion equation is:

\[
B_1(H_1) = \sum_{H_{i+1}} \Pr(H_{i+1}|H_i) \Pr(x_{i+1}|H_{i+1}) B_{i+1}(H_{i+1}) \\
B_n(H_n) = \Pr(H_{n+1} = \text{end}|H_n) \\
\vdots \\
B_0(H_0 = \text{start}) = \Pr(x_1, \cdots, X_n)
\]

3.2 Probability reasons

Definition 3.1

\[
B_i(H_i) = \sum_{H_{i+1}} \Pr(H_{i+1}|H_i) \Pr(x_{i+1}|H_{i+1}) B_{i+1}(H_{i+1})
\]

Theorem 3.2 \(B_i(H_i)\) is the probability of the observed sequence from \(x_n\) down-to and including \(x_{i+1}\), meaning:

\[
B_i(H_i) = \Pr(x_{i+1}, \cdots, x_n|H_i)
\]
**Proof:** $B_n(H_n)$ is the probability to end the sequence from the state $H_n$:

$$B_n(H_n) = \Pr(H_{n+1} = \text{end}|H_n)$$

In the general case, we look at:

$$\Pr(x_{i+1}, \ldots, x_n|H_i) = \sum_{H_{i+1}} \Pr(x_{i+1}, \ldots, x_n, \$, H_{i+1}|H_i)$$

$$= \sum_{H_{i+1}} \Pr(x_{i+1}|x_{i+2}, \ldots, x_n, \$, H_i, H_{i+1}) \Pr(H_{i+1}|H_i) \Pr(x_{i+2}, \ldots, x_n, \$|H_i, H_{i+1})$$

the first row following because of the law of total probability, the second row coming from the chain rule. From the HMM definition:

$$\Pr(x_{i+1}|x_{i+2}, \ldots, x_n, \$, H_i, H_{i+1}) = \Pr(x_{i+1}|H_i, H_{i+1}) = \Pr(x_{i+1}|H_{i+1})$$

$$\Pr(x_{i+2}, \ldots, x_n, \$|H_i, H_{i+1}) = \Pr(x_{i+2}, \ldots, x_n, \$|H_{i+1})$$

Using these two equation we get:

$$\Pr(x_{i+1}, \ldots, x_n|H_i) = \sum_{H_{i+1}} \Pr(x_{i+1}|H_{i+1}) \Pr(H_{i+1}|H_i) \Pr(x_{i+2}, \ldots, x_n, \$|H_{i+1})$$

$$= \sum_{H_{i+1}} Pr(x_{i+1}|H_{i+1}) Pr(H_{i+1}|H_i) B_i+1(H_{i+1})$$

$$= B_i(H_i)$$

**Corollary 3.3** $B_0(H_0 = \text{start})$ will give the final result of the problem:

$$B_0(H_0 = \text{start}) = \Pr(x_1, \ldots, x_n|\$$)

Using all these definitions we define the backward algorithm.

### 3.3 The backward algorithm

**Initialization:** $B_n(H_n) = \Pr(H_{n+1} = \text{end}|H_n)$.

**Recursion:** ($i = n-1, \ldots, 0$): $B_i(H_i) = \sum_{H_{i+1}} \Pr(H_{i+1}|H_i) \Pr(x_{i+1}|H_{i+1}) B_{i+1}(H_{i+1})$

**Termination:** $\Pr(x_1, \ldots, x_n) = B_0(H_0 = \text{start})$

Notice that we define $H_{n+1} = \text{end}$ and $H_0 = \text{start}$.

This is a dynamic programming algorithm and we can now fill the following matrix:

<table>
<thead>
<tr>
<th>State \ Time</th>
<th>\ldots</th>
<th>n-2</th>
<th>n-1</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>$B_{n-2}(H_{n-2} = 0)$</td>
<td>$B_{n-1}(H_{n-1} = 0)$</td>
<td>$B_n(H_n = 0)$</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>$B_{n-2}(H_{n-2} = 1)$</td>
<td>$B_{n-1}(H_{n-1} = 1)$</td>
<td>$B_n(H_n = 1)$</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>$B_{n-2}(H_{n-2} = 2)$</td>
<td>$B_{n-1}(H_{n-1} = 2)$</td>
<td>$B_n(H_n = 2)$</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
</tr>
</tbody>
</table>

- This matrix can be efficiently calculated (fill the matrix from right to left).
- This should be calculated separately for every output sequence $X = x_1, \ldots, x_n$. 


