M5MS10
Machine Learning

Spring 2018

Lectures 7/8

Dr Ben Calderhead

b.calderhead@imperial.ac.uk

M5MS10
Machine Learning

Spring 2018

Lectures 7/8

Dr Ben Calderhead

b.calderhead@imperial.ac.uk

IS10
Learning

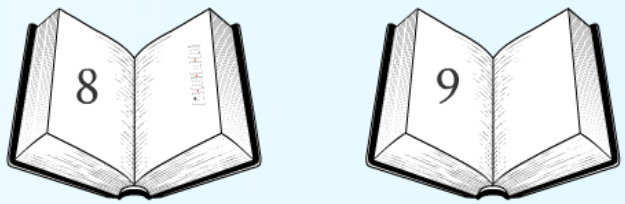
2018

es 7/8

lderhead

mperial.ac.uk

Clustering and Density Estimation



Density Estimation

Density Estimation

Density estimation is a fundamental problem in Machine Learning and Statistics.

In this lecture we shall consider *parametric approaches* to density estimation using *maximum likelihood*.

We shall then consider more complex mixture density models, and introduce the EM algorithm for learning the appropriate parameters for this approach.

Density Estimation

There are many cases where we observe some data and wish to infer the *underlying distribution* from which it was generated.

We have already seen an example of this in the generative approach to classification, where we need to estimate the *class conditional density* $p(\mathbf{x}|C=k)$.

We can give this density a *parametric form*, denoted by $p(\mathbf{x}|\theta_k)$, to form a likelihood function, which we can then maximise with respect to the parameters θ.

Density Estimation

We assume that there are N_k examples from class k and that its D features are distributed according to a multivariate Gaussian.

The likelihood function for class k therefore follows as

$$\begin{aligned}\prod_{n=1}^{N_k} p(\mathbf{x}_n|\theta_k) &= \prod_{n=1}^{N_k} p(\mathbf{x}_n|\mu_k,\Sigma_k) \\ &= \prod_{n=1}^{N_k} \frac{1}{\sqrt{(2\pi)^D|\Sigma_k|}} \exp\left\{-\frac{1}{2}(\mathbf{x}_n-\mu_k)^T\Sigma_k^{-1}(\mathbf{x}_n-\mu_k)\right\}\end{aligned}$$

We can work with the logarithm of this expression and ignore any constants,

$$\mathcal{L}_k = -\frac{N}{2}\log|\Sigma_k| - \frac{1}{2}\sum_{n=1}^{N_k}(\mathbf{x}_n-\mu_k)^T\Sigma_k^{-1}(\mathbf{x}_n-\mu_k)$$

MLE for Gaussian Distribution

We take derivatives of this *objective function* with respect to each of the parameters we want to learn. Remember we can expand and simplify the expression by ignoring all terms that are independent of the parameter of interest, since the derivative of these will be zero!

$$\begin{aligned}\frac{\partial}{\partial\mu_k}\mathcal{L}_k &= \frac{\partial}{\partial\mu_k}\left(\frac{1}{2}\sum_{n=1}^{N_k}\{2\mu_k^T\Sigma_k^{-1}\mathbf{x}_n - \mu_k^T\Sigma_k^{-1}\mu_k\}\right) \\ &= \sum_{n=1}^{N_k}\{\Sigma_k^{-1}\mathbf{x}_n - \Sigma_k^{-1}\mu_k\}\end{aligned}$$

MLE for Gaussian Distribution

Setting the gradient equal to zero we obtain,

$$\begin{aligned}\sum_{n=1}^{N_k}\Sigma_k^{-1}\mathbf{x}_n &= \sum_{n=1}^{N_k}\Sigma_k^{-1}\mu_k \\ &= N\Sigma_k^{-1}\mu_k\end{aligned}$$

Multiplying both sides by the matrix Σ_k we obtain the maximum likelihood estimate for the mean of the class conditional multivariate Gaussian distribution,

$$\hat{\mu}_k = \frac{1}{N_k}\sum_{n=1}^{N_k}\mathbf{x}_n$$

MLE for Gaussian Distribution

We can do the same for the covariance matrix parameter, Σ_k.

