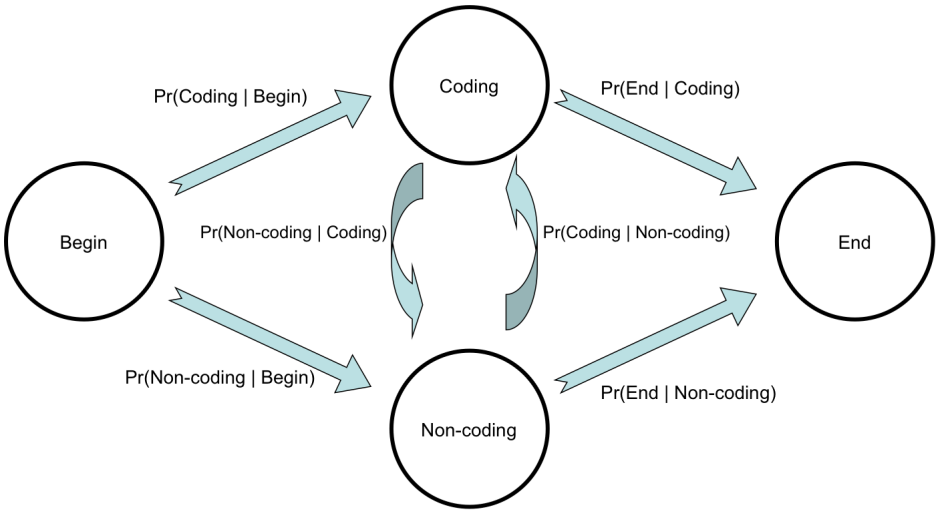
1 Introduction to Hidden Markov Models

1.1 A quick biological example

Some organisms have a genome that contains a higher GC content in protein coding regions than in non-coding regions. This can be modeled using a State Machine, which can be used to identify coding regions by their GC content.



- Each state outputs a base according to its probability distribution on bases.
- From the current state, the machine will pick the next state according to its transition probability distribution.
- The model is said to "generate" the observations (outputs).

	q_1	q_2	q_3	$\dots$			
States	N	N	N	C	C	C	C
Observations	A	T	T	G	A	C	G
	O_1	O_2	O_3	$\dots$			

1.2 HMM Applications

HMMs and variants of them have many applications in sequence analysis

- Predicting Exons and Introns in genomic DNA.

- Identifying functional motifs (domains) in proteins (profile HMM).
- Aligning two sequences (pair HMM).

1.3 Abstract definition of an HMM

An HMM consists of

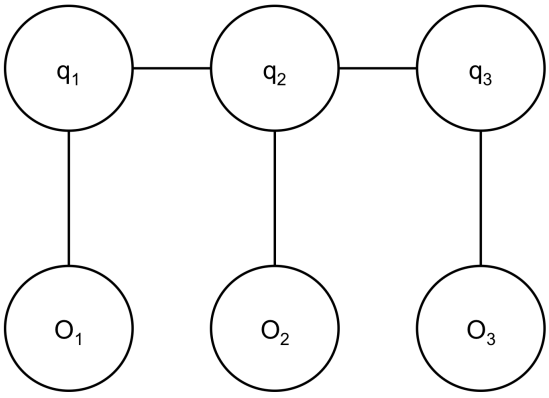
- A set of states : $\{ S_1, \ldots, S_N \}$.
- An alphabet of observation symbols : $\{ A_1, \ldots, A_M \}$. E.g. for DNA A, C, G, T.
- A probability distribution on pairs of sequences ($O_{1\ldots T}, q_{1\ldots T}$), where $O_{1\ldots T} \in \{A_1, \ldots, A_M\}^T$ (an *observation sequence*) and $q_{1\ldots T} \in \{S_1, \ldots, S_N\}^T$ (*a state sequence*), such that:

  1. The observation at time t, O_t, is independent of all other observations and states, given the state at time t, q_t.
  2. The state at time t, q_t, is independent of all earlier states and observations, given the state at time t-1, q_{t-1}.

  In formal notation:

  1. $Pr(O_t = A_k | q_{1\ldots T}, O_{1\ldots t-1}, O_{t+1\ldots T}) = Pr(O_t = A_k | q_t)$
  2. $Pr(q_t = S_k | q_{1\ldots t-1}, O_{1\ldots t-1}) = Pr(q_t = S_k | q_{t-1})$

1.4 Graphical Model Representation



- Direct influences (dependencies) are shown as edges.
- If vertex u lies on the path from x to y, then x and y are independent given $u : Pr(x|y,u) = Pr(x|u)$.

