Hidden Markov Models-3 Smoothing

M. R. Hasan
Readings

• Alpaydin: 15
• Russell & Norvig: 15
What We Will Cover

• Smoothing Problem
• Backward Variable
• Forward-Backward Algorithm
Background Assumed

• Recursion & Dynamic Programming
• Complexity Theory
• Graphical Models
• d-separation
Hidden Markov Model (HMM)

- We **formalize HMM** by using the following elements.

  \[ S = \{ S_1, S_2, \ldots, S_N \} \]

  \[ V = \{ v_1, v_2, \ldots, v_m \} \]

  \[ A = [a_{ij}] \text{ where } a_{ij} \equiv P(q_{t+1} = S_j | q_t = S_i) \]

  \[ B = [b_j(m)] \text{ where } b_j(m) \equiv P(O_t = v_m | q_t = S_j) \]

  \[ \Pi = [\pi_i] \text{ where } \pi_i \equiv P(q_1 = S_i) \]

  \[ \lambda = (A, B, \Pi) \]
HMM: Four Inference Problems

- Given a number of sequences of observations, we are interested in four inference problems:

  **Problem 1 (Evaluation):** Given a model $\lambda$, we would like to evaluate the probability of any given observation sequence $O = \{O_1 O_2 \ldots O_T\}$: $P(O \mid \lambda)$

As part of the evaluation problem, we solve two sub-problems:

- **Filtering:** Compute the posterior distribution over the *most recent state* given all evidence (observation) to date.
- **Prediction:** Compute the posterior distribution over the *future state* given all evidence (observation) to date.
HMM: Four Inference Problems

Problem 2 (Smoothing): Given a model $\lambda$, we would like to compute the posterior distribution over a past state given all evidence up to the present $O = \{O_1 O_2 \ldots O_T\}$: $P(q_t = S_i | O_{1:T}, \lambda)$

Problem 3 (Most likely Explanation): Given a model $\lambda$ and an observation sequence $O$, we would like to find out the state sequence $Q = \{q_1 q_2 \ldots q_T\}$, which has the highest probability of generating $O$; namely, we want to find $Q^*$ that maximizes $P(Q | O, \lambda)$.

Problem 4 (Learning): Given a training set of observation sequences, $X = \{O^k\}_{k}$, we would like to learn the model that maximizes the probability of generating $X$; namely, we want to find $\lambda^*$ that maximizes $P(X | \lambda)$. 
**Problem 1 (Evaluation):** Given a model $\lambda$, we would like to evaluate the probability of any given observation sequence $O = \{O_1, O_2 \ldots, O_T\}$: $P(O \mid \lambda)$

As part of the evaluation problem, we solve two sub-problems:

- **Filtering:** Compute the posterior distribution over the most recent state given all evidence (observation) to date.
- **Prediction:** Compute the posterior distribution over the future state given all evidence (observation) to date.

**Problem 2 (Smoothing):** Given a model $\lambda$, we would like to compute the posterior distribution over a past state given all evidence up to the present $O = \{O_1, O_2 \ldots, O_T\}$: $P(q_t = S_i \mid O_{1:T}, \lambda)$

**Problem 3 (Most likely Explanation):** Given a model $\lambda$ and an observation sequence $O$, we would like to find out the state sequence $Q = \{q_1, q_2 \ldots, q_T\}$, which has the highest probability of generating $O$; namely, we want to find $Q^*$ that maximizes $P(Q \mid O, \lambda)$.

**Problem 4 (Learning):** Given a training set of observation sequences, $X = \{O^k\}_k$, we would like to learn the model that maximizes the probability of generating $X$; namely, we want to find $\lambda^*$ that maximizes $P(X \mid \lambda)$. 

**HMM: Four Inference Problems**
Problem 2: Smoothing

Smoothing is about remaking the past.

We revisit past and compute the probability of a hidden state.