We can make use of the following identity for the derivative of the determinant of a matrix,

$$\frac{\partial}{\partial\Sigma_k}|\Sigma_k| = |\Sigma_k|\Sigma_k^{-1}$$

We therefore have,

$$\frac{\partial}{\partial\Sigma_k}\left(\frac{N_k}{2}\log|\Sigma_k|\right) = \frac{N_k}{2|\Sigma_k|}|\Sigma_k|\Sigma_k^{-1} = \frac{N_k}{2}\Sigma_k^{-1}$$

Note that these useful matrix identities can be found in the Matrix Cookbook.

MLE for Gaussian Distribution

Finally we can use the matrix identity

$$\frac{\partial}{\partial\mathbf{X}}\mathbf{a}^T\mathbf{X}^{-1}\mathbf{b} = -\mathbf{X}^{-1}\mathbf{a}\mathbf{b}^T\mathbf{X}^{-1}$$

to ascertain that,

$$\frac{\partial}{\partial\Sigma_k}\sum_{n=1}^{N_k}(\mathbf{x}_n-\mu_k)^T\Sigma_k^{-1}(\mathbf{x}_n-\mu_k) = -\sum_{n=1}^{N_k}\Sigma_k^{-1}(\mathbf{x}_n-\mu_k)(\mathbf{x}_n-\mu_k)^T\Sigma_k^{-1}$$

The derivative of the log likelihood with respect to the covariance parameter therefore follows as

$$\frac{\partial}{\partial\Sigma_k}\mathcal{L}_k = -\frac{N_k}{2}\Sigma_k^{-1} + \frac{1}{2}\sum_{n=1}^{N_k}\Sigma_k^{-1}(\mathbf{x}_n-\mu_k)(\mathbf{x}_n-\mu_k)^T\Sigma_k^{-1}$$

MLE for Gaussian Distribution

If we set the gradient equal to zero and employ the mean vectors with their maximum likelihood estimates, then as we might expect we obtain the following estimate for the class conditional likelihood,

$$\hat{\Sigma}_k = \frac{1}{N_k}\sum_{n=1}^{N_k}(\mathbf{x}_n-\hat{\mu}_k)(\mathbf{x}_n-\hat{\mu}_k)^T$$

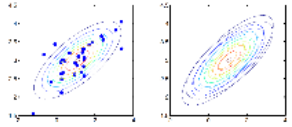
We note that we can employ this *maximum likelihood* procedure for any chosen parametric density, and indeed we could also define prior distributions and employ the *Bayesian approach*.

MLE for Gaussian Distribution

Let's consider 30 data points drawn from a bivariate Gaussian distribution with the following parameters,

$$\mu = \begin{pmatrix}1\\3\end{pmatrix} \qquad \Sigma = \begin{pmatrix}1.5 & 0.6\\0.6 & 0.4\end{pmatrix}$$

Clearly, as we increase the number of data points we employ, $N\to\infty$, the estimate will converge to the true value.



Density Estimation

Density estimation is a fundamental problem in Machine Learning and Statistics.

In this lecture we shall consider *parametric approaches* to density estimation using *maximum likelihood*.

We shall then consider more complex mixture density models, and introduce the EM algorithm for learning the appropriate parameters for this approach.

Density Estimation

There are many cases where we observe some data and wish to infer the *underlying distribution* from which it was generated.

We have already seen an example of this in the generative approach to classification, where we need to estimate the *class conditional density* $p(\mathbf{x}|C = k)$.

We can give this density a *parametric form*, denoted by $p(\mathbf{x}|\boldsymbol{\theta}_k)$, to form a likelihood function, which we can then maximise with respect to the parameters $\boldsymbol{\theta}$.

Density Estimation

We assume that there are N_k examples from class k and that its D features are distributed according to a multivariate Gaussian.