1.5 Factoring the joint distribution on state sequences and observed sequences

$$\begin{aligned} Pr(O_{1\ldots T}, q_{1\ldots T}) &= Pr(q_1) \cdot Pr(O_1|q_1) \cdot Pr(q_2|q_1, O_1) \cdot Pr(O_2|q_2, O_1, q_1) \cdot \ldots && \text{[CR]} \\ &= Pr(q_1) \cdot Pr(O_1|q_1) \prod_{t=2}^{T} Pr(q_t|q_{t-1}) \cdot Pr(O_t|q_t) && \text{[HMM independence]} \\ &= \prod_{t=1}^{T} Pr(q_t|q_{t-1}) \cdot Pr(O_t|q_t) \end{aligned}$$

where q_0 =BEGIN state

So an HMM can be completely specified by:

1. $Pr(q_1 = S_i|q_0 = BEGIN)$ for $i = 1, \ldots, N$, the initial state probabilities: π_i

2. $Pr(O_t = A_k|q_t = S_i)$ for $i = 1, \ldots, N, k = 1, \ldots, M$, the probability of outputting A_k from state $S_i : b_i(A_k)$

3. $Pr(q_t = S_j|q_{t-1} = S_i)$ for $i = 1, \ldots, N, j = 1, \ldots, N$, the probability that the HMM will transition to state S_j when in state $S_i : a_{ij}$.

These probabilities are the same for all t, they do not change!

2 The Decoding Problem

Given a series of observations $O_{1\ldots T}$, the decoding problem consists in finding the most likely state sequence, i.e.,

- The sequence of states that maximizes

$$Pr(q_{1\ldots T}|O_{1\ldots T}) = \frac{Pr(q_{1\ldots T}, O_{1\ldots T})}{Pr(O_{1\ldots T})}$$

- $O_{1\ldots T}$ is constant, so the same state sequence maximizes $Pr(q_{1\ldots T}, O_{1\ldots T})$.

2.1 The Viterbi Algorithm:Mechanics

- Dynamic Programming (DP) algorithm for determining the most likely sequence of states given a sequence of observations.

Example. A series of coin tosses using one fair coin and one coin with heads on both sides, switched with probability 0.25.

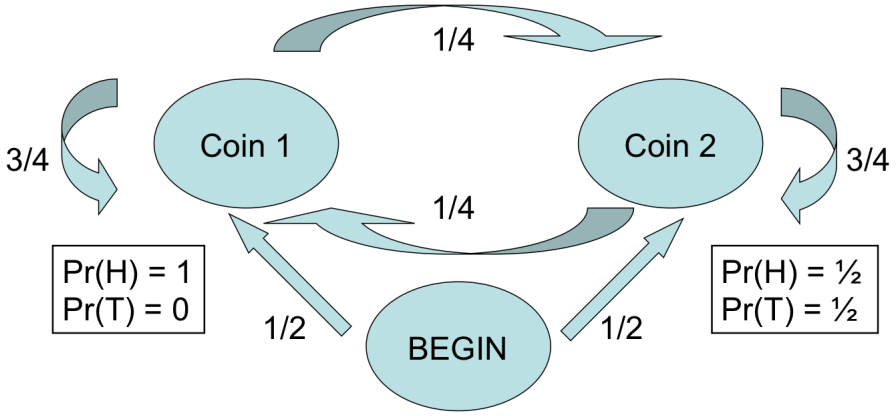


Figure 1: A simple HMM diagram

- What is the most likely sequence of states after you've seen T? How about TH? THH?
- Can be answered by filling out a dynamic programming table (Viterbi) as shown in Figure 2.
- If $\delta_t(i)$ is the table entry in row i and column t, then

$$\delta_t(i) = b_i(O_t) \max_{j=1}^{N} a_{ji}\delta_{t-1}(j)$$

  and $\delta_1(i) = b_i(O_1)\pi_i$
- Find a maximum cell in the final column
- Trace back to source of max in the previous column.

Example. Figure 3 shows the same Viterbi matrix as in Figure 2, but with the calculations in general notation.

