Overview

HMMs

- Introduction to HMMs models
- HMMs for ASR
- Likelihood computation with the forward algorithm
If $X$ is the sequence of acoustic feature vectors (observations) and $W$ denotes a word sequence, the most likely word sequence $W^*$ is given by

$$W^* = \arg \max_W P(W | X)$$
If \( X \) is the sequence of acoustic feature vectors (observations) and \( W \) denotes a word sequence, the most likely word sequence \( W^* \) is given by

\[
W^* = \arg \max_W P(W | X)
\]

Applying Bayes’ Theorem:

\[
P(W | X) = \frac{p(X | W) P(W)}{p(X)}
\]

\[
\propto p(X | W) P(W)
\]

\[
W^* = \arg \max_W p(X | W) \underbrace{P(W)}_{\text{Acoustic model}} \underbrace{P(W)}_{\text{Language model}}
\]

NB: \( X \) is used hereafter to denote the output feature vectors from the signal analysis module rather than DFT spectrum.
Acoustic Modelling

Recorded Speech

Signal Analysis

Hidden Markov model

Acoustic Model

Language Model

Search Space

Decoded Text (Transcription)

Training Data

ASR Lecture 4

Introduction to Hidden Markov Models

4
Hierarchical modelling of speech

Generative Model

"No right"

NO

RIGHT

n

oh

r

ai

t

Utterance W

Word

Subword

HMM

Acoustics X

ASR Lecture 4

Introduction to Hidden Markov Models
A statistical model for time series data with a set of discrete states \{1, \ldots, J\} (we index them by \(j\) or \(k\))

At each time step \(t\):
- the model is in a fixed state \(q_t\).
- the model generates an observation, \(x_t\), according to a probability distribution that is specific to the state.

We don’t actually observe which state the model is in at each time step – hence “hidden”.

Observations can be either continous or discrete (usually the former).
Imagine we know the state at a given time step $t$, $q_t = k$

Then the probability of being in a new state, $j$ at the next time step, is dependent only on $q_t$. This is the **Markov** assumption.

Alternatively: $q_{t+1}$ is *conditionally independent* of $q_1, \ldots, q_{t-1}$, given $q_t$. 
Note that **observation independence** is an assumption that naturally arises from the model: the probability of $x_t$ depends only on the state that generated it, $q_t$. 

$$P(x_{t-1}|q_{t-1}) \quad P(x_{t+1}|q_{t+1})$$

$$P(x_t|q_t)$$

$${q_{t-1} \rightarrow q_t \rightarrow q_{t+1}}$$

$$P(q_t|q_{t-1}) \quad P(q_{t+1}|q_t)$$

$${x_{t-1} \rightarrow x_t \rightarrow x_{t+1}}$$
The HMM topology determines the set of allowed transitions between states.
HMM topologies

- The HMM topology determines the set of allowed transitions between states
- In principle any topology is possible
HMM topologies

- The HMM topology determines the set of allowed transitions between states
- In principle any topology is possible
The HMM topology determines the set of allowed transitions between states.

In principle any topology is possible.

Not all transition probabilities are shown.
Example topologies

left-to-right model

parallel path left-to-right model

ergodic model

Speech recognition: left-to-right HMM with 3 ~ 5 states

Speaker recognition: ergodic HMM
We generally model words or phones with a left-to-right topology with self loops.
Traditional HMMs for ASR tend to model each phone with three distinct states (this also enforces a minimum phone duration of three frames of observations)
Traditional HMMs for ASR tend to model each phone with three distinct states (this also enforces a minimum phone duration of three frames of observations)
Traditional HMMs for ASR tend to model each phone with three distinct states (this also enforces a minimum phone duration of three frames of observations)

The phone model topologies can be concatenated to form a HMM for the whole word
Traditional HMMs for ASR tend to model each phone with three distinct states (this also enforces a minimum phone duration of three frames of observations)

This model naturally generates an alignment between states and observations (and hence words/phones).
Suppose we have a sequence of observations of length $T$, $X = (x_1, \ldots, x_T)$, and $Q$ is a known state sequence, $(q_1, \ldots, q_T)$. Then we can use the HMM to compute the joint likelihood of $X$ and $Q$:

$$P(X, Q; \lambda) = P(q_1)P(x_1|q_1)P(q_2|q_1)P(x_2|q_2) \ldots \tag{1}$$

$$= P(q_1)P(x_1|q_1) \prod_{t=2}^{T} P(q_t|q_{t-1})P(x_t|q_t) \tag{2}$$

$P(q_1)$ denotes the initial occupancy probability of each state.
The parameters of the model, \( \lambda \), are given by:

- Transition probabilities
  \[ a_{kj} = P(q_{t+1} = j | q_t = k) \]

- Observation probabilities
  \[ b_j(x) = P(x | q = j) \]
The three problems of HMMs

Working with HMMs requires the solution of three problems:

1. **Likelihood** Determine the overall likelihood of an observation sequence \( X = (x_1, \ldots, x_t, \ldots, x_T) \) being generated by a known HMM topology, \( \mathcal{M} \).
Working with HMMs requires the solution of three problems:

1. **Likelihood** Determine the overall likelihood of an observation sequence $X = (x_1, \ldots, x_t, \ldots, x_T)$ being generated by a known HMM topology, $M$.

2. **Decoding and alignment** Given an observation sequence and an HMM, determine the most probable hidden state sequence
The three problems of HMMs

Working with HMMs requires the solution of three problems:

1. **Likelihood** Determine the overall likelihood of an observation sequence \( X = (x_1, \ldots, x_t, \ldots, x_T) \) being generated by a known HMM topology, \( \mathcal{M} \).