The likelihood function for class k therefore follows as

$$\begin{aligned}\prod_{n=1}^{N_k} p(\mathbf{x}_n|\boldsymbol{\theta}_k) &= \prod_{n=1}^{N_k} p(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ &= \prod_{n=1}^{N_k} \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}_k|}} \exp\left\{-\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k)\right\}\end{aligned}$$

We can work with the logarithm of this expression and ignore any constants,

$$\mathcal{L}_k = -\frac{N}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}\sum_{n=1}^{N_k}(\mathbf{x}_n - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k)$$

MLE for Gaussian Distribution

We take derivatives of this *objective function* with respect to each of the parameters we want to learn. Remember we can expand and simplify the expression by ignoring all terms that are independent of the parameter of interest, since the derivative of these will be zero!

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\mu}_k}\mathcal{L}_k &= \frac{\partial}{\partial \boldsymbol{\mu}_k}\left(\frac{1}{2}\sum_{n=1}^{N_k}\{2\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_k^{-1}\mathbf{x}_n - \boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k\}\right) \\ &= \sum_{n=1}^{N_k}\{\boldsymbol{\Sigma}_k^{-1}\mathbf{x}_n - \boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k\}\end{aligned}$$

MLE for Gaussian Distribution

Setting the gradient equal to zero we obtain,

$$\begin{aligned}\sum_{n=1}^{N_k} \boldsymbol{\Sigma}_k^{-1}\mathbf{x}_n &= \sum_{n=1}^{N_k} \boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k \\ &= N\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k\end{aligned}$$

Multiplying both sides by the matrix $\boldsymbol{\Sigma}_k$ we obtain the maximum likelihood estimate for the mean of the class conditional multivariate Gaussian distribution,

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{N_k}\sum_{n=1}^{N_k}\mathbf{x}_n$$

MLE for Gaussian Distribution

We can do the same for the covariance matrix parameter, $\boldsymbol{\Sigma}_k$.

We can make use of the following identity for the derivative of the determinant of a matrix,

$$\frac{\partial}{\partial \boldsymbol{\Sigma}_k}|\boldsymbol{\Sigma}_k| = |\boldsymbol{\Sigma}_k|\boldsymbol{\Sigma}_k^{-1}$$

We therefore have,

$$\frac{\partial}{\partial \boldsymbol{\Sigma}_k}\left(\frac{N_k}{2}\log|\boldsymbol{\Sigma}_k|\right) = \frac{N_k}{2|\boldsymbol{\Sigma}_k|}|\boldsymbol{\Sigma}_k|\boldsymbol{\Sigma}_k^{-1} = \frac{N_k}{2}\boldsymbol{\Sigma}_k^{-1}$$

Note that these useful matrix identities can be found in the Matrix Cookbook.

MLE for Gaussian Distribution

Finally we can use the matrix identity

$$\frac{\partial}{\partial \mathbf{X}} \mathbf{a}^T \mathbf{X}^{-1} \mathbf{b} = -\mathbf{X}^{-1} \mathbf{a} \mathbf{b}^T \mathbf{X}^{-1}$$

to ascertain that,

$$\frac{\partial}{\partial \boldsymbol{\Sigma}_k} \sum_{n=1}^{N_k} (\mathbf{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) = -\sum_{n=1}^{N_k} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}$$

The derivative of the log likelihood with respect to the covariance parameter therefore follows as

$$\frac{\partial}{\partial \boldsymbol{\Sigma}_k} \mathcal{L}_k = -\frac{N_k}{2} \boldsymbol{\Sigma}_k^{-1} + \frac{1}{2} \sum_{n=1}^{N_k} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}$$

MLE for Gaussian Distribution

If we set the gradient equal to zero and employ the mean vectors with their maximum likelihood estimates, then as we might expect we obtain the following estimate for the class conditional likelihood,

$$\hat{\boldsymbol{\Sigma}}_k = \frac{1}{N_k} \sum_{n=1}^{N_k} (\mathbf{x}_n - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_k)^T$$

