Hidden Markov Models
Algorithms for decoding

Christian Nørgaard Storm Pedersen
cstorm@birc.au.dk
HMM joint probability distribution

\[ p(X, Z|\Theta) = p(z_1|\pi) \left[ \prod_{n=2}^{N} p(z_n|z_{n-1}, A) \right] \prod_{n=1}^{N} p(x_n|z_n, \phi) \]

Observables: \( X = \{x_1, \ldots, x_N\} \)

Latent states: \( Z = \{z_1, \ldots, z_N\} \)

Model parameters: \( \Theta = \{\pi, A, \phi\} \)

If \( A \) and \( \phi \) are the same for all \( n \) then the HMM is homogeneous.
HMM joint probability distribution

\[ p(X, Z|\Theta) = p(z_1|\pi) \prod_{n=2}^{N} p(z_n|z_{n-1}, A) \prod_{n=1}^{N} p(x_n|z_n, \phi) \]

If A and \( \phi \) are the same for all \( n \) then the HMM is homogeneous
Decoding using HMMs

Given a HMM $\Theta$ and a sequence of observations $X = x_1, ..., x_N$, find a plausible explanation, i.e. a sequence $Z^* = z_1^*, ..., z_N^*$ of values of the hidden variable.

**Viterbi decoding**

$Z^*$ is the overall most likely explanation of $X$:

$$Z^* = \arg \max_Z p(X, Z | \Theta)$$

**Posterior decoding**

$z_n^*$ is the most likely state to be in the $n$'th step:

$$z_n^* = \arg \max_{z_n} p(z_n | x_1, \ldots, x_N)$$
Viterbi decoding

Given $X$, find $Z^*$ such that: $Z^* = \arg \max_Z p(X, Z | \Theta)$

\[
p(X, Z^*) = \max_Z p(X, Z) = \max_{z_1, \ldots, z_N} p(x_1, \ldots, x_N, z_1, \ldots, z_N)
\]
\[
= \max_{z_N} \max_{z_1, \ldots, z_{N-1}} p(x_1, \ldots, x_N, z_1, \ldots, z_N)
\]
\[
= \max_{z_N} \omega(z_N)
\]

$z_N^* = \arg \max_{z_N} \omega(z_N)$

Where $\omega(z_n) \equiv \max_{z_1, \ldots, z_{n-1}} p(x_1, \ldots, x_n, z_1, \ldots, z_n)$ is the probability of the most likely sequence of states $z_1, \ldots, z_n$ ending in $z_n$ generating the observations $x_1, \ldots, x_n$.
**Viterbi decoding**

Given \( \mathbf{X} \), find \( \mathbf{Z}^* \) such that: 
\[
\mathbf{Z}^* = \arg \max_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\Theta)
\]

\[
p(\mathbf{X}, \mathbf{Z}^*) = \max_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}) = \max_{\mathbf{z}_1, \ldots, \mathbf{z}_N} \max_{\mathbf{z}_N} p(\mathbf{x}_1, \ldots, \mathbf{x}_N)
\]

\[
= \max_{\mathbf{z}_N} \omega(\mathbf{z}_N)
\]

\[
\mathbf{z}_N^* = \arg \max_{\mathbf{z}_N} \omega(\mathbf{z}_N)
\]

Where \( \omega(\mathbf{z}_n) \equiv \max_{\mathbf{z}_1, \ldots, \mathbf{z}_{n-1}} p(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{z}_1, \ldots, \mathbf{z}_n) \) is the probability of the most likely sequence of states \( \mathbf{z}_1, \ldots, \mathbf{z}_n \) ending in \( \mathbf{z}_n \) generating the observations \( \mathbf{x}_1, \ldots, \mathbf{x}_n \)
The $\omega$-recursion

$$
\omega(z_n) = \max_{z_1, \ldots, z_{n-1}} p(x_1, \ldots, x_n, z_1, \ldots, z_n)
$$

