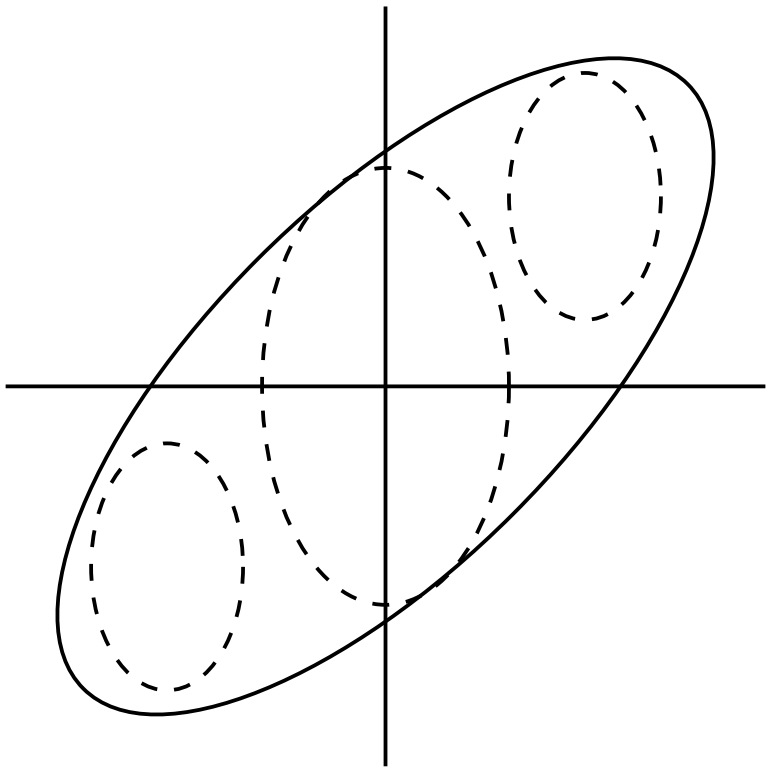
University of Cambridge
Engineering Part IIB

Module 4F10: Statistical Pattern Processing

Handout 3: Gaussian Mixture Models



Phil Woodland
pcw@eng.cam.ac.uk
Lent 2007

Introduction

It has so-far been assumed that the class-conditional pdfs can be adequately modelled by using a Gaussian distribution. We have also noted that we need to limit the number of parameters used to describe the distribution since these must be estimated from a limited training set.

The Gaussian form cannot adequately be used if the actual data is not close to Gaussian. For instance, multi-modal distributions can occur when the actual class could be decomposed into a number of identifiable sub-classes. This can happen for a number of causes. For instance in a speech recognition context because of

- grouping speech data from different speakers or accent groups
- analysing sounds independent from their immediate context which cause systematic variations
- modelling a larger group containing several sub-sounds such as the the set of front-vowels

These same type of effects often occur in other areas in which pattern recognition techniques are applied.

Furthermore if the distribution is non-symmetric or the data dimensions are correlated when we assume they are not (diagonal covariance assumption) then again the Gaussian assumption may be poor.

Mixtures of Gaussian distributions can help solve this prob-

lem in which the class conditional pdf is formed from a weighted sum of individual Gaussians. Given enough mixture components, it can be shown that Gaussian mixtures can model arbitrary distributions (but note the number of parameters that may be needed).

For a Gaussian mixture we have

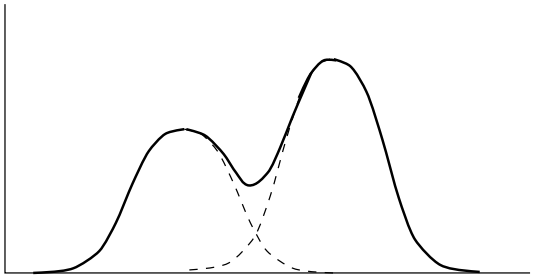
$$p(\boldsymbol{x}) = \sum_{m=1}^{M} c_m \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$$

where c_m is the component prior or *mixture weight* of each Gaussian component. For this to be a probability density function it is necessary that