To obtain the most likely state sequence after the observations THH, find the state (row) that gives the maximum δ in the the third column. In this case, $\delta_3(1) = 3/2^6$ is larger than $\delta_3(2) = 3^2/2^8$, so the final state of the most likely state sequence is state 1 (Coin 1). The second to the last state is also state 1 because $a_{1,1}\delta_2(1) > a_{2,1}\delta_2(2)$.

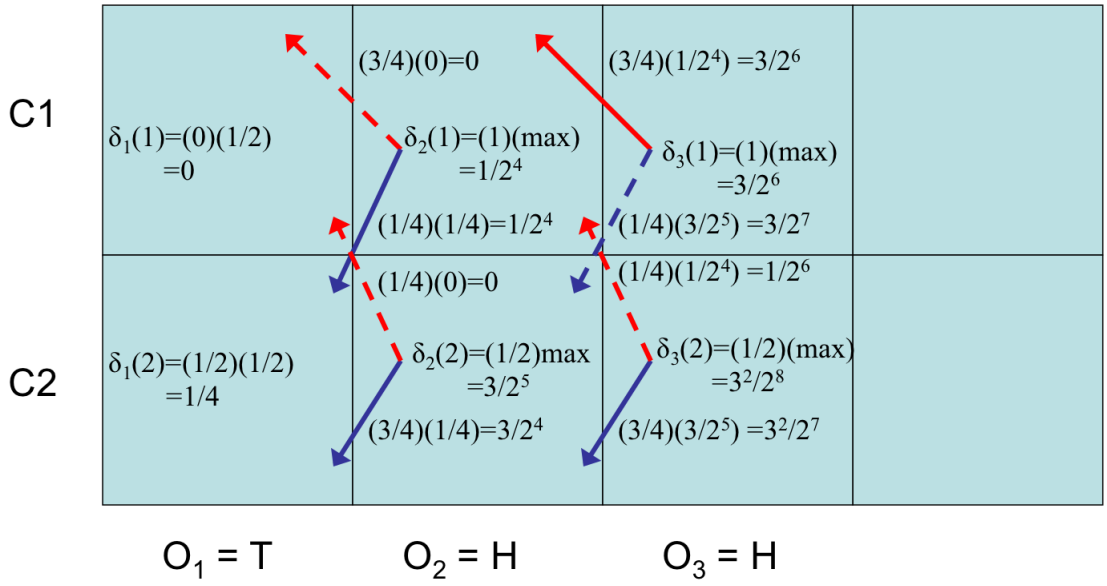


Figure 2: A partially completed Viterbi table

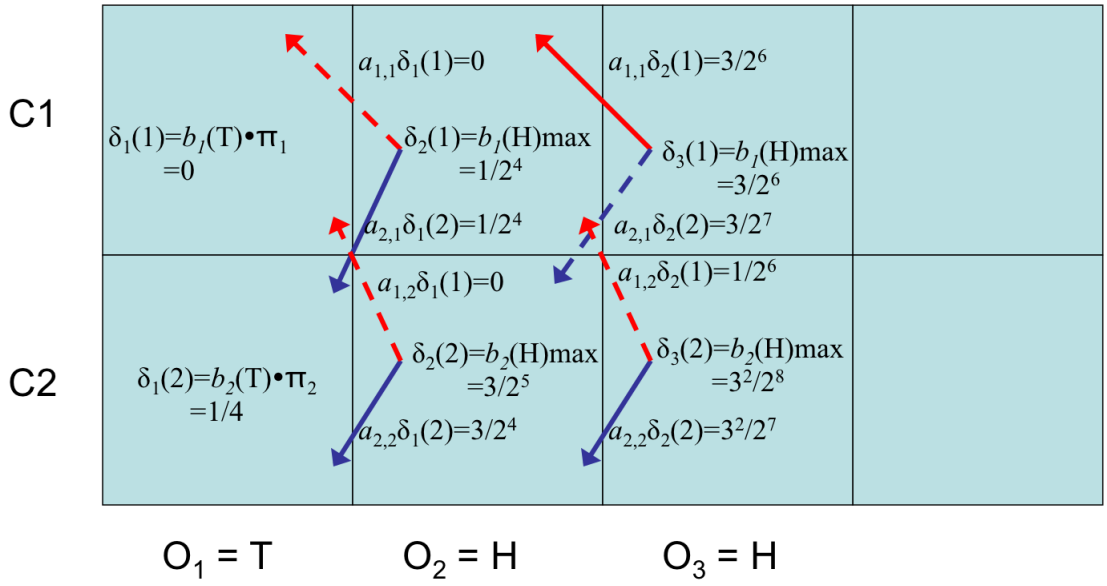


Figure 3: A partially completed Viterbi table in general notation

Exercise 2.1. For the HMM shown in Figure 1, fill out the 4th column of the table shown in Figure 2, assuming the 4th outcome was heads. Indicate which traceback pointer is most likely from each state, which state the traceback should start in, and what is the most likely state sequence.

Exercise 2.2. Follow the instructions for the previous exercise but assume the 4th outcome was tails.

2.2 Viterbi Algorithm: Correctness

Definition (argmax)**.** Let D be a set and $f : D \to \mathbb{R}$ a function and $S \subseteq D$. $\text{argmax}_{x \in S} f(x)$ is a member of S on which f takes its max value.

- E.g. $\text{argmax}_{x \in \mathbb{R}}[-(x-1)^2] = 1$, but $\max_{x \in \mathbb{R}}[-(x-1)^2] = 0$, (i.e. when $x = 1$).
- Note: For any constant C, $\text{argmax}_{x \in \mathbb{R}} Cf(x) = \text{argmax}_{x \in \mathbb{R}} f(x)$.

Theorem (Viterbi Traceback)**.** The final state of the most likely state sequence is $\text{argmax}_i \, \delta_T(i)$. If q^*_{t+1} is the $t+1^{th}$ state of the most likely state sequence then $\text{argmax}_i \, Pr(q^*_{t+1} | q_t = S_i)\delta_t(i)$ is the t^{th} state of the most likely state sequence.

To prove this, we must first have another theoreom.

Theorem (Viterbi Recursion)**.**