• Time complexity - for each cell in the table, or in other words for each time \(i\) and for each state \(j\) the algorithm’s calculate \(B_i(H_i = j)\). Iterating over all the table costs \(O(n \times |S|)\) where \(|S|\) is the number of states, calculation of each \(B_i(H_i = j)\) costs \(|S|\) since it passes over all states from time \(i + 1\). The total cost is \(O(n \times |S|^2)\).

• If the number of edges in the graph is small we could calculate the sum over \(u \in S\) more efficiently. When calculating \(B_i(v)\) we sum only over the states in \(S\) that have an edge to \(v\). In such a graph, the complexity is reduced to \(O(n \times E)\), where \(E\) is the number of edges.

• \(B_0(H_0 = \text{start})\) give us \(\Pr(x_1, \ldots, x_n)\).

• Memory complexity - if all what we need is the final result, the cost is just \(O(2 \times |S|)\) because the algorithm only uses the last column and there is no need to remember all the table. But, if we need all the table, for reasons that we will see later, the cost will be linear - \(O(n \times |S|)\).

**Example 3.4** Given the sequence \(X = ABBA\$\) and the HMM model from Example 1.4, we would like to evaluate \(\Pr(X)\) using the backward algorithm. We need to find \(B_0(\text{start})\), and for this to fill the matrix:

\[
\begin{array}{c|cccc}
\text{State} & 1 & 2 & 3 & 4 \\
\hline
\text{\$\text{(end)}} & A & B & B & A \\
\end{array}
\]

\[B_4(H_4 = 1) = \Pr(H_5 = \text{end}|1) = 0\]
\[B_4(H_4 = 2) = \Pr(H_5 = \text{end}|2) = 0.1\]

**update the matrix:**

\[
\begin{array}{c|cccc}
\text{State} & 1 & 2 & 3 & 4 \\
\hline
\text{\$\text{(end)}} & A & B & B & A \\
1 & 0 & 0 & 0.1 \\
2 & 0 & 0.45 & 0.045 \\
\end{array}
\]

\[B_3(H_3 = 1) = \Pr(H_4 = 1|1) \Pr(A|1)B_4(H_4 = 1) + \Pr(H_4 = 2|1) \Pr(A|2)B_4(H_4 = 2) =
= 0 + 0.3 \times 0.9 \times 0.1 = 0.027\]
\[B_3(H_3 = 2) = \Pr(H_4 = 1|2) \Pr(A|1)B_4(H_4 = 1) + \Pr(H_4 = 2|2) \Pr(A|2)B_4(H_4 = 2) =
= 0 + 0.5 \times 0.9 \times 0.1 = 0.045\]

**update the matrix:**

\[
\begin{array}{c|cccc}
\text{State} & 1 & 2 & 3 & 4 \\
\hline
\text{\$\text{(end)}} & A & B & B & A \\
1 & 0.072 & 0.27 & 0 \\
2 & 0.045 & 0.1 \\
\end{array}
\]

\[B_2(H_2 = 1) = \Pr(H_3 = 1|1) \Pr(B|1)B_4(H_4 = 1) + \Pr(H_3 = 2|1) \Pr(B|2)B_4(H_3 = 2) =
= 0.7 \times 0.5 \times 0.027 + 0.3 \times 0.1 \times 0.045 = 0.0108\]
\[B_2(H_2 = 2) = \Pr(H_3 = 1|2) \Pr(B|1)B_4(H_4 = 1) + \Pr(H_3 = 2|2) \Pr(B|2)B_4(H_3 = 2) =
= 0.4 \times 0.5 \times 0.027 + 0.5 \times 0.1 \times 0.045 = 0.00765\]

**update the matrix:**

\[
\begin{array}{c|cccc}
\text{State} & 1 & 2 & 3 & 4 \\
\hline
\text{\$\text{(end)}} & A & B & B & A \\
1 & 0.0108 & 0.27 & 0 \\
2 & 0.00765 & 0.045 & 0.1 \\
\end{array}
\]