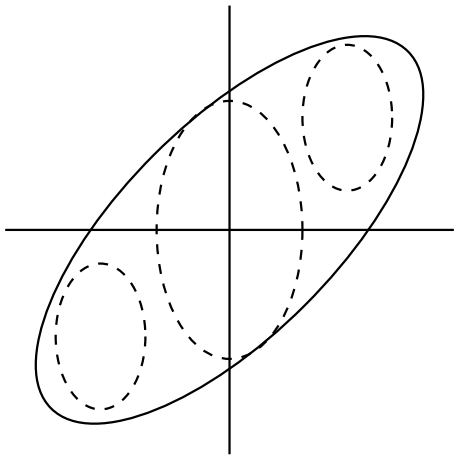
$$\sum_{m=1}^{M} c_m = 1 \quad \text{and} \quad c_m \geq 0$$

Some simple examples of modelling using a Gaussian mixture model include:

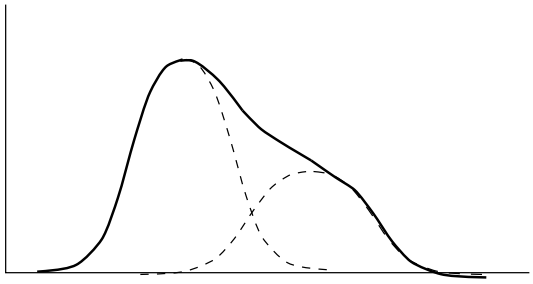
1. Modelling of multi-modal distributions.



2. Improved correlation-modelling when using diagonal covariance matrices.



3. Non-symmetric distributions.



Comparison of number of parameters:

Single Gaussian diagonal d mean + d variance $= 2d$

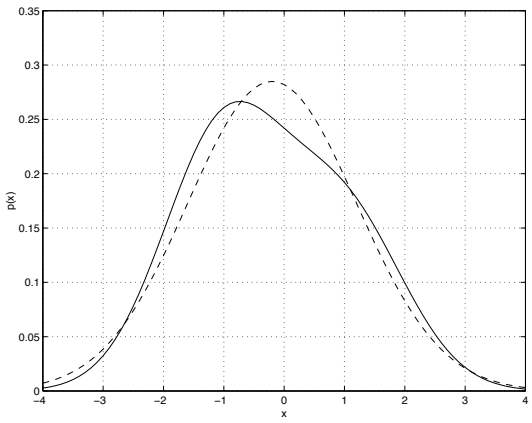
Single Gaussian full d mean + $d(d+1)/2$ cov $= d(d+3)/2$

M **diagonal Gaussian mixture** Md means + Md variances + $M-1$ (comp priors) $= M(2d+1)-1$

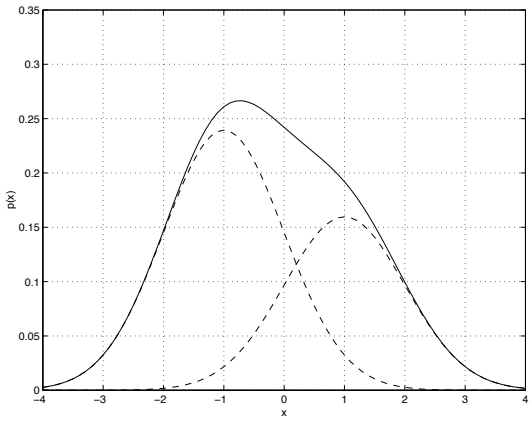
As d increases it can be advantageous to use a Gaussian mixture of diagonal distributions instead of a full covariance matrix, and can give more flexible modelling. (In general a mixture of full covariance Gaussians can also be used if enough training data is available).

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 generate by a Gaussian distribution of mean 1 and variance 1. What is the overall distribution of the noise observed?



If a single Gaussian is used as a model, there is a poor fit.



Two components fit the data "perfectly". What we are interested in is how to automatically train parameters of this mixture model from the observed data.