$$
= \max_{z_1, \ldots, z_{n-1}} p(z_1) \prod_{i=2}^{n} p(z_i|z_{i-1}) \prod_{i=1}^{n} p(x_i|z_i)
$$

$$
= p(x_n|z_n) \max_{z_1, \ldots, z_{n-1}} p(z_1) \prod_{i=2}^{n} p(z_i|z_{i-1}) \prod_{i=1}^{n-1} p(x_i|z_i)
$$

$$
= p(x_n|z_n) \max_{z_{n-1}} \max_{z_1, \ldots, z_{n-2}} p(z_1)p(z_n|z_{n-1}) \prod_{i=2}^{n-1} p(z_i|z_{i-1}) \prod_{i=1}^{n-1} p(x_i|z_i)
$$

$$
= p(x_n|z_n) \max_{z_{n-1}} p(z_n|z_{n-1}) \max_{z_1, \ldots, z_{n-2}} p(z_1) \prod_{i=2}^{n-1} p(z_i|z_{i-1}) \prod_{i=1}^{n-1} p(x_i|z_i)
$$

$$
= p(x_n|z_n) \max_{z_{n-1}} p(z_n|z_{n-1}) \omega(z_{n-1})
$$
The $\omega$-recursion

$$\omega(z_n) = \max_{z_1, \ldots, z_{n-1}} p(x_1, \ldots, x_n, z_1, \ldots, z_n)$$

$$= \max_{z_1, \ldots, z_{n-1}} p(z_1) \prod_{i=2}^{n} p(z_i | z_{i-1}) \prod_{i=1}^{n} p(x_i | z_i)$$

$$= p(x_n | z_n) \max_{z_1, \ldots, z_{n-1}} p(z_1) \prod_{i=2}^{n} p(z_i | z_{i-1}) \prod_{i=1}^{n} p(x_i | z_i)$$

$$= p(x_n | z_n) \max_{z_1, \ldots, z_{n-1}} p(z_1) p(z_n | z_{n-1}) \prod_{i=2}^{n-1} p(z_i | z_{i-1}) \prod_{i=1}^{n-1} p(x_i | z_i)$$

$$= p(x_n | z_n) \max_{z_{n-1}} p(z_{n-1}) \max_{z_1, \ldots, z_{n-2}} p(z_1) p(z_n | z_{n-1}) \prod_{i=2}^{n-1} p(z_i | z_{i-1}) \prod_{i=1}^{n-1} p(x_i | z_i)$$

$$= p(x_n | z_n) \max_{z_{n-1}} p(z_{n-1}) \omega(z_{n-1})$$
The $\omega$-recursion

$\omega(z_n)$ is the probability of the most likely sequence of states $z_1, \ldots, z_n$ ending in $z_n$ generating the observations $x_1, \ldots, x_n$

$$\omega(z_n) \equiv \max_{z_1, \ldots, z_{n-1}} p(x_1, \ldots, x_n, z_1, \ldots, z_n)$$

Recursion:

$$\omega(z_n) = p(x_n | z_n) \max_{z_{n-1}} \omega(z_{n-1}) p(z_n | z_{n-1})$$

Basis:

$$\omega(z_1) = p(x_1, z_1) = p(z_1) p(x_1 | z_1)$$
The $\omega$-recursion

// Pseudo code for computing $\omega[k][n]$ for some $n>1$

$\omega[k][n] = 0$

for $j = 1$ to $K$:

$\omega[k][n] = \max( \omega[k][n], p(x[n] \mid k) \times \omega[j][n-1] \times p(k \mid j) )$

$$\omega(z_n) \equiv \max_{z_1, \ldots, z_{n-1}} p(x_1, \ldots, x_n, z_1, \ldots, z_n)$$

Recursion:

$$\omega(z_n) = p(x_n \mid z_n) \max_{z_{n-1}} \omega(z_{n-1}) p(z_n \mid z_{n-1})$$