We "smooth" out our beliefs (mistakes)!

Problem 2 (Smoothing): Given a model $\lambda$, we would like to compute the posterior distribution over a past state given all evidence up to the present $O = \{O_1, O_2, \ldots, O_T\}$: $P(q_t = S_i | O_{1:T}, \lambda)$
Problem 2: Smoothing

• For example, imagine a problem in which based on the observation of umbrella we estimate the probability of rain.

• **Day 1:** umbrella was observed, and we compute the probability of rain on day 1

• **Day 2:** umbrella was observed *again*, and we **re-compute** the probability of rain on **day 1**

Hence, based on the **recent observation** (evidence) we **re-estimate (smooth)** our belief about a past state.

By **re-making the past** we are able to **discover a smoother pattern** of the state sequence.
Problem 2: Smoothing

To solve the smoothing problem, we can break the problem into two parts.

The red line explains the starting part of the sequence until time $t$ and ends in $q_t = S_1$.

The blue line explains the ending part until time $T$. 

Problem 2 (Smoothing): Given a model $\lambda$, we would like to compute the posterior distribution over a past state given all evidence up to the present $O = \{O_1 O_2 \ldots O_T\}: P(q_t = S_1 | O_{1:T}, \lambda)$
Problem 2: Smoothing

- To solve the smoothing problem, we can break the problem into two parts.

We use the forward variable $\alpha_t(i)$ (red line) to explain the starting part of the sequence until time $t$ and ends in $q_t = S_i$.

$$\alpha_t(i) \equiv P(O_1 O_2 ... O_t, q_t = S_i | \lambda)$$

For explaining the blue line which is the ending part until time $T$ we will use another variable called the backward variable $\beta_t(i)$:

$$\beta_t(i) \equiv P(O_{t+1:T} | q_t = S_i, \lambda)$$
Problem 2: Smoothing

• Now we will define the **backward variable** (and Backward algorithm).

**Problem 2 (Smoothing):** Given a model $\lambda$, we would like to compute the posterior distribution over a past state given all evidence up to the present $O = \{O_1 O_2 \ldots O_T\}$: $P(q_t = S_i | O_{1:T}, \lambda)$

For explaining the blue line which is the **ending part** until time $T$ we will use another variable called the backward variable $\beta_t(i)$:

$$\beta_t(i) \equiv P(O_{t+1:T} | q_t = S_i, \lambda)$$
Problem 2: Backward Algorithm

- The **backward variable** $\beta_t(i)$ is the probability of being in state $q_t = S_i$ at time $t$ and observing the partial sequence $O_{t+1}, O_{t+2}, \ldots, O_T$.

$$\beta_t(i) \equiv P(O_{t+1:T} | q_t = S_i, \lambda)$$
Problem 2: Backward Algorithm

\[ \beta_t(i) \equiv P(O_{t+1:T} \mid q_t = S_i, \lambda) \]

We marginalize over the next state \( j \) (at \( t+1 \)) so that we can use the model (transition probabilities, etc.):

\[
P(O_{t+1:T} \mid q_t = S_i, \lambda) = \sum_{j=1}^{N} P(O_{t+1:T}, q_{t+1} = S_j \mid q_t = S_i, \lambda)
\]

Using the observation:
\[
P(a, b, c \mid d, e) = P(a, b \mid c, d, e) P(c \mid d, e)
\]

Let’s derive this observation:

\[
= \sum_{j=1}^{N} P(O_{t+1, O_{t+2:T}}, q_{t+1} = S_j \mid q_t = S_i, \lambda) P(q_{t+1} = S_j \mid q_t = S_i, \lambda)
\]
Problem 2: Backward Algorithm