Likelihood Function for Mixture Models

We will estimate the parameters of a Gaussian mixture model using maximum likelihood (note mixtures of other distributions could also be considered). Note that if it was known with which mixture component each training data vector was associated then it would be a fairly straightforward task since we could estimate M separate Gaussians and the component priors could be determined from the relative frequency of each mixture component in the training data.

To do maximum likelihood estimation, first we need the log likelihood function for the data

$$l(\boldsymbol{\theta}) = \sum_{k=1}^{n} \ln p(\boldsymbol{x}_k) = \sum_{k=1}^{n} \ln \left[\sum_{m=1}^{M} p(\boldsymbol{x}_k|m) c_m \right]$$

where the dependence on the pdf for mixture component m is explicit.

For ease of presentation, we will consider Gaussian distributions of the form $\boldsymbol{\Sigma}_m = \sigma_m^2 \mathbf{I}$, although the principles can be easily extended to more general covariance matrices.

Therefore

$$l(\boldsymbol{\theta}) = \sum_{k=1}^{n} \ln \left[\sum_{m=1}^{M} c_m \frac{1}{(2\pi\sigma_m^2)^{d/2}} exp \left\{ \frac{-\left\| \boldsymbol{x}_k - \boldsymbol{\mu}_m \right\|^2}{2\sigma_m^2} \right\} \right]$$

Now, it is necessary to find the partial derivative of $l(\boldsymbol{\theta})$ with respect to the parameters of the mixture distribution.

Due to the form of the log likelihood we will (during the derivation below) use the substitution (from Bayes' noting that the c_m is a prior probability)

$$P(m|\boldsymbol{x}_k) = \frac{p(\boldsymbol{x}_k|m)c_m}{p(\boldsymbol{x}_k)}$$

where $P(m|\boldsymbol{x}_k)$ is the posterior probability of mixture component m being associated with vector $\boldsymbol{x}_k$ and the denominator here is given by the probability density of the vector from the entire mixture distribution *i.e.*

$$p(\boldsymbol{x}_k) = \sum_{m=1}^{M} c_m p(\boldsymbol{x}_k|m)$$

Considering a particular parameter θ_m that is associated with (only) the m^{th} mixture component. Since

$$l(\boldsymbol{\theta}) = \sum_{k=1}^{n} \ln \left[\sum_{m=1}^{M} p(\boldsymbol{x}_k|m)c_m \right]$$

then

$$\frac{\partial l(\boldsymbol{\theta})}{\partial \theta_m} = \sum_{k=1}^{n} \frac{1}{p(\boldsymbol{x}_k)} \frac{\partial \left[p(\boldsymbol{x}_k|m)c_m \right]}{\partial \theta_m}$$

Now using

$$\frac{\partial \left[\ln \left(p(\boldsymbol{x}_k|m)c_m \right) \right]}{\partial \theta_m} = \frac{1}{p(\boldsymbol{x}_k|m)c_m} \frac{\partial \left[p(\boldsymbol{x}_k|m)c_m \right]}{\partial \theta_m}$$

yields

$$\frac{\partial l(\boldsymbol{\theta})}{\partial \theta_m} = \sum_{k=1}^{n} P(m|\boldsymbol{x}_k) \frac{\partial \left[\ln \left(p(\boldsymbol{x}_k|m)c_m \right) \right]}{\partial \theta_m}$$

Now since

$$\frac{\partial \left[\ln \left(p(\boldsymbol{x}_k|m)c_m\right)\right]}{\partial \theta_m} = \frac{\partial}{\partial \theta_m}\left[\ln c_m - \frac{d}{2}\ln(2\pi) - \frac{d}{2}\ln(\sigma_m^2) - \frac{\left\|\boldsymbol{x}_k - \boldsymbol{\mu}_m\right\|^2}{2\sigma_m^2}\right]$$