$$\delta_t(i) = \max_{q_{1\ldots t-1}} Pr(q_{1\ldots t-1}, q_t = S_i, O_{1\ldots t}),$$

the joint probability of the most likely state sequence ending in state S_i at time t and all the observations up to time t.

Proof. By induction. Base case.

$$\begin{aligned} \delta_1(i) &= b_i(O_1)\pi_i \\ &= Pr(O_1 | q_1 = S_i) Pr(q_1 = S_i) \\ &= Pr(q_1 = S_i, O_1) \end{aligned}$$

Now suppose the claim is true for times $1 \ldots t-1$. This supposition is called the *induction hypothesis*. Using the induction hypothesis, we will show it is true for

time t. Several lines are left blank as exercises.

$$\begin{aligned}
\delta_t(i) &= b_i(O_t) \max_{j=1}^{N} a_{ji}\delta_{t-1}(j) && \text{[Defn. of } \delta_t(i)\text{]} \\
&= b_i(O_t) \max_{j=1}^{N} a_{ji} \max_{q_{1\ldots t-2}} Pr(q_{t-1} = S_j, q_{1\ldots t-2}, O_{1\ldots t-1}) && \text{[Induction hyp.]} \\
&= \max_{j=1}^{N} \max_{q_{1\ldots t-2}} b_i(O_t) a_{ji} Pr(q_{t-1} = S_j, q_{1\ldots t-2}, O_{1\ldots t-1}) && \text{[Move Constants]} \\
&= \max_{j=1}^{N} \max_{q_{1\ldots t-2}} b_i(O_t) Pr(q_t = S_i | q_{t-1} = S_j) Pr(q_{t-1} = S_j, q_{1\ldots t-2}, O_{1\ldots t-1}) && \text{[Defn } a_{ji}\text{]} \\
&= \text{See exercises} && \text{[HMM indep.]} \\
&= \max_{j=1}^{N} \max_{q_{1\ldots t-2}} b_i(O_t) Pr(q_{1\ldots t-2}, q_{t-1} = S_j, q_t = S_i, O_{1\ldots t-1}) && \text{[Chain rule]} \\
&= \max_{q_{1\ldots t-1}} b_i(O_t) Pr(q_{1\ldots t-1}, q_t = S_i, O_{1\ldots t-1}) && \text{[Absorb } \max_{j=1}^{N} \text{ into } \max_{q_{1\ldots t-1}}\text{]} \\
&= \max_{q_{1\ldots t-1}} Pr(O_t | q_t = S_i) Pr(q_{1\ldots t-1}, q_t = S_i, O_{1\ldots t-1}) && \text{Defn } b_i(O_t) \\
&= \max_{q_{1\ldots t-1}} Pr(O_t | q_{1\ldots t-1}, q_t = S_i, O_{1\ldots t-1}) Pr(q_{1\ldots t-1}, q_t = S_i, O_{1\ldots t-1}) && \text{[HMM Indep.]} \\
&= \text{See exercises} && \text{[Chain rule]}
\end{aligned}$$