Derivation of $P(a, b, c \mid d, e) = P(a, b \mid c, d, e) P(c \mid d, e)$

\[
P(a, b, c, d, e) = P(a, b, c \mid d, e) P(d, e)
\]

\[
P(a, b, c, d, e) = P(a, b \mid c, d, e) P(c, d, e)
\]

\[
= P(a, b \mid c, d, e) P(c \mid d, e) P(d, e)
\]

\[
P(a, b, c \mid d, e) P(d, e) = P(a, b \mid c, d, e) P(c \mid d, e) P(d, e)
\]

\[
P(a, b, c \mid d, e) = P(a, b \mid c, d, e) P(c \mid d, e)
\]
Problem 2: Backward Algorithm

\[ P(O_{t+1:T} \mid q_t = S_i, \lambda) \]

\[ P(\lambda \mid O_{1:T}, q_t = S_i) = \sum_{t=1}^{N} P(O_{t+1:T}, q_{t+1} = S_j \mid q_t = S_i, \lambda) \]

\[ = \sum_{t=1}^{N} P(O_{t+1:T}, q_{t+1} = S_j \mid q_t = S_i, \lambda) \]

\[ = \sum_{t=1}^{N} P(O_{t+1:T} \mid q_{t+1} = S_j, q_t = S_i, \lambda) P(q_{t+1} = S_j \mid q_t = S_i, \lambda) \]

\[ = \sum_{t=1}^{N} P(O_{t+1} \mid q_{t+1} = S_j, q_t = S_i, \lambda) P(O_{t+2:T} \mid q_{t+1} = S_j, q_t = S_i, \lambda) P(q_{t+1} = S_j \mid q_t = S_i, \lambda) \]

\[ = \sum_{t=1}^{N} \frac{P(O_{t+1}, O_{t+2:T} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(q_{t+2:T} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(q_{t+1} \mid q_t = S_i)}{P(q_{t+1} \mid q_t = S_i)} \]

\[ = \sum_{t=1}^{N} \frac{P(O_{t+1} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(O_{t+2:T} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(q_{t+1} \mid q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \]

\[ = \sum_{t=1}^{N} \frac{P(O_{t+1} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(O_{t+2:T} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(q_{t+1} \mid q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \]

\[ = \sum_{t=1}^{N} \frac{P(O_{t+1} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(O_{t+2:T} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(q_{t+1} \mid q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \]

\[ = \sum_{t=1}^{N} \frac{P(O_{t+1} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(O_{t+2:T} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(q_{t+1} \mid q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \]

\[ = \sum_{t=1}^{N} \frac{P(O_{t+1} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(O_{t+2:T} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(q_{t+1} \mid q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \]

\[ = \sum_{t=1}^{N} \frac{P(O_{t+1} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(O_{t+2:T} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(q_{t+1} \mid q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \]

\[ = \sum_{t=1}^{N} \frac{P(O_{t+1} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(O_{t+2:T} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(q_{t+1} \mid q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \]

\[ = \sum_{t=1}^{N} \frac{P(O_{t+1} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(O_{t+2:T} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(q_{t+1} \mid q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \]

\[ = \sum_{t=1}^{N} \frac{P(O_{t+1} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(O_{t+2:T} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(q_{t+1} \mid q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \]

\[ = \sum_{t=1}^{N} \frac{P(O_{t+1} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(O_{t+2:T} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(q_{t+1} \mid q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \]

\[ = \sum_{t=1}^{N} \frac{P(O_{t+1} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(O_{t+2:T} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(q_{t+1} \mid q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \]

\[ = \sum_{t=1}^{N} \frac{P(O_{t+1} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(O_{t+2:T} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(q_{t+1} \mid q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \]

\[ = \sum_{t=1}^{N} \frac{P(O_{t+1} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(O_{t+2:T} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(q_{t+1} \mid q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \]

\[ = \sum_{t=1}^{N} \frac{P(O_{t+1} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(O_{t+2:T} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(q_{t+1} \mid q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \]

\[ = \sum_{t=1}^{N} \frac{P(O_{t+1} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(O_{t+2:T} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(q_{t+1} \mid q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \]