2. **Decoding and alignment** Given an observation sequence and an HMM, determine the most probable hidden state sequence.

3. **Training** Given an observation sequence and an HMM, find the state occupation probabilities.
Computing likelihood

1 Likelihood Determine the overall likelihood of an observation sequence \( X = (x_1, \ldots, x_t, \ldots, x_T) \) being generated by a known HMM topology, \( \mathcal{M} \).

→ the forward algorithm

NB. We do not know the state sequence!
By the HMM topology, $\mathcal{M}$, we can mean:

- A restricted left-to-right topology based on a known word/sentence, leading to a “trellis-like” structure over time
- A much less restricted topology based on a grammar or language model – or something in between
- Some algorithms are not (generally) suitable for unrestricted topologies
Example: trellis for a 3-state phone HMM
Goal: determine $p(X|M)$

$\sum_{Q \in Q} P(X, Q|M) = P(q_1) P(x_1|q_1) \prod_{t=2}^{T} P(q_t|q_{t-1}) P(x_t|q_t)$

How many paths $Q$ do we have to calculate?

$\sim N \times N \times \ldots \times N \times N \times \ldots \times N \times N = N^T$ times

$N$: number of HMM states

$T$: length of observation

e.g. $N^T \approx 10^{10}$ for $N = 3$, $T = 20$

Computation complexity of multiplication: $O(2^TN^T)$
Goal: determine $p(X|\mathcal{M})$

Sum over all possible state sequences $Q = (q_1, \ldots, q_T)$ that could result in the observation sequence $X$

$$p(X|\mathcal{M}) = \sum_{Q \in \mathcal{Q}} P(X, Q|\mathcal{M})$$

$$= P(q_1)P(x_1|q_1) \prod_{t=2}^{T} P(q_t|q_{t-1})P(x_t|q_t)$$
Likelihood

- Goal: determine $p(X|M)$
- Sum over all possible state sequences $Q = (q_1, \ldots, q_T)$ that could result in the observation sequence $X$

$$p(X|M) = \sum_{Q \in Q} P(X, Q|M)$$

$$= P(q_1)P(x_1|q_1) \prod_{t=2}^{T} P(q_t|q_{t-1})P(x_t|q_t)$$

- How many paths $Q$ do we have to calculate?

$$\sim N \times N \times \cdots N = N^T$$

$N$: number of HMM states

$T$: length of observation

e.g. $N^T \approx 10^{10}$ for $N=3, T=20$
Likelihood

- Goal: determine $p(X|\mathcal{M})$
- Sum over all possible state sequences $Q = (q_1, \ldots, q_T)$ that could result in the observation sequence $X$

$$p(X|\mathcal{M}) = \sum_{Q \in \mathcal{Q}} P(X, Q|\mathcal{M})$$

$$= P(q_1)P(x_1|q_1) \prod_{t=2}^{T} P(q_t|q_{t-1})P(x_t|q_t)$$

- How many paths $Q$ do we have to calculate?

$$\sim N \times N \times \cdots N = N^T$$

$N$ : number of HMM states
$T$ : length of observation

e.g. $N^T \approx 10^{10}$ for $N = 3$, $T = 20$

- Computation complexity of multiplication: $O(2TN^T)$
The Forward algorithm:

- Rather than enumerating each sequence, compute the probabilities recursively (exploiting the Markov assumption)
The **Forward algorithm**: 

- Rather than enumerating each sequence, compute the probabilities recursively (exploiting the Markov assumption)
- Reduces the computational complexity to $O(TN^2)$
Likelihood: The Forward algorithm

The **Forward algorithm**:

- Rather than enumerating each sequence, compute the probabilities recursively (exploiting the Markov assumption)
- Reduces the computational complexity to $O(TN^2)$
- Visualise the problem as a *state-time trellis*
Define the *Forward probability*, $\alpha_t(j)$: the probability of observing the observation sequence $x_1 \ldots x_t$ and being in state $j$ at time $t$:

$$\alpha_j(t) = p(x_1, \ldots, x_t, q_t = j | M)$$

We can *recursively* compute this probability.
Initial and final state probabilities

It what follows it is convenient to define:

- an additional single initial state $S_I = 0$, with transition probabilities
  \[ a_{0j} = P(q_1 = j) \]
  denoting the probability of starting in state $j$

- a single final state, $S_E$, with transition probabilities $a_{jE}$
  denoting the probability of the model terminating in state $j$.

- $S_I$ and $S_E$ are both *non-emitting*
Likelihood: The Forward recursion

- **Initialisation**

\[
\alpha_j(0) = 1 \quad j = 0 \\
\alpha_j(0) = 0 \quad j \neq 0
\]

- **Recursion**

\[
\alpha_j(t) = \sum_{i=1}^{J} \alpha_i(t-1) a_{ij} b_j(x_t) \quad 1 \leq j \leq J, \ 1 \leq t \leq T
\]

- **Termination**

\[
p(X | \mathcal{M}) = \alpha_E = \sum_{i=1}^{J} \alpha_i(T) a_{iE}
\]

- $s_I$: initial state, $s_E$: final state
\[ \alpha_j(t) = p(x_1, \ldots, x_t, q_t = j|\mathcal{M}) = \sum_{i=1}^{J} \alpha_i(t-1) a_{ij} b_j(x_t) \]
More HMM algorithms

- Finding the most likely path with the Viterbi algorithm
- Parameter estimation:
  - the Forward-Backward algorithm
  - the Expectation-Maximisation algorithm