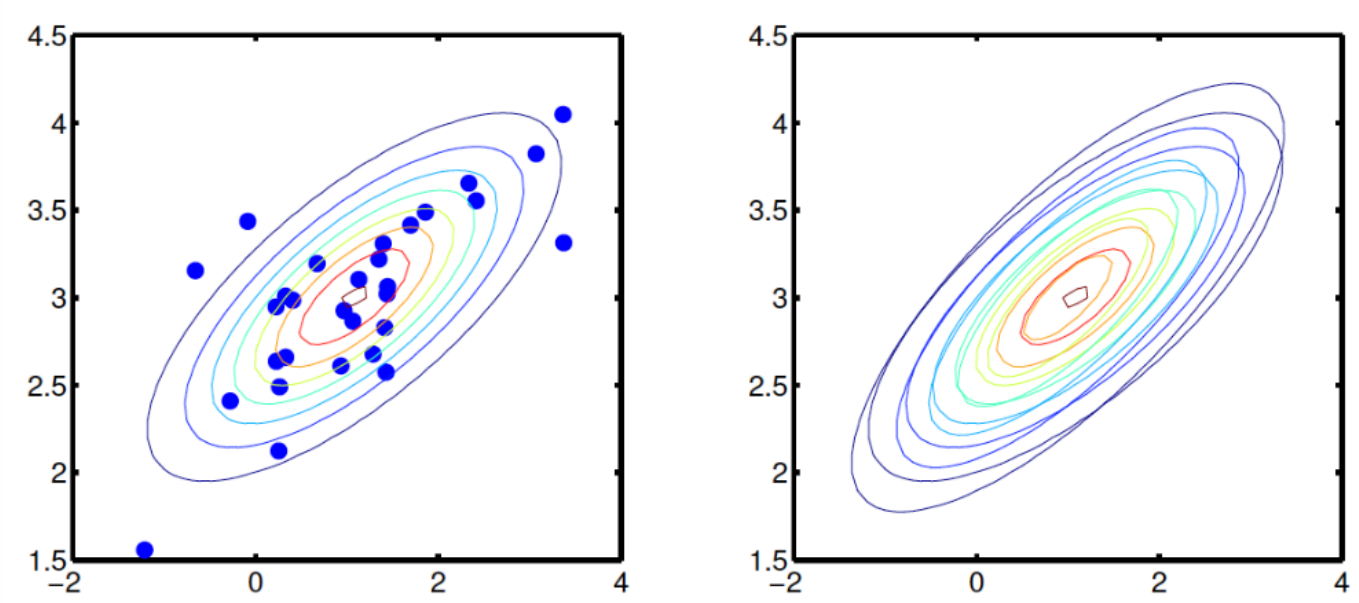
We note that we can employ this *maximum likelihood* procedure for any chosen parametric density, and indeed we could also define prior distributions and employ the *Bayesian approach*.

MLE for Gaussian Distribution

Let's consider 30 data points drawn from a bivariate Gaussian distribution with the following parameters,

$$\boldsymbol{\mu} = \begin{pmatrix} 1 \\ 3 \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 1.5 & 0.6 \\ 0.6 & 0.4 \end{pmatrix}$$

Clearly, as we increase the number of data points we employ, $N \to \infty$, the estimate will converge to the true value.



Density Estimation

Density Estimation

Density estimation is a fundamental problem in Machine Learning and Statistics.

In this lecture we shall consider *parametric approaches* to density estimation using *maximum likelihood*.

We shall then consider more complex mixture density models, and introduce the EM algorithm for learning the appropriate parameters for this approach.

Density Estimation

There are many cases where we observe some data and wish to infer the *underlying distribution* from which it was generated.

We have already seen an example of this in the generative approach to classification, where we need to estimate the *class conditional density* $p(\mathbf{x}|C=k)$.

We can give this density a *parametric form*, denoted by $p(\mathbf{x}|\theta_k)$, to form a likelihood function, which we can then maximise with respect to the parameters θ.

Density Estimation

We assume that there are N_k examples from class k and that its D features are distributed according to a multivariate Gaussian.