Basis:

$$\omega(z_1) = p(x_1, z_1) = p(z_1) p(x_1 \mid z_1)$$
The $\omega$-recursion

// Pseudo code for computing $\omega[k][n]$ for some $n>1$

$\omega[k][n] = 0$

for $j = 1$ to $K$:

$\omega[k][n] = \max( \omega[k][n], p(x[n] | k) \times \omega[j][n-1] \times p(k | j) )$

$\omega(z_n) \equiv \max_{z_1, \ldots, z_{n-1}} p(x_1, \ldots, x_n, z_1, \ldots, z_n)$

Recursion:

$\omega(z_n) = p(x_n | z_n) \max_{z_{n-1}} \omega(z_{n-1}) p(z_n | z_{n-1})$

Basis:

$\omega(z_1) = p(x_1, z_1) = p(z_1) p(x_1 | z_1)$

Computing $\omega$ takes time $O(K^2N)$ and space $O(KN)$ using memorization
ω(z_n) is the probability of the most likely sequence of states z_1,...,z_n ending in z_n generating the observations x_1,...,x_n. We find Z* by backtracking:

\[
\begin{align*}
Z_N^* &= \arg \max_{Z_N} \omega(Z_N) = \arg \max_{Z_N} \max_{Z_{N-1}} (p(x_N|Z_N) \omega(z_{n-1}) p(Z_N|Z_{N-1})) \\
Z_{N-1}^* &= \arg \max_{Z_{N-1}} (p(x_N|Z_N^*) \omega(z_{N-1}) p(Z_N^*|Z_{N-1})) \\
& \vdots
\end{align*}
\]

ω[k][n] = ω(z_n) if z_n is state k
\[ \omega(z^n) \] is the probability of the most likely sequence of states \( z_1, \ldots, z_n \) generating the observations \( x_1, \ldots, x_n \).

We find \( Z^* \) by backtracking:

\[
\begin{align*}
Z^*_N &= \text{arg max}_k \omega[k][N] \\
Z^*_{n-1} &= \text{arg max}_{z_{n-1}} \left( p(x_{n-1} \mid z_{n-1}) \cdot \omega(z_n \mid z_{n-1}) \cdot p(z_n \mid z_{n-1}) \right)
\end{align*}
\]

\[
\omega[k][n] = \omega(z_n) \text{ if } z_n \text{ is state } k
\]

\[
\text{Pseudocode for backtracking}
\]

```plaintext
// Pseudocode for backtracking
z[1..N] = undef
z[N] = arg max_k \omega[k][N]

for n = N-1 to 1:
    z[n] = arg max_k ( p(x[n+1] \mid z[n+1]) \cdot \omega[k][n] \cdot p(z[n+1] \mid k) )

print z[1..N]
```
\( \omega(z^n) \) is the probability of the most likely sequence of states \( z_1, \ldots, z_n \) ending in \( z_n \) generating the observations \( x_1, \ldots, x_n \). We find \( Z^* \) by backtracking:

\[
\begin{align*}
\omega[k][n] &= \omega(z^n) \text{ if } z_n \text{ is state } k \\
\text{Backtracking takes time } O(KN) \text{ and space } O(KN) \text{ using } \omega 
\end{align*}
\]
Decoding using HMMs

Given a HMM $\Theta$ and a sequence of observations $X = x_1, ..., x_N$, find a plausible explanation, i.e. a sequence $Z^* = z_1^*, ..., z_N^*$ of values of the hidden variable.