\[ = \sum_{t=1}^{N} \frac{P(O_{t+1} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(O_{t+2:T} \mid q_{t+1} = S_j, q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \frac{P(q_{t+1} \mid q_t = S_i)}{P(q_{t+1} = S_j \mid q_t = S_i)} \]
Problem 2: Backward Algorithm

\[ \beta_t(i) \equiv P(O_{t+1:T} \mid q_t = S_i, \lambda) \]

\[
= \sum_{j=1}^{N} P(O_{t+1} \mid q_{t+1} = S_j, \lambda) P(O_{t+2:T} \mid q_{t+1} = S_j, \lambda) P(q_{t+1} = S_j \mid q_t = S_i, \lambda)
\]

\[
= \sum_{j=1}^{N} b_j(O_{t+1}) \beta_{t+1}(j) a_{ij}
\]

\[
= \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)
\]

This sum is on the states at \( t+1 \)

Recursion:

\[ \beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \]
Problem 2: Backward Algorithm

• Thus we can compute the backward variable $\beta_t(i)$ by recursion.

\[ \beta_t(i) \equiv P(O_{t+1:T} \mid q_t = S_i, \lambda) \]

We initialize (at $t = T$) the backward probability with

\[ \beta_T(i) = P(O_{t+1:T} \mid q_T = S_i, \lambda) \]

\[ \beta_T(i) = P( \mid q_T = S_i, \lambda) = 1 \]

Because $O_{t+1:T}$ is an empty sequence with the probability of observing 1.

Recursion:

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

Initialization:

\[ \beta_T(i) = 1 \]
Problem 2: Scaling in Backward Algorithm

• For scaling $\beta_t(i)$ we use the **same scaling or normalizing coefficient** for each time $t$ that we used for the **forward variable** $\alpha_t(i)$.

• Recall, for normalizing $\alpha_t(i)$ the **scaling** coefficient $c_t$ is defined as follows.

$$c_t = \frac{1}{\sum_j \alpha_t(j)}$$

Since the magnitudes of $\alpha$ and $\beta$ are **comparable**, using the **same** scaling coefficients on the $\beta$s is an **effective way** of keeping the computation within **reasonable bounds**.
Problem 2: Scaling in Backward Algorithm

- We compute the normalized $\hat{\beta}_t(i)$ at each time-step by multiplying $\beta$ by the same $c_t$ from the scaling of the forward variable.

\[
\hat{\beta}_t(i) = c_t \beta_t(i)
\]

Note that $\hat{\beta}_t(i)$ do not sum to 1.

Also, since $\beta$ is computed backwards, the $\hat{\beta}_T(i)$ for the final time-step $t = T$ is computed as follows:

For $i = 1$ to $N$

\[
\hat{\beta}_T(i) = c_T
\]
Problem 2: Smoothing

Let’s go back to the smoothing problem.

We want to compute the posterior distribution over a past state \(q_t\) given all evidence (observations) up to present \(O_{1:T}\):

\[
P(q_t = S_i | O_{1:T}, \lambda) = \frac{P(O_{1:T} | q_t = S_i, \lambda) P(q_t = S_i | \lambda)}{P(O_{1:T} | \lambda)}
\]

We used the following observation:

\[
P(a | b, c) = \frac{P(b | a, c) P(a | c)}{P(b | c)}
\]
Problem 2: Smoothing

Derivation of the following observation:

\[ P(a \mid b, c) = \frac{P(b \mid a, c) \cdot P(a \mid c)}{P(b \mid c)} \]

\[ P(a, b, c) = P(a \mid b, c) \cdot P(b, c) \]
\[ = P(a \mid b, c) \cdot P(b \mid c) \cdot P(c) \]

\[ P(a, b, c) = P(b \mid a, c) \cdot P(a, c) \]
\[ = P(b \mid a, c) \cdot P(a \mid c) \cdot P(c) \]