□

Proof. **Viterbi Traceback**. The last state of the most likely state sequence is:

$$\text{argmax}_i \max_{q_{1\ldots T-1}} Pr(q_{1\ldots T-1}, q_T = S_i, O_{1\ldots T}),$$

which, by the Recursion Theorem, is the same as:

$$\text{argmax}_i \, \delta_T(i)$$

This establishes the base case for a proof by induction where we prove something is true at an earlier time if it is true at a later time.

Suppose q^*_{t+1} is the $t+1^{th}$ state of the most likely state sequence. This is the induction hypothesis. Given the induction hypothesis, we will prove that $\text{argmax}_i \, Pr(q^*_{t+1} | q_t = S_i)\delta_t(i)$ is the t^{th} state of the most likely state sequence.

$\text{argmax}_i \, Pr(q^*_{t+1}|q_t = S_i)\delta_t(i)$

$$
\begin{aligned}
&= \text{argmax}_i \, Pr(q^*_{t+1}|q_t = S_i) \max_{q_{1\ldots t-1}} Pr(q_{1\ldots t-1}, q_t = S_i, O_{1\ldots t}) && \text{[Recursion Thm.]}\\
&= \text{argmax}_i \max_{q_{1\ldots t-1}} Pr(q^*_{t+1}|q_t = S_i) Pr(q_{1\ldots t-1}, q_t = S_i, O_{1\ldots t}) && \text{[Move const.]}\\
&= \text{argmax}_i \max_{q_{1\ldots t-1}} Pr(q^*_{t+1}|q_{1\ldots t-1}, q_t = S_i, O_{1\ldots t}) Pr(q_{1\ldots t-1}, q_t = S_i, O_{1\ldots t}) && \text{[See exercises]}\\
&= \text{argmax}_i \max_{q_{1\ldots t-1}} Pr(q^*_{t+1}, q_{1\ldots t-1}, q_t = S_i, O_{1\ldots t}) && \text{[Chain Rule]}\\
&= \text{argmax}_i [\max_{q_{t+2\ldots T}} Pr(q_{t+2\ldots T}, O_{t+1\ldots T}|q^*_{t+1})] \max_{q_{1\ldots t-1}} Pr(q^*_{t+1}, q_{1\ldots t-1}, q_t = S_i, O_{1\ldots t}) && \text{[Const. in argmax]}\\
&= \text{argmax}_i \max_{q_{1\ldots t-1}} \max_{q_{t+2\ldots T}} Pr(q_{t+2\ldots T}, O_{t+1\ldots T}|q^*_{t+1}) Pr(q^*_{t+1}, q_{1\ldots t-1}, q_t = S_i, O_{1\ldots t}) && \text{[Move const.]}\\
&= \text{argmax}_i \max_{q_{1\ldots t-1}, q_{t+2\ldots T}} Pr(q_{t+2\ldots T}, O_{t+1\ldots T}|q^*_{t+1}, q_{1\ldots t-1}, q_t = S_i, O_{1\ldots t}) && \\
&\qquad Pr(q^*_{t+1}, q_{1\ldots t-1}, q_t = S_i, O_{1\ldots t}) && \text{[HMM Indep.]}\\
&= \text{argmax}_i \max_{q_{1\ldots t-1}, q_{t+2\ldots T}} Pr(q_{t+2\ldots T}, O_{t+1\ldots T}, q^*_{t+1}, q_{1\ldots t-1}, q_t = S_i, O_{1\ldots t}) && \text{[See exercises]}\\
&= \text{argmax}_i \max_{q_{1\ldots t-1}, q_{t+2\ldots T}} Pr(q_{t+2\ldots T}, q^*_{t+1}, q_{1\ldots t-1}, q_t = S_i, O_{1\ldots T}) && \text{Notation}
\end{aligned}
$$

which is the t^{th} state of the most likely state sequence. □

Exercise 2.3. Two lines in the proof of the Viterbi Recursion theorem in the course notes were left blank, though the justifications were provided. Please fill in those two lines.