The likelihood function for class k therefore follows as

$$\begin{aligned}\prod_{n=1}^{N_k} p(\mathbf{x}_n|\theta_k) &= \prod_{n=1}^{N_k} p(\mathbf{x}_n|\mu_k,\Sigma_k) \\ &= \prod_{n=1}^{N_k} \frac{1}{\sqrt{(2\pi)^D|\Sigma_k|}} \exp\left\{-\frac{1}{2}(\mathbf{x}_n-\mu_k)^T\Sigma_k^{-1}(\mathbf{x}_n-\mu_k)\right\}\end{aligned}$$

We can work with the logarithm of this expression and ignore any constants,

$$\mathcal{L}_k = -\frac{N}{2}\log|\Sigma_k| - \frac{1}{2}\sum_{n=1}^{N_k}(\mathbf{x}_n-\mu_k)^T\Sigma_k^{-1}(\mathbf{x}_n-\mu_k)$$

MLE for Gaussian Distribution

We take derivatives of this *objective function* with respect to each of the parameters we want to learn. Remember we can expand and simplify the expression by ignoring all terms that are independent of the parameter of interest, since the derivative of these will be zero!

$$\begin{aligned}\frac{\partial}{\partial\mu_k}\mathcal{L}_k &= \frac{\partial}{\partial\mu_k}\left(\frac{1}{2}\sum_{n=1}^{N_k}\{2\mu_k^T\Sigma_k^{-1}\mathbf{x}_n - \mu_k^T\Sigma_k^{-1}\mu_k\}\right) \\ &= \sum_{n=1}^{N_k}\{\Sigma_k^{-1}\mathbf{x}_n - \Sigma_k^{-1}\mu_k\}\end{aligned}$$

MLE for Gaussian Distribution

Setting the gradient equal to zero we obtain,

$$\begin{aligned}\sum_{n=1}^{N_k}\Sigma_k^{-1}\mathbf{x}_n &= \sum_{n=1}^{N_k}\Sigma_k^{-1}\mu_k \\ &= N\Sigma_k^{-1}\mu_k\end{aligned}$$

Multiplying both sides by the matrix Σ_k we obtain the maximum likelihood estimate for the mean of the class conditional multivariate Gaussian distribution,

$$\hat{\mu}_k = \frac{1}{N_k}\sum_{n=1}^{N_k}\mathbf{x}_n$$

MLE for Gaussian Distribution

We can do the same for the covariance matrix parameter, Σ_k.

We can make use of the following identity for the derivative of the determinant of a matrix,

$$\frac{\partial}{\partial\Sigma_k}|\Sigma_k| = |\Sigma_k|\Sigma_k^{-1}$$

We therefore have,

$$\frac{\partial}{\partial\Sigma_k}\left(\frac{N_k}{2}\log|\Sigma_k|\right) = \frac{N_k}{2|\Sigma_k|}|\Sigma_k|\Sigma_k^{-1} = \frac{N_k}{2}\Sigma_k^{-1}$$

Note that these useful matrix identities can be found in the Matrix Cookbook.

MLE for Gaussian Distribution

Finally we can use the matrix identity

$$\frac{\partial}{\partial\mathbf{X}}\mathbf{a}^T\mathbf{X}^{-1}\mathbf{b} = -\mathbf{X}^{-1}\mathbf{a}\mathbf{b}^T\mathbf{X}^{-1}$$

to ascertain that,

$$\frac{\partial}{\partial\Sigma_k}\sum_{n=1}^{N_k}(\mathbf{x}_n-\mu_k)^T\Sigma_k^{-1}(\mathbf{x}_n-\mu_k) = -\sum_{n=1}^{N_k}\Sigma_k^{-1}(\mathbf{x}_n-\mu_k)(\mathbf{x}_n-\mu_k)^T\Sigma_k^{-1}$$

The derivative of the log likelihood with respect to the covariance parameter therefore follows as

$$\frac{\partial}{\partial\Sigma_k}\mathcal{L}_k = -\frac{N_k}{2}\Sigma_k^{-1} + \frac{1}{2}\sum_{n=1}^{N_k}\Sigma_k^{-1}(\mathbf{x}_n-\mu_k)(\mathbf{x}_n-\mu_k)^T\Sigma_k^{-1}$$