\[ P(a \mid b, c) \cdot P(b \mid c) \cdot P(c) = P(b \mid a, c) \cdot P(a \mid c) \cdot P(c) \]

\[ P(a \mid b, c) \cdot P(b \mid c) = P(b \mid a, c) \cdot P(a \mid c) \]

\[ P(a \mid b, c) = \frac{P(b \mid a, c) \cdot P(a \mid c)}{P(b \mid c)} \]
Problem 2: Smoothing

\[ P(q_t = S_i | O_{1:T}, \lambda) = \frac{P(O_{1:T} | q_t = S_i, \lambda) P(q_t = S_i | \lambda)}{P(O_{1:T} | \lambda)} \]

\[ = \frac{P(O_{1:t}, O_{t+1:T} | q_t = S_i, \lambda) P(q_t = S_i | \lambda)}{P(O_{1:T} | \lambda)} \]

\[ = \frac{P(O_{1:t} | q_t = S_i, \lambda)P(O_{t+1:T} | q_t = S_i, \lambda) P(q_t = S_i | \lambda)}{\sum_{j=1}^{N} P(O_{1:T}, q_t = S_j | \lambda)} \]

\[ = \frac{[P(O_{1:t} | q_t = S_i, \lambda)P(q_t = S_i | \lambda)] P(O_{t+1:T} | q_t = S_i, \lambda)}{\sum_{j=1}^{N} P(O_{1:T} | q_t = S_j, \lambda)P(q_t = S_j | \lambda)} \]

**Numerator**: we combine the 1\textsuperscript{st} and the 3\textsuperscript{rd} term.

**Denominator**: We use the following observation to derive the denominator.
\[ P(a, b | c) = P(a | b, c) P(b | c) \]
Problem 2: Smoothing

Derivation of the following observation:

\[ P(a, b \mid c) = P(a \mid b, c) P(b \mid c) \]

\[
\begin{align*}
P(a, b, c) &= P(a, b \mid c) P(c) \\
P(a, b, c) &= P(a \mid b, c) P(b, c) \\
&= P(a \mid b, c) P(b \mid c) P(c) \\
P(a, b \mid c) P(c) &= P(a \mid b, c) P(b \mid c) P(c) \\
P(a, b \mid c) &= P(a \mid b, c) P(b \mid c)
\end{align*}
\]
Problem 2: Smoothing

\[ P(q_t = S_i \mid O_{1:T}, \lambda) = \frac{P(O_{1:T} \mid q_t = S_i, \lambda) P(q_t = S_i \mid \lambda)}{P(O_{1:T} \mid \lambda)} \]

\[ = \frac{[P(O_{1:t} \mid q_t = S_i, \lambda)P(q_t = S_i \mid \lambda)] P(O_{t+1:T} \mid q_t = S_i, \lambda)}{\sum_{j=1}^{N} P(O_{1:T} \mid q_t = S_j, \lambda)P(q_t = S_j \mid \lambda)} \]

Recall forward & backward variables:

\[ \alpha_t(i) \equiv P(O_1 O_2 \ldots O_t, q_t = S_i \mid \lambda) \]

\[ \beta_t(i) \equiv P(O_{t+1:T} \mid q_t = S_i, \lambda) \]

Numerator: we combined the 1st and the 2nd term.
Problem 2: Smoothing

The algorithm for computing smoothing is called the **Forward-Backward Algorithm**.

\[
P(q_t = S_i | O_{1:T}, \lambda) = \frac{P(O_{1:T} | q_t = S_i, \lambda) P(q_t = S_i | \lambda)}{P(O_{1:T} | \lambda)}
\]

\[
= \frac{P(O_{1:t}, q_t = S_i, \lambda) P(O_{t+1:T} | q_t = S_i, \lambda)}{\sum_{j=1}^{N} P(O_{1:T} | q_t = S_j, \lambda) P(q_t = S_j | \lambda)}
\]

\[
P(q_t = S_i | O_{1:T}, \lambda) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j) \beta_t(j)}
\]