Exercise 2.4. Two justification lines in the proof of the Viterbi Traceback theorem were also left blank. Please fill in those two lines.

Exercise 2.5. There is a casino that sometimes uses a fair die and sometimes uses a loaded die. You get to see the outcomes of the rolls, but not which die is used. For each die, you are given the probabilty of the faces, which are numbered 1 to 6. You are also given the probabilities of switching dice and the probability of starting with each die type – in short, an HMM.

Suppose you know that an Occasionally Dishonest Casino never uses the loaded die more than 2 times in a row. If you do nothing about this, the state sequence returned by the Viterbi algorithm may be an impossible one with 3 or more consecutive uses of the loaded die. This may happen if, by chance, some fair rolls that look more likely under the loaded model are adjacent to loaded rolls that also look more likely under the loaded model.

Your friend says, "No problem! As you trace back, if you have already gone through 2 consecutive instances of the "loaded"; state, automatically trace back to the "fair" state next, ignoring the calculated deltas.

This is easy to implement, and it will give you a state sequence with no runs of 3 or more "loaded" states. However, it is not guaranteed to be the most likely sequence with no runs of 3 or more "loaded" states.

1. Explain in simple English why this may not yield the most likely state sequence without 3 consecutive "loaded" states. Give an example where the result is not optimal, using the Occasionally Dishonest Casino. Specify the complete HMM and demonstrate, using the begin, transition, and emission probabilities, that the suggested algorithm does not produce the most likely state sequence with at most two consecutive "loaded" states.

2. Point out where the proofs of the Traceback Theorem and the Viterbi Recursion Theorem break down when the constraint forbidding more than 2 consecutive loaded rolls is imposed. I.e., why do the guarantees provided by these theorems no longer hold when this constraint is imposed?

   (An alternative algorithm would be to change the transition probabilities on the forward pass of Viterbi. Whenever the traceback from the loaded state at time t goes through the loaded state at $t-1$ and $t-2$, the transition probability from loaded at $t-1$ to loaded at t is set to 0, forcing the traceback away from the path through loaded at $t-1$ and $t-2$. This may give a different state sequence than the first algorithm. However, it is again not guaranteed to be the most likely state sequence with no more 3 consecutive loaded states. The reasons are the same.)

3. It is possible to correctly model the casino that never uses the loaded die more than twice in a row with an HMM, just not the two-state HMM we've talked about so far. What is the structure of the correct HMM for this situation, in terms of the number of states, their interpretations, and the allowable transitions among them?

2.3 Application of 4 rules for manipulating conditional probability expressions

- Goal : given some independence assumptions, simplify a joint probability $Pr(x_1, x_2, \ldots, x_n)$.

  1. Use chain rule to factor the joint distribution into a product of conditional distributions.
  2. Manipulate factors so that the independence relations can be used to replace complex expressions by simpler ones by:
     - Using Bayes Rule to swap a variable behind the conditioning bar with one in front.
     - Introducing a new variable in front of the conditioning bar and summing it out.

– Using an exhaustive conditionalization to introduce a new variable behind the conditioning bar.

3 Summing Out States

Used to compute:

- $Pr(O_{1...T})$, the probability of generating $O_{1...T}$ by any state sequence.
- Probability of being in state S_i at time t, $Pr(q_t = S_i|O_{1...T})$.
- Most likely state at time t, $\text{argmax}_i \, Pr(q_t = S_i|O_{1...T})$.

Note: Most likely state sequence (Viterbi) $\neq$ sequence of most likely states.

Definition. Define $\alpha_t(i) = Pr(O_{1...t}, q_t = S_i)$.

Remark. The sum of the final alphas is the probability of the observations.

$$\sum_i^N \alpha_T(i) = \sum_i^N Pr(O_{1...T}, q_T = S_i) = Pr(O_{1...T})$$

Theorem (Forward algorithm)**.** The alphas can be calculated as follows:

$$\alpha_1(i) = b_i(O_1)\pi_i$$
$$\alpha_t(i) = b_i(O_t)\sum_{j=1}^N a_{ji}\alpha_{t-1}(j).$$

Same as δ recursion except that max is replaced by sum.

