University of Cambridge

MPhil in Computer Speech Text & Internet Technology

Module: Speech Processing 1

Lecture 8: Gaussian Mixture Models

Michaelmas 2003


**Introduction**

When we discussed Bayes’ decision rule there was the assumption that the class-conditional PDFs were correct. It is only when the correct class-conditional PDFs and prior probabilities are used that Bayes’ decision rule will yield the minimum error classifier. Motivated by the central limit theorem we examined the form of the decision boundaries generated by Gaussian distributions.

**What if the PDF is not Gaussian?**

In practice the distributions may not be Gaussian. The data may be:

- distributed asymmetrically about the mean;
- bimodally (or multi-modally) distributed.

In order to get as close as possible to the minimum error classifier we need would need to use a PDF that better models the data.
Mixture Models

One powerful (and very general) scheme for handling these problems is to use a *mixture model*.

A mixture model has two parts:

- **Component distribution**: the PDF (or PMF) associated with a particular *component* of the mixture model;

- **Component prior** (or weight): the “weighting” given to a particular component.

The general form is

\[ p(x|\theta) = \sum_{m=1}^{M} P(\omega_m)p(x|\omega_m, \theta_m) \]

where \( P(\omega_m) \) and \( p(x|\omega_m, \theta_m) \) are the \( m^{th} \) component prior and distribution respectively. From the definition of a valid PDF (or PMF) the following constraints on the priors must be satisfied:

\[
\sum_{m=1}^{M} P(\omega_m) = 1, \quad P(\omega_m) \geq 0 \quad m = 1, \ldots, M
\]
Simple Example

Noise is generated by one of two sources. 60% of the time it is generated by a Gaussian distribution of mean -1 and variance 1. 40% of the time it is generated by a Gaussian distribution of mean 1 and variance 1. What is the overall distribution of the noise observed?

A single Gaussian is used to model the data.

Two components fit the data “perfectly”.
Maximum Likelihood Estimates

We would like to find the maximum likelihood (ML) estimate of the model parameters given a set of independent training data

\[ X = \{x_1, \ldots, x_n\} \]

The likelihood of the training data is given by

\[ p(X|\theta) = \prod_{i=1}^{n} p(x_i|\theta) \]

As usual we take the log-likelihood, so in ML training we aim to maximise, with respect to \( \theta \),

\[ l(\theta) = \log(p(X|\theta)) = \sum_{i=1}^{n} \log(p(x_i|\theta)) = \sum_{i=1}^{n} \log\left(\sum_{m=1}^{M} P(\omega_m) p(x_i|\omega_m, \theta_m)\right) \]

subject to the constraints that

\[ \sum_{m=1}^{M} P(\omega_m) = 1, \quad P(\omega_m) \geq 0 \quad m = 1, \ldots, M \]

The set of model, \( \theta \) parameters consists of the parameters for each of the \( M \) components, \( \theta_m \), and the prior distributions for each of the \( M \) components, \( P(\omega_m), \ m = 1, \ldots, M \).

The most common form is when the component distribution is Gaussian.
Gaussian Mixture Models

Gaussian Mixture Models (GMMs), sometimes called mixture of Gaussians, may be written as

\[
p(x|\theta) = \sum_{m=1}^{M} P(\omega_m)p(x|\theta_m, \omega_m) = \sum_{m=1}^{M} P(\omega_m)\mathcal{N}(x; \mu_m, \Sigma_m)
\]

The model parameters are:

- \(M\) component priors \(P(\omega_1), \ldots, P(\omega_M)\);
- \(M\) mean vectors \(\mu_1, \ldots, \mu_M\);
- \(M\) covariance matrices \(\Sigma_1, \ldots, \Sigma_M\).

In the notation used on the previous slide

\[
\theta_m = \begin{bmatrix} \mu_m \\ \text{vec}(\Sigma_m) \end{bmatrix}
\]

We would like to find the ML estimate of the model parameters. Differentiating the expression for \(l(\theta)\) with respect to the mean of component \(m\) yields an expression that is not simple to equate to zero. (Those who are confident and very keen may want to try it!!)

We need a simple scheme to allow us to estimate the model parameters. But first let’s look at some uses.
Modelling PDFs

- Asymmetric and Bimodal distributions

![Graphs showing Asymmetric and Bimodal Distributions](a) Asymmetric Distribution  (b) Bimodal Distribution

- Correlation-modelling using diagonal covariance matrices

![Graphs showing Correlation-modelling](c) Contours  (d) Distribution
Number of Parameters

Why use multiple components to model the correlation?