**Viterbi decoding**

$Z^*$ is the overall most likely explanation of $X$:

$$Z^* = \arg \max_Z p(X, Z | \Theta)$$

**Posterior decoding**

$z_n^*$ is the most likely state to be in the $n$'th step:

$$z_n^* = \arg \max_{z_n} p(z_n | x_1, \ldots, x_N)$$
**Posterior decoding**

Given $\mathbf{X}$, find $\mathbf{Z}^*$, where $z^*_n = \arg \max_{z_n} p(z_n | x_1, \ldots, x_N)$ is the most likely state to be in the $n$'th step.

$$p(z_n | x_1, \ldots, x_N) = \frac{p(z_n, x_1, \ldots, x_N)}{p(x_1, \ldots, x_N)}$$

$$= \frac{p(x_1, \ldots, x_{n-1}, z_n, x_n, x_{n+1}, \ldots, x_N)}{p(x_1, \ldots, x_N)}$$

$$= \frac{p(x_1, \ldots, x_{n-1}, z_n, x_n, x_{n+1}, \ldots, x_N | z_n)}{p(x_1, \ldots, x_N | z_n)}$$

$$= \frac{\alpha(z_n) \beta(z_n)}{p(X)}$$

$$z^*_n = \arg \max_{z_n} p(z_n | x_1, \ldots, x_N) = \arg \max_{z_n} \alpha(z_n) \beta(z_n) / p(X)$$
Posterior decoding

$\alpha(z_n)$ is the joint probability of observing $x_1, \ldots, x_n$ and being in state $z_n$

$$\alpha(z_n) \equiv p(x_1, \ldots, x_n, z_n)$$

$\beta(z_n)$ is the conditional probability of future observation $x_{n+1}, \ldots, x_N$ assuming being in state $z_n$

$$\beta(z_n) \equiv p(x_{n+1}, \ldots, x_N | z_n)$$

$\alpha[k][n] = \alpha(z_n)$ if $z_n$ is state $k$

$\beta[k][n] = \beta(z_n)$ if $z_n$ is state $k$
Posterior decoding

\( \alpha(z_n) \) is the joint probability of observing \( x_1, \ldots, x_n \) and being in state \( z_n \)

\[
\alpha(z_n) \equiv p(x_1, \ldots, x_n, z_n)
\]

\( \beta(z_n) \) is the conditional probability of future observation \( x_{n+1}, \ldots, x_N \) assuming being in state \( z_n \)

\[
\beta(z_n) \equiv p(x_{n+1}, \ldots, x_N | z_n)
\]

Using \( \alpha(z_n) \) and \( \beta(z_n) \) we get the likelihood of the observations as:

\[
p(X) = \sum_{z_n} \alpha(z_n) \beta(z_n)
\]

\[
p(X) = \sum_{z_N} \alpha(z_N)
\]

\[
z^*_n = \arg \max_{z_n} p(z_n | x_1, \ldots, x_N) = \arg \max_{z_n} \alpha(z_n) \beta(z_n) / p(X)
\]
The forward algorithm

$\alpha(z_n)$ is the joint probability of observing $x_1, \ldots, x_n$ and being in state $z_n$

$$\alpha(z_n) \equiv p(x_1, \ldots, x_n, z_n)$$

$\alpha[k][n] = \alpha(z_n)$ if $z_n$ is state $k$
The $\alpha$-recursion

\[
\alpha(z_n) = p(x_1, \ldots, x_n, z_n)
\]

\[
= \sum_{z_1, \ldots, z_{n-1}} p(x_1, \ldots, x_n, z_1, \ldots, z_n)
\]

\[
= \sum_{z_1, \ldots, z_{n-1}} p(z_1) \prod_{i=2}^{n} p(z_i | z_{i-1}) \prod_{i=1}^{n} p(x_i | z_i)
\]

\[
= p(x_n | z_n) \sum_{z_1, \ldots, z_{n-1}} p(z_1) \prod_{i=2}^{n} p(z_i | z_{i-1}) \prod_{i=1}^{n-1} p(x_i | z_i)
\]