**α_t(i) ≡ P(O_1 O_2 ... O_t, q_t = S_i | \lambda)**

**β_t(i) ≡ P(O_{t+1:T} | q_t = S_i, \lambda)**
Observe how nicely the $\alpha_t(i)$ and $\beta_t(i)$ **split the sequence** between them.

\[ P(q_t = S_i | O_{1:T}, \lambda) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j) \beta_t(j)} \]

$\alpha_t(i)$: The forward variable *(red line)* explains the **starting part** of the sequence until time $t$ and ends in $q_t = S_i$.

$\beta_t(i)$: The backward variable *(blue line)* explains the **ending part** until time $T$.

We need to **normalize by dividing this over all possible intermediate states** that can be traversed **at time t**.

The **numerator** $\alpha_t(i) \beta_t(i)$ explains the **whole sequence** given that at time $t$, the system in state $S_i$. 
The Forward-Backward algorithm is a special instance of the junction tree algorithm, which is an efficient exact method for inference over Bayesian networks.

\[
P(q_t = S_i \mid O_{1:T}, \lambda) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j) \beta_t(j)}
\]
Problem 2: Smoothing

- Why is smoothing useful?
- We use the following example to illustrate this.

\[
A = \begin{bmatrix}
\text{Rainy} & \text{Dry} \\
0.7 & 0.3 \\
0.3 & 0.7
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
\text{Umbrella} & \text{No Umbrella} \\
\text{Rainy} & 0.9 & 0.1 \\
\text{Dry} & 0.2 & 0.8
\end{bmatrix}
\]

\[
\Pi = \begin{bmatrix}
\text{Rainy} & \text{Dry} \\
0.5 & 0.5
\end{bmatrix}
\]

Observation = (Umbrella, Umbrella)

After observing umbrella on day 1, we compute the **probability of rain** on day 1.

Then, after observing umbrella on day 2, we compute the **smoothed probability** of rain on day 1.
After observing umbrella on day 1, we compute the **probability of rain on day 1**

Then, after observing umbrella on day 2, we compute the **smoothed probability** of rain on day 1

Notice, after observing umbrella on the second day, the **smoothed estimate for rain on day 1** is increased (0.883).

---

### Table: Probability of Rain and Umbrella

<table>
<thead>
<tr>
<th></th>
<th>Day 1 = Umbrella</th>
<th>Day 2 = Umbrella</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P(R_1 = \text{Rainy} \mid u_1) )</td>
<td>0.818</td>
<td>0.883</td>
</tr>
<tr>
<td>( P(R_1 = \text{Dry} \mid u_1) )</td>
<td>0.182</td>
<td>0.117</td>
</tr>
</tbody>
</table>

---

\[ A = \begin{array}{c|cc} \text{Rainy} & \text{Dry} \\ \hline \text{Rainy} & 0.7 & 0.3 \\ \text{Dry} & 0.3 & 0.7 \end{array} \]

\[ B = \begin{array}{c|cc} \text{Umbrella} & \text{No Umbrella} \\ \hline \text{Rainy} & 0.9 & 0.1 \\ \text{Dry} & 0.2 & 0.8 \end{array} \]

\[ \Pi = \begin{array}{c|cc} \text{Rainy} & \text{Dry} \\ \hline 0.5 & 0.5 \end{array} \]
Problem 2: Smoothing

- Smoothing is about **remaking past**.
- We revisit past and compute the probability of a hidden state.
- We "**smooth**" out our beliefs (mistakes)!
- Smoothing provides a **better estimate of the state than was available at the time**, because it incorporates more observations (evidence).
- In particular, when tracking a moving object with inaccurate position observations, **smoothing gives a smoother estimated trajectory** than filtering.
- That’s why smoothing is so named!