Hidden Markov Models
and
Gaussian Mixture Models

Peter Bell

Automatic Speech Recognition— ASR Lecture 2
16 January 2020
Overview

HMMs and GMMs

- Introduction to HMMs: Hidden Markov models
- Univariate and multivariate Gaussians
- Gaussian mixture models
- Introduction to the EM algorithm

Warning: the maths starts here!
Overview

HMMs and GMMs

- Introduction to HMMs: Hidden Markov models
- Univariate and multivariate Gaussians
- Gaussian mixture models
- Introduction to the EM algorithm

Warning: the maths starts here!
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:

$$P(W | X) = \frac{p(X | W) P(W)}{p(X)} \propto p(X | W) P(W)$$

$W^* = \arg \max_W p(X | W) P(W)$
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:

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

$$\propto p(X | W) P(W)$$

$$W^* = \arg \max_W p(X | W) P(W)$$

Acoustic model \hspace{1cm} 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 → Hidden Markov model → Acoustic Model → Search Space → Decoded Text (Transcription)

- Signal Analysis
- Training Data
- Language Model
Hierarchical modelling of speech

Generative Model

"No right"

Utterance $W$

Word

Subword

HMM

Acoustics $X$

ASR Lecture 2

Hidden Markov Models and Gaussian Mixture 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 continuous 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 \( q_t \).

This means we can parametrise the model with parameters \( \lambda \):

- Transition probabilities \( a_{kj} = P(q_{t+1} = j | q_t = k) \)
- Observation probabilities \( b_j(x) = P(x | q = j) \)
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, $q_t$. 
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
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

Not all transition probabilities are shown
Example topologies

left-to-right model

parallel path left-to-right model

ergodic model

\[
\begin{pmatrix}
a_{11} & a_{12} & 0 \\
0 & a_{22} & a_{23} \\
0 & 0 & a_{33}
\end{pmatrix}
\]

\[
\begin{pmatrix}
a_{11} & a_{12} & a_{13} & 0 & 0 \\
0 & a_{22} & a_{23} & a_{24} & 0 \\
0 & 0 & a_{33} & a_{34} & a_{35} \\
0 & 0 & 0 & a_{44} & a_{45} \\
0 & 0 & 0 & 0 & a_{55}
\end{pmatrix}
\]

\[
\begin{pmatrix}
a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\
a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\
a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\
a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\
a_{51} & a_{52} & a_{53} & a_{54} & a_{55}
\end{pmatrix}
\]

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.

![Diagram of HMMs for ASR](image)
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).
A note on HMM observation probabilities

<table>
<thead>
<tr>
<th>Continuous (density) HMM</th>
<th>Observation prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete (probability) HMM</td>
<td>continuous discrete</td>
</tr>
<tr>
<td>Semi-continuous HMM (tied-mixture HMM)</td>
<td>continuous tied mixture</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Observation prob.</strong></td>
<td></td>
</tr>
<tr>
<td>GMM, DNN</td>
<td></td>
</tr>
<tr>
<td>Vector quantisation</td>
<td></td>
</tr>
</tbody>
</table>

**Diagram:**

- $a_{01}$, $a_{12}$, $a_{23}$, $a_{34}$
- $a_{11}$, $a_{22}$, $a_{33}$
- $b_1(x)$, $b_2(x)$, $b_3(x)$

**Legend:**

- $a_{ij}$: Transition probability from state $i$ to state $j$.
- $b(x)$: Emission probability distribution.

**Emission pdfs:**

- $b_1(x)$: Red bell curve.
- $b_2(x)$: Red bell curve.
- $b_3(x)$: Red bell curve.
Computing likelihoods with the HMM

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 $Q$:

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

(1)

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

(2)

$P(q_1)$ denotes the initial occupancy probability of each state.
Consider a real valued random variable $X$

- Cumulative distribution function (cdf) $F(x)$ for $X$:
  \[
  F(x) = P(X \leq x)
  \]

- To obtain the probability of falling in an interval we can do the following:
  \[
  P(a < X \leq b) = P(X \leq b) - P(X \leq a) = F(b) - F(a)
  \]
The rate of change of the cdf gives us the *probability density function* (pdf), \( p(x) \):

\[
p(x) = \frac{d}{dx} F(x) = F'(x)
\]

\[
F(x) = \int_{-\infty}^{x} p(x)dx
\]

\( p(x) \) is *not* the probability that \( X \) has value \( x \). But the pdf is proportional to the probability that \( X \) lies in a small interval centred on \( x \).