\[
= p(x_n | z_n) \sum_{z_{n-1}} \sum_{z_1, \ldots, z_{n-2}} p(z_1) p(z_n | z_{n-1}) \prod_{i=2}^{n-1} p(z_i | z_{i-1}) \prod_{i=1}^{n-1} p(x_i | z_i)
\]

\[
= p(x_n | z_n) \sum_{z_{n-1}} p(z_n | z_{n-1}) \sum_{z_1, \ldots, z_{n-2}} p(z_1) \prod_{i=2}^{n-1} p(z_i | z_{i-1}) \prod_{i=1}^{n-1} p(x_i | z_i)
\]

\[
= p(x_n | z_n) \sum_{z_{n-1}} p(z_n | z_{n-1}) \alpha(z_{n-1})
\]
The $\alpha$-recursion

\[
\alpha(z_n) = p(x_1, \ldots, x_n, z_n) \\
= \sum_{z_1, \ldots, z_{n-1}} p(x_1, \ldots, x_n, z_1, \ldots, z_n) \\
= \sum_{z_1, \ldots, z_{n-1}} p(z_1) \prod_{i=2}^{n} p(z_i | z_{i-1}) \prod_{i=1}^{n} p(x_i | z_i) \\
= p(x_n | z_n) \sum_{z_1, \ldots, z_{n-1}} p(z_1) \prod_{i=2}^{n} p(z_i | z_{i-1}) \prod_{i=1}^{n} p(x_i | z_i) \\
= p(x_n | z_n) \sum_{z_{n-1}} \sum_{z_1, \ldots, z_{n-2}} p(z_1) p(z_n | z_{n-1}) \prod_{i=2}^{n-1} p(z_i | z_{i-1}) \prod_{i=1}^{n-1} p(x_i | z_i) \\
= p(x_n | z_n) \sum_{z_{n-1}} p(z_n | z_{n-1}) \sum_{z_1, \ldots, z_{n-2}} p(z_1) \prod_{i=2}^{n} p(z_i | z_{i-1}) \prod_{i=1}^{n} p(x_i | z_i) \\
= p(x_n | z_n) \sum_{z_{n-1}} p(z_n | z_{n-1}) \alpha(z_{n-1})
\]
The forward algorithm

\( \alpha(z_n) \) is the joint probability of observing \( x_1, \ldots, x_n \) and being in state \( z_n \)

\[
\alpha(z_n) \equiv p(x_1, \ldots, x_n, z_n)
\]

**Recursion:**

\[
\alpha(z_n) = p(x_n | z_n) \sum_{z_{n-1}} \alpha(z_{n-1}) p(z_n | z_{n-1})
\]

**Basis:**

\[
\alpha(z_1) = p(x_1, z_1) = p(z_1) p(x_1 | z_1)
\]

\( \alpha[k][n] = \alpha(z_n) \) if \( z_n \) is state \( k \)
The forward algorithm

// Pseudo code for computing $\alpha[k][n]$ for some $n>1$
$\alpha[k][n] = 0$
for $j = 1$ to $K$:
    $\alpha[k][n] = \alpha[k][n] + p(x[n] | k) \ast \alpha[j][n-1] \ast p(k | j)$

Recursion:

$\alpha(z_n) = p(x_n | z_n) \sum_{z_{n-1}} \alpha(z_{n-1}) p(z_n | z_{n-1})$

Basis:

$\alpha(z_1) = p(x_1, z_1) = p(z_1) p(x_1 | z_1)$

$\alpha[k][n] = \alpha(z_n)$ if $z_n$ is state $k$
The forward algorithm

// Pseudo code for computing $\alpha[k][n]$ for some $n>1$
$\alpha[k][n] = 0$
for $j = 1$ to $K$:
    $\alpha[k][n] = \alpha[k][n] + p(x[n] \mid k) \times \alpha[j][n-1] \times p(k \mid j)$

Recursion:

$\alpha(z_n) = p(x_n \mid z_n) \sum_{z_{n-1}} \alpha(z_{n-1}) p(z_n \mid z_{n-1})$