Proof. Omitted lines are exercises.

$$\begin{aligned}
\alpha_1(i) &= Pr(O_1, q_1 = S_i) && \text{[Defn. } \alpha\text{]} \\
&= Pr(O_1 | q_1 = S_i) Pr(q_1 = S_i) && \text{[Chain Rule]} \\
&= b_i(O_1)\pi_i && \text{[Defn. } \beta, \pi\text{]}
\end{aligned}$$

$$\begin{aligned}
\alpha_t(i) &= Pr(O_{1\ldots t}, q_t = S_i) && \text{[Defn. } \alpha\text{]} \\
&= && \text{[Chain Rule]} \\
&= && \text{[HMM Indep.]} \\
&= b_i(O_t) Pr(O_{1\ldots t-1}, q_t = S_i) && \text{[Defn } b\text{]} \\
&= b_i(O_t) \sum_{j=1}^{N} Pr(O_{1\ldots t-1}, q_t = S_i, q_{t-1} = S_j) && \text{[Summing Out]} \\
&= && \text{[Chain Rule]} \\
&= b_i(O_t) \sum_{j=1}^{N} Pr(q_t = S_i | q_{t-1} = S_j) Pr(O_{1\ldots t-1}, q_{t-1} = S_j) && \text{[HMM indep.]} \\
&= b_i(O_t) \sum_{j=1}^{N} a_{ji} \alpha_{t-1}(j) && \text{[Defn. } a, \alpha\text{]}
\end{aligned}$$

□

Exercise 3.1. Derive the recursion for the forward probabilities (α) by filling in the partial derivation in the course notes.

4 Posterior Decoding

- Viterbi finds the state sequence $q^*_{1\ldots T}$ that maximizes $Pr(q_{1\ldots T} | O_{1\ldots T})$
- For a given t, q^*_t may not maximize $Pr(q_t | O_{1\ldots T})$.
- We may care only about some parts of the sequence.
  - E.g. whether a particular genomic region is transcribed.
  - The rest of the genome is just evidence
- *Posterior decoding* is choosing the most likely state at each point in time, $\text{argmax}_i \, Pr(q_t = S_i | O_{1\ldots T})$, regardless of the states chosen for other times.
- To carry out posterior decoding, we calculate $Pr(q_t = S_i | O_{1\ldots T})$ for each state i and time t using the forward-backward algorithm.

Definition. The *backward probabilities*, $\beta_t(i)$, are: $\beta_t(i) \equiv Pr(O_{t+1\ldots T}|q_t = S_i)$.

Theorem (Backward Algorithm)**.** The betas can be calculated as follows

$$\beta_T(i) = 1$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$$

Theorem. The posterior probability of state S_i at time t is:

$$Pr(q_t = S_i|O_{1\ldots T}) = \frac{\beta_t(i)\alpha_t(i)}{\sum_i \beta_t(i)\alpha_t(i)} \tag{1}$$

Proof.

$$\begin{aligned}
Pr(q_t = S_i|O_{1\ldots T}) &= \frac{Pr(q_t = S_i, O_{1\ldots T})}{Pr(O_{1\ldots T})} && \text{[Defn. Cond. Prob.]} \\
&= \frac{Pr(q_t = S_i, O_{1\ldots T})}{\sum_k Pr(q_t = S_k, O_{1\ldots T})} && \text{[Summing out]} \\
Pr(q_t = S_i, O_{1\ldots T}) &= Pr(O_{t+1\ldots T}|q_t = S_i, O_{1\ldots t}) \cdot Pr(q_t = S_i, O_{1\ldots t}) && \text{[Chain Rule]} \\
&= Pr(O_{t+1\ldots T}|q_t = S_i) \cdot Pr(q_t = S_i, O_{1\ldots t}) && \text{[HMM Indep.]} \\
&= \beta_t(i)\alpha_t(i), \text{ so,} && \text{[Defn. } \alpha, \beta\text{]} \\
Pr(q_t = S_i|O_{1\ldots T}) &= \frac{\beta_t(i)\alpha_t(i)}{\sum_k \beta_t(k)\alpha_t(k)}
\end{aligned}$$

□

- **Algorithm**: precompute α and β matrices.
- Compute

  $$\text{argmax}_{i=1}^{N} Pr(q_t = S_i|O_{1\ldots T}) = \text{argmax}_{i=1}^{N} \alpha_t(i)\beta_t(i)$$

  Note that the denominator $\sum_k \beta_t(k)\alpha_t(k)$ is constant that does not depend on i, so it does not affect which i gives the maximum value. Therefore, it can be dropped from within the argmax
- TIME=$O(RT)$ where R =# non-zero transitions.