Notation: \( p \) for pdf, \( P \) for probability
The Gaussian (or Normal) distribution is the most common (and easily analysed) continuous distribution. It is also a reasonable model in many situations (the famous “bell curve”). If a (scalar) variable has a Gaussian distribution, then it has a probability density function with this form:

\[ p(x | \mu, \sigma^2) = \mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( \frac{-(x - \mu)^2}{2\sigma^2} \right) \]

The Gaussian is described by two parameters:
- the mean \( \mu \) (location)
- the variance \( \sigma^2 \) (dispersion)
Gaussians have the same shape, with the location controlled by the mean, and the spread controlled by the variance.

One-dimensional Gaussian with zero mean and unit variance ($\mu = 0$, $\sigma^2 = 1$):
Properties of the Gaussian distribution

\[ \mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi \sigma^2}} \exp \left( -\frac{(x - \mu)^2}{2\sigma^2} \right) \]
Estimate mean and variance parameters of a Gaussian from data \(x_1, x_2, \ldots, x_T\)

Use the following as the estimates:

\[
\hat{\mu} = \frac{1}{T} \sum_{t=1}^{T} x_t \quad \text{(mean)}
\]

\[
\hat{\sigma}^2 = \frac{1}{T} \sum_{t=1}^{T} (x_t - \hat{\mu})^2 \quad \text{(variance)}
\]
Consider the log likelihood of a set of \( T \) training data points \( \{x_1, \ldots, x_T\} \) being generated by a Gaussian with mean \( \mu \) and variance \( \sigma^2 \):

\[
L = \ln p(\{x_1, \ldots, x_T\} | \mu, \sigma^2) = -\frac{1}{2} \sum_{t=1}^{T} \left( \frac{(x_t - \mu)^2}{\sigma^2} - \ln \sigma^2 - \ln(2\pi) \right)
\]

\[
= -\frac{1}{2\sigma^2} \sum_{t=1}^{T} (x_t - \mu)^2 - \frac{T}{2} \ln \sigma^2 - \frac{T}{2} \ln(2\pi)
\]

By maximising the log likelihood function with respect to \( \mu \) show that the maximum likelihood estimate for the mean is indeed the sample mean:

\[
\mu_{ML} = \frac{1}{T} \sum_{t=1}^{T} x_t.
\]
The multivariate Gaussian distribution

The $D$-dimensional vector $\mathbf{x} = (x_1, \ldots, x_D)^T$ follows a multivariate Gaussian (or normal) distribution if it has a probability density function of the following form:

$$p(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp \left( -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right)$$

The pdf is parameterised by the mean vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_D)^T$ and the covariance matrix $\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{11} & \cdots & \sigma_{1D} \\ \vdots & \ddots & \vdots \\ \sigma_{D1} & \cdots & \sigma_{DD} \end{pmatrix}$.
The multivariate Gaussian distribution

- The $D$-dimensional vector $\mathbf{x} = (x_1, \ldots, x_D)^T$ follows a multivariate Gaussian (or normal) distribution if it has a probability density function of the following form:

$$p(\mathbf{x} | \mu, \Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp \left( -\frac{1}{2} (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) \right)$$

The pdf is parameterised by the mean vector $\mu = (\mu_1, \ldots, \mu_D)^T$ and the covariance matrix $\Sigma = \begin{pmatrix} \sigma_{11} & \cdots & \sigma_{1D} \\ \vdots & \ddots & \vdots \\ \sigma_{D1} & \cdots & \sigma_{DD} \end{pmatrix}$.

- The 1-dimensional Gaussian is a special case of this pdf.
The multivariate Gaussian distribution

- The $D$-dimensional vector $\mathbf{x} = (x_1, \ldots, x_D)^T$ follows a multivariate Gaussian (or normal) distribution if it has a probability density function of the following form:

$$p(\mathbf{x} | \mu, \Sigma) = \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} \exp \left( -\frac{1}{2} (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) \right)$$

The pdf is parameterised by the mean vector $\mu = (\mu_1, \ldots, \mu_D)^T$ and the covariance matrix $\Sigma = \begin{pmatrix} \sigma_{11} & \ldots & \sigma_{1D} \\ \vdots & \ddots & \vdots \\ \sigma_{D1} & \ldots & \sigma_{DD} \end{pmatrix}$.