Basis:

$\alpha(z_1) = p(x_1, z_1) = p(z_1) p(x_1 \mid z_1)$

Computing $\alpha$ takes time $O(K^2N)$ and space $O(KN)$ using memorization.

being in state $z_n$
The backward algorithm

$\beta(z_n)$ is the conditional probability of future observation $x_{n+1}, \ldots, x_N$ assuming being in state $z_n$

$$\beta(z_n) \equiv p(x_{n+1}, \ldots, x_N | z_n)$$

$\beta[k][n] = \beta(z_n)$ if $z_n$ is state $k$
The $\beta$-recursion

$$\beta(z_n) = p(x_{n+1}, \ldots, x_N | z_n)$$
$$= \sum_{z_{n+1}, \ldots, z_N} p(x_{n+1}, \ldots, x_N, z_{n+1}, \ldots, z_N | z_n)$$
$$= \sum_{z_{n+1}, \ldots, z_N} p(x_{n+1}, \ldots, x_N, z_n, z_{n+1}, \ldots, z_N) / p(z_n)$$
$$= \sum_{z_{n+1}, \ldots, z_N} p(z_n) \prod_{i=n+1}^{N} p(z_i | z_{i-1}) \prod_{i=n+1}^{N} p(x_i | z_i) / p(z_n)$$
$$= \sum_{z_{n+1}, \ldots, z_N} \prod_{i=n+1}^{N} p(z_i | z_{i-1}) \prod_{i=n+1}^{N} p(x_i | z_i)$$
$$= \sum_{z_{n+1} \ldots, z_N} \sum_{z_{n+1}} p(z_{n+1} | z_n) p(x_{n+1} | z_{n+1}) \prod_{i=n+2}^{N} p(z_i | z_{i-1}) \prod_{i=n+2}^{N} p(x_i | z_i)$$
$$= \sum_{z_{n+1}} p(z_{n+1} | z_n) p(x_{n+1} | z_{n+1}) \sum_{z_{n+2}, \ldots, z_N} \prod_{i=n+2}^{N} p(z_i | z_{i-1}) \prod_{i=n+2}^{N} p(x_i | z_i)$$
$$= \sum_{z_{n+1}} p(z_{n+1} | z_n) p(x_{n+1} | z_{n+1}) p(x_{n+2}, \ldots, x_N | z_{n+1})$$
$$= \sum_{z_{n+1}} p(z_{n+1} | z_n) p(x_{n+1} | z_{n+1}) \beta(z_{n+1})$$
The $\beta$-recursion

\[
\beta(z_n) = p(x_{n+1}, \ldots, x_N | z_n)
\]

\[
= \sum_{z_{n+1}, \ldots, z_N} p(x_{n+1}, \ldots, x_N, z_{n+1}, \ldots, z_N | z_n)
\]

\[
= \sum_{z_{n+1}, \ldots, z_N} p(x_{n+1}, \ldots, x_N, z_n, z_{n+1}, \ldots, z_N)/p(z_n)
\]

\[
= \sum_{z_{n+1}, \ldots, z_N} p(z_n) \prod_{i=n+1}^N p(z_i | z_{i-1}) \prod_{i=n+1}^N p(x_i | z_i)/p(z_n)
\]

\[
= \sum_{z_{n+1}, \ldots, z_N} \prod_{i=n+1}^N p(z_i | z_{i-1}) \prod_{i=n+1}^N p(x_i | z_i)
\]

\[
= \sum_{z_{n+1}, z_{n+2}, \ldots, z_N} p(z_{n+1} | z_n)p(x_{n+1} | z_{n+1}) \sum_{z_{n+2}, \ldots, z_N} \prod_{i=n+2}^N p(z_i | z_{i-1}) \prod_{i=n+2}^N p(x_i | z_i)
\]

\[
= \sum_{z_{n+1}} p(z_{n+1} | z_n)p(x_{n+1} | z_{n+1}) \sum_{z_{n+2}, \ldots, z_N} \prod_{i=n+2}^N p(z_i | z_{i-1}) \prod_{i=n+2}^N p(x_i | z_i)
\]

\[
= \sum_{z_{n+1}} p(z_{n+1} | z_n)p(x_{n+1} | z_{n+1}) \beta(z_{n+1})
\]
The backward algorithm