5
\[ B_1(H_1 = 1) = \Pr(H_2 = 1 | 1) \Pr(B_1 | B_2) \Pr(H_2 = 1) + \Pr(H_2 = 2 | 1) \Pr(B_1 | B_2) \Pr(H_2 = 2) = 0.7 \times 0.5 \times 0.00108 + 0.3 \times 0.1 \times 0.00765 = 0.0040095 \]

\[ B_1(H_1 = 2) = \Pr(H_2 = 1 | 2) \Pr(B_1 | B_2) \Pr(H_2 = 1) + \Pr(H_2 = 2 | 2) \Pr(B_1 | B_2) \Pr(H_2 = 2) = 0.4 \times 0.5 \times 0.0054 + 0.5 \times 0.1 \times 0.00765 = 0.0014625 \]

update the matrix:

<table>
<thead>
<tr>
<th>State \ Time</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.0040095</td>
<td>0.0108</td>
<td>0.027</td>
<td>0.00108</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>0.0014625</td>
<td>0.00765</td>
<td>0.045</td>
<td>0.1</td>
<td></td>
</tr>
</tbody>
</table>

The final result is:

\[ \Pr(ABBA) = \Pr(H_1 = 1 | \text{start}) \Pr(A | 1) \Pr(B_1 | H_1 = 1) + \Pr(H_1 = 2 | \text{start}) \Pr(A | 1) \Pr(B_1 | H_1 = 2) \]

\[ = 1 \times 0.5 \times 0.0040095 + 0 = 0.00200475 \]

### 3.4 Forward Algorithm

**Definition 3.5**

\[ F_{i+1}(H_{i+1}) = \sum_{H_i} F_i(H_i) \Pr(H_{i+1} | H_i) \Pr(x_{i+1} | H_{i+1}) \]

**Theorem 3.6** \( F_{i+1}(H_{i+1}) \) is the probability of the observed sequence from \( x_1 \) up-to and including \( x_{i+1} \), meaning:

\[ F_{i+1}(H_{i+1}) = \Pr(x_1, \ldots, x_{i+1}, H_{i+1}) \]

**Proof:** \( F_0(H_0) \) is the probability to start the sequence

\[ F_0(\text{start}) = 1 \]

in the general case, we look at:

\[ \Pr(x_1, \ldots, x_{i+1}, H_{i+1}) = \sum_{H_i} \Pr(x_1, \ldots, x_i | H_i) \Pr(H_{i+1} | x_1, \ldots, x_i, H_i) \Pr(x_{i+1} | x_1, \ldots, x_i, H_i) \]

\[ = \sum_{H_i} F_i(H_i) \Pr(H_{i+1} | H_i) \Pr(x_{i+1} | H_{i+1}) \]

\[ = F_{i+1}(H_{i+1}) \]

the first row following from the law of total probability and the chain rule, the second row coming from the HMM properties as we saw above.

**Corollary 3.7** \( F_{n+1}(H_{n+1} = \text{end}) \) will give the final result of the problem:

\[ F_{n+1}(H_{n+1} = \text{end}) = \Pr(x_1, \ldots, x_n) \]
4 Second problem - inference

4.1 Why would we need two different algorithms to answer the same question?

We know that:
\[ F_{i+1}(H_{i+1}) = \Pr(x_1, \ldots, x_i, H_i) \]
\[ B_i(H_i) = \Pr(x_{i+1}, \ldots, x_n | H_i) \]

Given a beginning of a specific observation sequence \( x_1, \ldots, x_i \), we would like to know the state \( H_i \), meaning:
\[ \Pr(H_i | x_1, \ldots, x_i) \]

From the definition of conditional probability:
\[ \Pr(H_i | x_1, \ldots, x_i) = \frac{\Pr(H_i, x_1, \ldots, x_i)}{\Pr(x_1, \ldots, x_i)} \]

Since
\[ \Pr(H_i, x_1, \ldots, x_i) = F_{i+1}(H_{i+1}) \]
\[ \Pr(x_1, \ldots, x_i) = \sum_{H_i} \Pr(x_1, \ldots, x_i, H_i) = \sum_{H_{i+1}} F_{i+1}(H_{i+1}) \]

we get:
\[ \Pr(H_i | x_1, \ldots, x_i) = \frac{F_{i+1}(H_{i+1})}{\sum_{H_{i+1}} F_{i+1}(H_{i+1})} \]

In the same way, we can find \( H_i \) given the end of a specific observation sequence - \( x_i, \ldots, x_n \) using \( B_i(H_i) \).