Then, for the mean of the m^{th} mixture component

$$\frac{\partial l(\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_m} = \sum_{k=1}^{n} P(m|\boldsymbol{x}_k)\frac{(\boldsymbol{x}_k - \boldsymbol{\mu}_m)}{\sigma_m^2}$$

and

$$\frac{\partial l(\boldsymbol{\theta})}{\partial \sigma_m} = \sum_{k=1}^{n} P(m|\boldsymbol{x}_k)\left[\frac{\left\|\boldsymbol{x}_k - \boldsymbol{\mu}_m\right\|^2}{\sigma_m^3} - \frac{d}{\sigma_m}\right]$$

At the maximum of the likelihood function these derivatives must equal zero, and hence at that point

$$\hat{\boldsymbol{\mu}}_m = \frac{\sum_{k=1}^{n} P(m|\boldsymbol{x}_k)\boldsymbol{x}_k}{\sum_{k=1}^{n} P(m|\boldsymbol{x}_k)}$$

and

$$\hat{\sigma}_m^2 = \frac{1}{d}\frac{\sum_{k=1}^{n} P(m|\boldsymbol{x}_k)\left\|\boldsymbol{x}_k - \hat{\boldsymbol{\mu}}_m\right\|^2}{\sum_{k=1}^{n} P(m|\boldsymbol{x}_k)}$$

Note that these equations are coupled non-linear equations for the Gaussian parameters since the values of $P(m|\boldsymbol{x}_k)$ are functions of the Gaussian mixture parameters!

For a general covariance matrix, the covariance matrix estimate should satisfy:

$$\hat{\boldsymbol{\Sigma}}_m = \frac{\sum_{k=1}^{n} P(m|\boldsymbol{x}_k)(\boldsymbol{x}_k - \hat{\boldsymbol{\mu}}_m)(\boldsymbol{x}_k - \hat{\boldsymbol{\mu}}_m)'}{\sum_{k=1}^{n} P(m|\boldsymbol{x}_k)}$$

To estimate the component priors from the maximum of the log likelihood function, we note the constraint that the component priors must sum to one and be positive. This can be done using the method of Lagrange multipliers[1]. Add to the log likelihood a function that is equal to zero when the constraints are satisfied and maximise this new function.

In this case add $\lambda(\sum_{m=1}^{M} c_m - 1)$ so

$$\frac{\partial \left(l(\boldsymbol{\theta}) + \lambda(\sum_{m=1}^{M} c_m - 1) \right)}{\partial c_m} = \sum_{k=1}^{n} \frac{P(m|\boldsymbol{x}_k)}{c_m} - \lambda$$

This implies that at the required maximum (equating to zero)

$$\hat{c}_m = \frac{1}{\lambda} \sum_{k=1}^{n} P(m|\boldsymbol{x}_k)$$

The constraint that $\sum_{m=1}^{M} c_m = 1$ gives

$$\begin{aligned} \sum_{m=1}^{M} \frac{1}{\lambda} \sum_{k=1}^{n} P(m|\boldsymbol{x}_k) &= 1 \\ \lambda &= \sum_{k=1}^{n} \sum_{m=1}^{M} P(m|\boldsymbol{x}_k) \\ &= n \end{aligned}$$

so

$$\hat{c}_m = \frac{1}{n} \sum_{k=1}^{n} P(m|\boldsymbol{x}_k)$$

[1] see Bishop (1995) p.64 for an alternative method

Lagrange Optimisation

- Assume a extremum (maximum/minimum) of a scalar valued function $f(\boldsymbol{x})$ is required subject to a constraint.
- If the constraint can be expressed as $g(\boldsymbol{x}) = c$ then we can transform the constrained optimisation into an unconstrained one by finding the extremum of the Lagrangian function
$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) + \lambda \left[g(\boldsymbol{x}) - c \right]$$
- λ is called the Lagrange multiplier
- Find for extremum
$$\frac{\partial L(\boldsymbol{x}, \lambda)}{\partial \boldsymbol{x}} = \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} + \lambda \frac{\partial \left[g(\boldsymbol{x}) - c \right]}{\partial \boldsymbol{x}} = 0$$
Solve to give the required value of $\boldsymbol{x}$ and λ and hence extremum of $f(\boldsymbol{x})$ subject to the constraint $g(\boldsymbol{x}) = c$.
- Note also that
$$\frac{\partial L(\boldsymbol{x}, \lambda)}{\partial \lambda} = 0$$
when the constraint is satisfied.
- We will use this method in several places in this course.