$\beta(z_n)$ is the conditional probability of future observation $x_{n+1}, \ldots, x_N$ assuming being in state $z_n$

$$\beta(z_n) \equiv p(x_{n+1}, \ldots, x_N | z_n)$$

Recursion:

$$\beta(z_n) = \sum_{z_{n+1}} \beta(z_{n+1}) p(x_{n+1} | z_{n+1}) p(z_{n+1} | z_n)$$

Basis:

$$\beta(z_N) = 1$$

$\beta[k][n] = \beta(z_n)$ if $z_n$ is state $k$
The backward algorithm

// Pseudo code for computing $\beta[k][n]$ for some $n< N$
$\beta[k][n] = 0$
for $j = 1$ to $K$:
    $\beta[k][n] = \beta[k][n] + p(j | k) \times p(x[n+1] | j) \times \beta[j][n+1]$

$\beta(z_n) \equiv p(x_{n+1}, \ldots, x_N | z_n)$

Recursion:

$$\beta(z_n) = \sum_{z_{n+1}} \beta(z_{n+1}) p(x_{n+1} | z_{n+1}) p(z_{n+1} | z_n)$$

Basis:

$$\beta(z_N) = 1$$

$\beta[k][n] = \beta(z_n)$ if $z_n$ is state $k$
The backward algorithm

// Pseudo code for computing $\beta[k][n]$ for some $n<N$

$\beta[k][n] = 0$

for $j = 1$ to $K$:

$\beta[k][n] = \beta[k][n] + p(j \mid k) \cdot p(x[n+1] \mid j) \cdot \beta[j][n+1]$

$\beta(z_n) \equiv p(x_{n+1}, \ldots, x_N \mid z_n)$

Recursion:

$\beta(z_n) = \sum_{z_{n+1}} \beta(z_{n+1}) p(x_{n+1} \mid z_{n+1}) p(z_{n+1} \mid z_n)$

Basis:

$\beta(z_N) = 1$

Computing $\beta$ takes time $O(K^2N)$ and space $O(KN)$ using memorization.
Posterior decoding

$\alpha(z_n)$ is the joint probability of observing $x_1, \ldots, x_n$ and being in state $z_n$

$$\alpha(z_n) \equiv p(x_1, \ldots, x_n, z_n)$$

$\beta(z_n)$ is the conditional probability of future observation $x_{n+1}, \ldots, x_N$ assuming being in state $z_n$

$$\beta(z_n) \equiv p(x_{n+1}, \ldots, x_N | z_n)$$

Using $\alpha(z_n)$ and $\beta(z_n)$ we get the likelihood of the observations as:

$$p(X) = \sum_{z_n} \alpha(z_n) \beta(z_n)$$

$$p(X) = \sum_{z_N} \alpha(z_N)$$

$$z_n^* = \arg\max_{z_n} p(z_n | x_1, \ldots, x_N) = \arg\max_{z_n} \alpha(z_n) \beta(z_n) / p(X)$$
Posterior decoding

\[ \alpha(z_n) \] is the joint probability of observing \( x_1, \ldots, x_n \) and being in state \( z_n \)

\[ \beta(z_n) \] is the conditional probability of future observation \( x_{n+1}, \ldots, x_N \) assuming being in state \( z_n \)

Using \( \alpha(z_n) \) and \( \beta(z_n) \) we get the likelihood of the observations as:

\[
p(X) = \sum_{z_n} \alpha(z_n) \beta(z_n)
\]

\[
p(X) = \sum_{z_N} \alpha(z_N)
\]

\[
z^*_n = \arg \max_{z_n} p(z_n | x_1, \ldots, x_N) = \arg \max_{z_n} \frac{\alpha(z_n) \beta(z_n)}{p(X)}
\]
Viterbi vs. Posterior decoding

A sequence of states $z_1, \ldots, z_N$ where $p(x_1, \ldots, x_N, z_1, \ldots, z_N) > 0$ is a legal (or syntactically correct) decoding of $X$.