4.2 Second problem

Now, we can solve the second question that we asked above: Given a specific sequence \( X = x_1, \ldots, x_n \), we would like to evaluate the probability that at time \( i \) the machine was at state \( H_i \).

\[ \Pr(H_i | x_1, \ldots, x_n) = \frac{\Pr(H_i, x_1, \ldots, x_n)}{\Pr(x_1, \ldots, x_n)} \]

We know how to find \( \Pr(x_1, \ldots, x_n) \) using the Backward or the Forward algorithms. From the chain rule:
\[ \Pr(H_i, x_1, \ldots, x_n) = \Pr(H_i, x_1, \ldots, x_i) \Pr(x_i, \ldots, x_n | x_1, \ldots, x_i, H_i) \]

From the i.i.d. assumption we get:
\[ \Pr(x_i, \ldots, x_n | x_1, \ldots, x_i, H_i) = \Pr(x_i, \ldots, x_n | H_i) = B_i(H_i) \]

So, the answer to the second problem is:
\[ \Pr(H_i | x_1, \ldots, x_n) = \frac{F_i(H_i) B_i(H_i)}{\Pr(x_1, \ldots, x_n)} = \frac{F_i(H_i) B_i(H_i)}{\sum_{H_i'} F_i(H_i') B_i(H_i')} \]

The running-time cost of this algorithm is twice the cost of the algorithm that solve the first problem. The memory cost is to remember the two tables - Forward and Backward. This is a linear cost that help us give a quick answer for many problems.
5 Numerical stability

Multiplying many probabilities gives small numbers. Such numbers can cause numerical problems:

- Underflow errors.
- Wrong numbers are calculated.

For instance, using the backward algorithm on 1000 based sequences, assuming that the product of one emission and one transition probability is typically 0.1, as we can see in Example 3.4, the probability of the backward result would be of the order $10^{-1000}$. Numbers from that order will cause most computers one of this two problems. 2 solutions for this problem are:
1. Log transformation
2. Scaling of probabilities

5.1 The log transformation

- Replace $F_i(H_i)$ with $f_i(H_i)$ where:
  $$f_i(H_i) = \ln(F_i(H_i))$$
  $$\Rightarrow f_{i+1}(H_{i+1}) = \ln \sum_{H'_i} e^{f_i(H_i)} \Pr(H_{i+1}|H_i) \Pr(x_{i+1}|H_{i+1})$$

  - Underflow problem is solved because $\log 10^{-100000} = -100000$.
  - Product operation replaced with sum operation that is often faster.
  - Dealing with sum operation using the equation:
    $$\ln(e^{a_1} + e^{a_2}) = c + \ln(e^{a_1-c} + e^{a_2-c})$$
    let:
    $$c = \max\{a_1, a_2\}$$
    $$b = \min\{a_1, a_2\} - c$$
    we will get:
    $$c + \ln(1 + e^b)$$
    in this case even if $b$ is very small the knowledge is not lost because most of it is in $c$. In other wards in case that $a_1 > a_2$ we will save just $a_1$ that is more important.

5.2 Scaling of probabilities

An alternative way is to rescaled the $F$ and $B$ variables, so they stay within a manageable numerical interval.

$$\tilde{F}_i(H_i) = \frac{F_i(H_i)}{\sum_{H'_i} F_i(H'_i)}$$
$$\tilde{B}_i(H_i) = \frac{B_i(H_i)}{\sum_{H'_i} B_i(H'_i)}$$

In this way we can simply solve the second problem:

$$\Pr(H_i|x_1, \ldots, x_n, \$) = \frac{F_i(H_i)B_i(H_i)}{\sum_{H'_i} F_i(H'_i)B_i(H'_i)} = \frac{\tilde{F}_i(H_i)\tilde{B}_i(H_i)}{\sum_{H'_i} \tilde{F}_i(H'_i)\tilde{B}_i(H'_i)}$$

For the first problem we need just to remember for each column in the table what was the sum of all the lines before we normalized them.