- **Single Gaussian: diagonal covariance matrix**

\[ d \text{ mean} + d \text{ variance} = 2d \]

- **Single Gaussian: full covariance matrix**

\[ d \text{ mean} + d(d + 1)/2 \text{ cov} = d(d + 3)/2 \]

- **M diagonal Gaussian mixture**

\[ Md \text{ means} + Md \text{ variances} + M - 1 \text{ (mix weights)} \]

\[ = M(2d + 1) - 1 \]

As \( d \) increases it can be advantageous to use a Gaussian mixture of diagonal distributions instead of a full covariance matrix, and can give more flexible modelling.

Typical values used in speech recognition are \( d = 39 \) and \( M = 10 \). This gives:

- Single Gaussian diagonal: 78
- Single Gaussian full: 819
- 10 diagonal Gaussian mixture: 789
Simple “EM”

The above diagram shows two Gaussian components, each having an equal prior \(P(\omega_1) = P(\omega_2) = 0.5\). We have single dimensional training data \(x_1, \ldots, x_n\).

If the correct component for each observation was known we could use the standard estimation formulae for each of the components.

- **But** we don’t know the correct component.

- **But** we can estimate the assignment using the Gaussians above.

This is a classification problem, so use Bayes’ to assign it.
Simple “EM” (cont)

Let the assignment variable be $z$. For the above example we label all the data using

$$z_i = \begin{cases} 
\omega_1 & \text{if } x_i > 0 \\
\omega_2 & \text{if } x_i < 0 
\end{cases}$$

The estimate of the mean is then

$$\hat{\mu}_1 = \frac{1}{n_1} \sum_{i=\omega_1} x_i$$

$n_1$ is the number of samples assigned to $\omega_1$. The variance is

$$\hat{\sigma}_1^2 = \frac{1}{n_1} \sum_{i=\omega_1} (x_i - \hat{\mu}_1)^2$$

The prior can be estimated simply by using the relative frequencies of class $\omega_1$ and $\omega_2$

$$\hat{P}(\omega_1) = \frac{n_1}{n}$$

Similarly for class $\omega_2$.

We now have new estimates of the model parameters. We can therefore get new estimates of the assignment $z$. We can carry on doing this loop until nothing changes.

This is a simple iterative process that is guaranteed not to decrease the likelihood.
EM for GMMs

The simple “EM” used a hard assignment. In practice this simple partitioning does not occur between the two classes. We would rather use a soft, probabilistic assignment scheme. Now for each point we calculate the posterior probability of a component. The parameters at iteration $k$ are given by $\theta^{(k)}$. For component $j$

$$P(\omega_j|x_i, \theta^{(k)}) = \frac{P^{(k)}(\omega_j)p(x_i|\omega_j, \theta_j^{(k)})}{\sum_{m=1}^{M} P^{(k)}(\omega_m)p(x_i|\omega_m, \theta_m^{(k)})}$$

$P^{(k+1)}(\omega_m)$, $\mu^{(k+1)}_m$ and $\Sigma^{(k+1)}_m$ are the prior, mean and covariance matrix of component $\omega_m$ at iteration $k + 1$. The re-estimation formulae for the mean and covariance matrix of component $\omega_j$

$$P^{(k+1)}(\omega_j) = \frac{1}{n} \sum_{i=1}^{n} P(\omega_j|x_i, \theta^{(k)})$$

$$\mu^{(k+1)}_j = \frac{\sum_{i=1}^{n} P(\omega_j|x_i, \theta^{(k)})x_i}{\sum_{i=1}^{n} P(\omega_j|x_i, \theta^{(k)})}$$

$$\Sigma^{(k+1)}_j = \frac{\sum_{i=1}^{n} P(\omega_j|x_i, \theta^{(k)})(x_i - \mu^{(k+1)}_j)(x_i - \mu^{(k+1)}_j)^T}{\sum_{i=1}^{n} P(\omega_j|x_i, \theta^{(k)})}$$

This is **Expectation Maximisation**.
Example

The E-M algorithm was applied to the problem of estimating the parameters of a mixture model (mixture weights all set equal and not updated) as shown below. There are 5 Gaussian components in the mixture and the covariance matrices are diagonal.
Extracting Structure

Rather than using GMMs (or more generally mixture models) to model arbitrary PDFs, they may be used to extract structure from unlabelled data. Consider some simple speech data from which the formants (F1 and F2) have been extracted.

If this data was not labelled a two component GMM could be used to model the data. The hope is that the components will model different classes of the underlying data. Though it is not possible to label what the class component labels are it can be used to decide if data is similar to one class or another.

The problems are:

- how many components to use;
- the final solution depends on the initialisation!
Statistical Pattern Processing

The aim of this part of the module is to provide the mathematical and statistical background for speech pattern processing.

• 1: Introduction.

• 2–3: Basic Probability Theory.
  – Discrete Random Variables;
  – Continuous Random Variables.

• 4–6: Basic Pattern Processing.
  – Bayes’ Decision Rule;
  – Maximum Likelihood Training;
  – Linear Decision Boundaries.

• 7: Linear Projection Schemes.

• 8: Mixture Models.