Viterbi finds the most likely syntactically correct decoding of $X$.

What does Posterior decoding find?

Does it always find a syntactically correct decoding of $X$?
A sequence of states $z_1, \ldots, z_N$ where $p(x_1, \ldots, x_N, z_1, \ldots, z_N) > 0$ is a legal (or syntactically correct) decoding of $X$.

Viterbi finds the most likely syntactically correct decoding of $X$.

What does Posterior decoding find?

Does it always find a syntactically correct decoding of $X$?

Emits a sequence of A and Bs following either the path 12....2 or 13....3 with equal probability

I.e. Viterbi finds either 12...2 or 13...3, while Posterior finds that 2 and 3 are equally likely for $n > 1$. 
Recall: Using HMMs

- Determine the likelihood of a sequence of observations.
- Find a plausible underlying explanation (or decoding) of a sequence of observations.

\[
p(X|\Theta) = \sum_{Z} p(X, Z|\Theta) = \sum_{Z_N} \alpha(z_N)
\]

The sum has \(K^N\) terms, but it turns out that it can be computed in \(O(K^2N)\) time by computing the \(\alpha\)-table using the forward algorithm and summing the last column:

\[
p(X) = \alpha[1][N] + \alpha[2][N] + \ldots + \alpha[K][N]
\]
Summary

- **Viterbi-** and **Posterior decoding** for finding a plausible underlying explanation (sequence of hidden states) of a sequence of observation

- **forward-backward algorithms** for computing the likelihood of being in a given state in the \( n \)'th step, and for determining the likelihood of a sequence of observations.
Viterbi

Recursion: \[ \omega(z_n) = p(x_n | z_n) \max_{z_{n-1}} \omega(z_{n-1}) p(z_n | z_{n-1}) \]

Basis: \[ \omega(z_1) = p(x_1, z_1) = p(z_1) p(x_1 | z_1) \]

Forward

Recursion: \[ \alpha(z_n) = p(x_n | z_n) \sum_{z_{n-1}} \alpha(z_{n-1}) p(z_n | z_{n-1}) \]

Basis: \[ \alpha(z_1) = p(x_1, z_1) = p(z_1) p(x_1 | z_1) \]

Backward

Recursion: \[ \beta(z_n) = \sum_{z_{n+1}} \beta(z_{n+1}) p(x_{n+1} | z_{n+1}) p(z_{n+1} | z_n) \]

Basis: \[ \beta(z_N) = 1 \]
**Problem:** The values in the $\omega$-, $\alpha$-, and $\beta$-tables can come very close to zero, by multiplying them we potentially exceed the precision of double precision floating points and get underflow.

**Next:** How to implement the basic algorithms (forward, backward, and Viterbi) in a “numerically” sound manner.

**Recursion:**  
\[
\alpha(z_n) = p(x_n|z_n) \sum_{z_{n-1}} \alpha(z_{n-1})p(z_n|z_{n-1})
\]

**Basis:**  
\[
\alpha(z_1) = p(x_1, z_1) = p(z_1)p(x_1|z_1)
\]

**Backward**

**Recursion:**  
\[
\beta(z_n) = \sum_{z_{n+1}} \beta(z_{n+1})p(x_{n+1}|z_{n+1})p(z_{n+1}|z_n)
\]

**Basis:**  
\[
\beta(z_N) = 1
\]