Parameter Estimation for Gaussian Mixtures

The previous development gave relationships that must be satisfied at the maximum of the likelihood function but because the right-hand side depends on $P(m|\boldsymbol{x}_k)$, it doesn't give a closed form solution. However it implies that an iterative solution may be appropriate.

Given the differentials of the log likelihood function, a number of different optimisation schemes could be used (including gradient descent). The method here is based on the very general iterative Expectation-Maximisation algorithm.

Each iteration of the E-M algorithm for Gaussian Mixtures operates in two stages:

1. Find the posterior probability of mixture component occupation using the current parameter values.
2. Update the parameters of the Gaussian mixture as though the posterior probabilities were the true values.

Thus using the superscipt old for the parameters from the previous iteration and new for the updated parameters, the re-estimation equations for parameter estimation for a Gaussian mixture model can be expressed as:

$$\boldsymbol{\mu}_m^{\mathrm{new}} = \frac{\sum_k P^{\mathrm{old}}(m|\boldsymbol{x}_k)\boldsymbol{x}_k}{\sum_k P^{\mathrm{old}}(m|\boldsymbol{x}_k)}$$

$$(\sigma_m^{\text{new}})^2 = \frac{1}{d}\frac{\sum_k P^{\text{old}}(m|\boldsymbol{x}_k)\left\|\boldsymbol{x}_k - \boldsymbol{\mu}_m^{\text{new}}\right\|^2}{\sum_k P^{\text{old}}(m|\boldsymbol{x}_k)}$$

$$c_m^{\text{new}} = \frac{1}{n}\sum_k P^{\text{old}}(m|\boldsymbol{x}_k)$$

The application of these equations is guaranteed to provide an increase in the likelihood function unless the likelihood function is at a local maximum (proof of E-M next lecture ...).

Therefore the overall procedure is

1. Initialise the parameters of the mixture (e.g. set all component priors to be equal, all variances to be equal, and use different values for the mean vectors)
2. Compute $P^{\text{old}}(m|\boldsymbol{x}_k)$ for every data point and accumulate the statistics for the numerators and denominators of the re-estimation formulae. Also compute the log likelihood of the data set
3. Update the parameters as necessary.
4. If the log likelihood increase is less than a threshold stop, else goto 2.

Note that only a local maximum of the likelihood fuction is found by this procedure so the initialisation of the scheme is important, and can have problems (e.g. a variance can tend to zero and likelihood become infinite!)

Simple Worked Example

Consider some data from 2 classes:

Class 1 has points $\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.6 \\ 0.6 \end{bmatrix}, \begin{bmatrix} 0.7 \\ 0.4 \end{bmatrix}$

Class 2 has points $\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.25 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.3 \\ 0.4 \end{bmatrix}$

The aim is to build a mixture model on the composite data. The variances of both components are fixed at the identity matrix. Initial values of the means are

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 0.25 \\ 0.25 \end{bmatrix}, \quad \boldsymbol{\mu}_2 = \begin{bmatrix} 0.75 \\ 0.75 \end{bmatrix}$$

First the posteriors for the two model sets are required.
Class 1 has posteriors :

Comp1	0.5	0.3775	0.4750	0.4875
Comp2	0.5	0.6225	0.5250	0.5125

Class 2 has posteriors:

Comp1	0.6225	0.5	0.4688	0.5374
Comp2	0.3775	0.5	0.5312	0.4626

This gives updated means of:

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 0.4491 \\ 0.5143 \end{bmatrix}, \quad \boldsymbol{\mu}_2 = \begin{bmatrix} 0.5129 \\ 0.5851 \end{bmatrix}$$

This model has an increased likelihood of generating the data. Further iterations will increase the likelihood further.

Further 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 (though not constrained to be equal in each dimension). The figures shows the evaloution of training on intialisation, 1 iteration, 3 iterations and 16 iterations.