- The 1-dimensional Gaussian is a special case of this pdf
- The argument to the exponential $0.5(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)$ is referred to as a *quadratic form*. 
Covariance matrix

- The mean vector $\mu$ is the expectation of $x$:
  \[
  \mu = E[x]
  \]

- The covariance matrix $\Sigma$ is the expectation of the deviation of $x$ from the mean:
  \[
  \Sigma = E[(x - \mu)(x - \mu)^T]
  \]

- $\Sigma$ is a $D \times D$ symmetric matrix:
  \[
  \sigma_{ij} = E[(x_i - \mu_i)(x_j - \mu_j)] = E[(x_j - \mu_j)(x_i - \mu_i)] = \sigma_{ji}
  \]

- The sign of the covariance helps to determine the relationship between two components:
  - If $x_j$ is large when $x_i$ is large, then $(x_i - \mu_i)(x_j - \mu_j)$ will tend to be positive;
  - If $x_j$ is small when $x_i$ is large, then $(x_i - \mu_i)(x_j - \mu_j)$ will tend to be negative.
Spherical Gaussian

\[ \mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \rho_{12} = 0 \]

NB: Correlation coefficient  \[ \rho_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}} \quad (-1 \leq \rho_{ij} \leq 1) \]
Diagonal Covariance Gaussian

\[ \mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \]

\[ \Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix} \]

\[ \rho_{12} = 0 \]

NB: Correlation coefficient

\[ \rho_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii} \sigma_{jj}}} \quad (-1 \leq \rho_{ij} \leq 1) \]
Full covariance Gaussian

\[ \mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 1 & -1 \\ -1 & 4 \end{pmatrix} \quad \rho_{12} = -0.5 \]

NB: Correlation coefficient

\[ \rho_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii} \sigma_{jj}}} \quad (-1 \leq \rho_{ij} \leq 1) \]
Parameter estimation of a multivariate Gaussian distribution

It is possible to show that the mean vector $\hat{\mu}$ and covariance matrix $\hat{\Sigma}$ that maximise the likelihood of the training data are given by:

\[
\hat{\mu} = \frac{1}{T} \sum_{t=1}^{T} x_t
\]

\[
\hat{\Sigma} = \frac{1}{T} \sum_{t=1}^{T} (x_t - \hat{\mu})(x_t - \hat{\mu})^T
\]

where $x_t = (x_{t1}, \ldots, x_{tD})^T$.

NB: $T$ denotes either the number of samples or vector transpose depending on context.
Example data
Maximum likelihood fit to a Gaussian
Data in clusters (example 1)

\[ \mu_1 = (0, 0)^T \quad \mu_2 = (1, 1)^T \quad \Sigma_1 = \Sigma_2 = 0.2 I \]
Example 1 fit by a Gaussian

\[ \mu_1 = (0, 0)^T \quad \mu_2 = (1, 1)^T \quad \Sigma_1 = \Sigma_2 = 0.2 \mathbf{I} \]
k-means clustering

- k-means is an automatic procedure for clustering unlabelled data
- Requires a prespecified number of clusters
- Clustering algorithm chooses a set of clusters with the minimum within-cluster variance
- Guaranteed to converge (eventually)
- Clustering solution is dependent on the initialisation
k-means example: data set
k-means example: initialisation
k-means example: iteration 1 (assign points to clusters)
k-means example: iteration 1 (recompute centres)
k-means example: iteration 2 (assign points to clusters)
k-means example: iteration 2 (recompute centres)
k-means example: iteration 3 (assign points to clusters)

No changes, so converged
A more flexible form of density estimation is made up of a linear combination of component densities:

\[ p(x) = \sum_{m=1}^{M} P(m)p(x \mid m) \]

This is called a *mixture model* or a *mixture density*

- \( p(x \mid m) \): component densities
- \( P(m) \): mixing parameters

Generative model:

1. Choose a mixture component based on \( P(m) \)
2. Generate a data point \( x \) from the chosen component using \( p(x \mid m) \)
Gaussian mixture model

- The most important mixture model is the Gaussian Mixture Model (GMM), where the component densities are Gaussians.
- Consider a GMM, where each component Gaussian $\mathcal{N}(\mathbf{x}; \mu_m, \Sigma_m)$ has mean $\mu_m$ and a spherical covariance $\Sigma_m = \sigma_m^2 \mathbf{I}$.

\[
p(\mathbf{x}) = \sum_{m=1}^{M} P(m) P(\mathbf{x} | m) = \sum_{m=1}^{M} P(m) \mathcal{N}(\mathbf{x}; \mu_m, \sigma_m^2 \mathbf{I})
\]
GMM Parameter estimation when we know which component generated the data

- Define the indicator variable \( z_{mt} = 1 \) if component \( m \) generated data point \( x_t \) (and 0 otherwise)
- If \( z_{mt} \) wasn't hidden then we could count the number of observed data points generated by \( m \):
  \[
  N_m = \sum_{t=1}^{T} z_{mt}
  \]
- And estimate the mean, variance and mixing parameters as:
  \[
  \hat{\mu}_m = \frac{\sum_t z_{mt} x_t}{N_m}
  \]
  \[
  \hat{\sigma}_m^2 = \frac{\sum_t z_{mt} \| x_t - \hat{\mu}_m \|^2}{N_m}
  \]
  \[
  \hat{P}(m) = \frac{1}{T} \sum_t z_{mt} = \frac{N_m}{T}
  \]
GMM Parameter estimation when we don’t know which component generated the data

**Problem:** we don’t know $z_{mt}$ - which mixture component a data point comes from...
Problem: we don’t know $z_{mt}$ - which mixture component a data point comes from...