Exercise 4.1. Derive the backward probabilities (β) from scratch. This is very similar to the derivation of the forward probabilities that you did in the previous assignment.

Exercise 4.2. Manually compute the forward (alpha), backward (beta), and posterior probabilities of each state of the model in Figure 1 for observation sequences "TH", "THH", and "THHH". Turn in all three tables for all three input sequences. Also, note the sequence of most likely states for each coin-flip in each sequence.

1. Does seeing more coin-flips change the alphas calculated for the earlier coin-flips? How about the betas?
2. Does seeing more coin-flips change the most likely states for earlier coin-flips?
3. Does posterior decoding give the same state sequence as Viterbi for all three cases? If not, please try to explain any differences between the two. Why does it make sense that the most likely state sequence and the sequence of most likely states would be different in these cases?

5 HMM Parameter Estimation from labeled data

- Given an "example run" $(O_{1\ldots T}, q_{1\ldots T})$ of the HMM
  - E.g. DNA sequence with known exon-intron structures.
- Use maximum likelihood

$$\hat{b}_i(A_j) = \frac{\text{count}_{t=1}^{T}(O_t = A_j, q_t = S_i)}{\text{count}_{t=1}^{T}(q_t = S_i)}$$

$$\hat{a}_{ij} = \frac{\text{count}_{t=0}^{T-1}(q_t = S_i, q_{t+1} = S_j)}{\text{count}_{t=0}^{T-1}(q_t = S_i)}$$

- Or use a Bayesian estimator (e.g + pseudocounts).

6 HMM Parameter Estimation from Unlabeled Data

Use Expectation Maximization. State labels are hidden data.

- Start with arbitrary parameters
- Use them to compute posterior distribution on states at each time.
- Repeat until convergence:

- Expectation: Compute expected counts from posterior distributions.
- Maximization: Use Max. Likelihood to re-estimate parameters.

- Called Forward-Backward or Baum-Welch re-estimation.
- Likelihood of observations guaranteed to increase every cycle.
- Moves toward local (not necessarily global) max of likelihood.
- Define $\gamma_t(i) = Pr(q_t = S_i|O_{1...T})$, posterior probability S_i at t.
- Define $\xi_t(i,j) = Pr(q_t = S_i, q_{t+1} = S_j|O_{1...T})$ posterior of $S_i \to S_j$ at t
  - Note $\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j)$
  - $\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum \alpha_T(i)}$ [Proof is left as an exercise.]
- $\sum_{t=0}^{T-1} \xi_t(i,j)$ is expected # $S_i \to S_j$ transitions while generating $O_{1...T}$
- $\sum_{t=0}^{T-1} \gamma_t(i) = \sum_{t=0}^{T-1} \sum_{j=1}^{N} \xi_t(i,j)$ is expected # times in S_i.
- Re-estimation: Same as for labeled data, but use expected counts:

$$\hat{a}'_{i,j} = \frac{E[\#i \to j \text{ transitions}]}{E[\#\text{times in } S_i]} = \frac{\sum_{t=0}^{T-1} \xi_t(ij)}{\sum_{t=0}^{T-1} \gamma_t(i)} \tag{2}$$

$$\hat{b}_i(A_k) = \frac{E[\#\text{times in } S_i \text{ when } A_k \text{ is emitted}]}{E[\#\text{times in } S_i]} = \frac{\sum_{\{t \text{ s.t. } O_t = A_k\}} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{3}$$

Exercise 6.1. Recall that, by definition

$$\xi_t(i,j) = Pr(q_t = S_i, q_{t+1} = S_j|O_{1...T}) = \frac{Pr(q_t = S_i, q_{t+1} = S_j, O_{1...T})}{Pr(O_{1...T})}$$

Provide a derivation showing

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum \alpha_T(i)}$$