Idea: use the posterior probability $P(m|x)$, which gives the probability that component $m$ was responsible for generating data point $x$.

$$P(m|x) = \frac{p(x|m)P(m)}{p(x)} = \frac{p(x|m)P(m)}{\sum_{m'=1}^{M} p(x|m')P(m')}$$

The $P(m|x)$s are called the *component occupation probabilities* (or sometimes called the *responsibilities*)

Since they are posterior probabilities:

$$\sum_{m=1}^{M} P(m|x) = 1$$
Soft assignment

- Estimate "soft counts" based on the component occupation probabilities $P(m|x_t)$:

$$N_m^* = \sum_{t=1}^{T} P(m|x_t)$$

- We can imagine assigning data points to component $m$ weighted by the component occupation probability $P(m|x_t)$

- So we could imagine estimating the mean, variance and prior probabilities as:

$$\hat{\mu}_m = \frac{\sum_t P(m|x_t)x_t}{\sum_t P(m|x_t)} = \frac{\sum_t P(m|x_t)x_t}{N_m^*}$$

$$\hat{\sigma}_m^2 = \frac{\sum_t P(m|x_t)\|x_t - \hat{\mu}_m\|^2}{\sum_t P(m|x_t)} = \frac{\sum_t P(m|x_t)\|x_t - \hat{\mu}_m\|^2}{N_m^*}$$

$$\hat{P}(m) = \frac{1}{T} \sum_t P(m|x_t) = \frac{N_m^*}{T}$$
**EM algorithm**

- **Problem!** Recall that:

\[
P(m|x) = \frac{p(x|m)P(m)}{p(x)} = \frac{p(x|m)P(m)}{\sum_{m'=1}^{M} p(x|m')P(m')}
\]

We need to know \( p(x|m) \) and \( P(m) \) to estimate the parameters of \( P(m|x) \), and to estimate \( P(m) \).

- **Solution:** an iterative algorithm where each iteration has two parts:
  - Compute the component occupation probabilities \( P(m|x) \) using the current estimates of the GMM parameters (means, variances, mixing parameters) (E-step)
  - Computed the GMM parameters using the current estimates of the component occupation probabilities (M-step)

Starting from some initialisation (e.g. using k-means for the means) these steps are alternated until convergence

This is called the **EM Algorithm** and can be shown to maximise the likelihood. (NB: local maximum rather than global)
The likelihood of a data set \( \mathbf{X} = \{ \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T \} \) is given by:

\[
\mathcal{L} = \prod_{t=1}^{T} p(\mathbf{x}_t) = \prod_{t=1}^{T} \sum_{m=1}^{M} p(\mathbf{x}_t | m) P(m)
\]

We can regard the negative log likelihood as an error function:

Considering the derivatives of \( E \) with respect to the parameters, gives expressions like the previous slide
Example 1 fit using a GMM
Example 1 fit using a GMM

Fitted with a two component GMM using EM
Peakily distributed data (Example 2)

\[
\begin{align*}
\mu_1 &= \mu_2 = [0 \quad 0]^T \\
\Sigma_1 &= 0.1I \\
\Sigma_2 &= 2I
\end{align*}
\]
Example 2 fit by a Gaussian

\[ \mu_1 = \mu_2 = [0, 0]^T \quad \Sigma_1 = 0.1I \quad \Sigma_2 = 2I \]
Example 2 fit by a GMM
Example 2 fit by a GMM

Fitted with a two component GMM using EM
Example 2: component Gaussians

\[
P(x | m=1)
\]

\[
P(x | m=2)
\]
Comments on GMMs

- GMMs trained using the EM algorithm are able to self-organise to fit a data set.
- Individual components take responsibility for parts of the data set (probabilistically).
- Soft assignment to components not hard assignment — “soft clustering”.
- GMMs scale very well, e.g.: large speech recognition systems can have 30,000 GMMs, each with 32 components: sometimes 1 million Gaussian components!! And the parameters all estimated from (a lot of) data by EM.
Back to HMMs:

- Likelihood computation with the Forward algorithm
- Finding the most likely path with the Viterbi algorithm
- Parameter estimation with the Forward-Backward algorithm
References: HMMs