MLE for Gaussian Distribution

If we set the gradient equal to zero and employ the mean vectors with their maximum likelihood estimates, then as we might expect we obtain the following estimate for the class conditional likelihood,

$$\hat{\Sigma}_k = \frac{1}{N_k}\sum_{n=1}^{N_k}(\mathbf{x}_n-\hat{\mu}_k)(\mathbf{x}_n-\hat{\mu}_k)^T$$

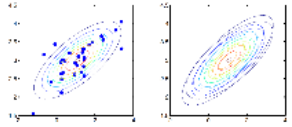
We note that we can employ this *maximum likelihood* procedure for any chosen parametric density, and indeed we could also define prior distributions and employ the *Bayesian approach*.

MLE for Gaussian Distribution

Let's consider 30 data points drawn from a bivariate Gaussian distribution with the following parameters,

$$\mu = \begin{pmatrix}1\\3\end{pmatrix} \qquad \Sigma = \begin{pmatrix}1.5 & 0.6\\0.6 & 0.4\end{pmatrix}$$

Clearly, as we increase the number of data points we employ, $N\to\infty$, the estimate will converge to the true value.



Mixture Models

Non-Gaussian Example

Now let's consider an example where we have data and wrongly assume that it is drawn from a Gaussian distribution.

In particular, let us draw some data from a mixture of two Gaussian distributions, defined by

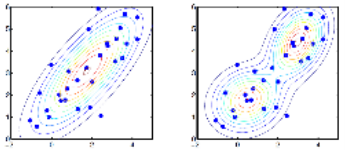
$$\mu_1 = \begin{pmatrix} 0.5 \\ 2 \end{pmatrix} \qquad \Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\mu_2 = \begin{pmatrix} 3 \\ 4 \end{pmatrix} \qquad \Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

There are therefore two underlying generating process for the data we observe.

Non-Gaussian Example

Here we can see the data with the contours of the maximum likelihood estimated *single* Gaussian model, and the true mixture of two Gaussian distributions from which the data was actually sampled.



Mixture Models

Let's consider a more flexible model that comprises a mixture of Gaussian distributions.

In the case of two Gaussians, this can be represented as

$$\begin{aligned} p(\mathbf{x}|\boldsymbol{\theta}) &= \pi p(\mathbf{x}|\boldsymbol{\theta}_1) + (1-\pi)p(\mathbf{x}|\boldsymbol{\theta}_2) \\ &= \pi \mathcal{N}_{\mathbf{x}}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1-\pi)\mathcal{N}_{\mathbf{x}}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \end{aligned}$$

where our set of parameters is now,

$$\boldsymbol{\theta} = [\pi, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2] = [\pi, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2]$$

The parameter π denotes the probability that the data point $\mathbf{x}$ was generated from $p(\mathbf{x}|\boldsymbol{\theta}_1)$, and so there is probability $1-\pi$ that the point was generated from $p(\mathbf{x}|\boldsymbol{\theta}_2)$.

Mixture Models

The most general case is where there are M Gaussian distributions in the mixture model, in which case the probability density can be expressed as,

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{m=1}^{M} \pi_m p(\mathbf{x}|\boldsymbol{\theta}_m)$$

where the parameter set is now defined as,

$$\boldsymbol{\theta} = [\pi_1, \ldots, \pi_M, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M]$$

and $\sum_{m=1}^{M} \pi_m = 1$, since π_m describes the probability of the data point being generated by the mth component.

Mixture Models

We therefore need to estimate the full set of parameters for this mixture model.

Given some data points $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ we assume the mixture model $p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{m=1}^{M} \pi_m p(\mathbf{x}|\boldsymbol{\theta}_m)$.

We need estimates of each π_m and so if we have labels we could just count how many points in $\mathcal{D}$ come from each of the m components, then normalise by the total number N.

i.e. if N_m points in $\mathcal{D}$ come from component m then we use the estimate,

$$\hat{\pi}_m = \frac{N_m}{N}$$

Similarly, for the mean and covariance parameters we can simply use the maximum likelihood expressions we have derived already, for each of the m components.

EM Algorithm

However... often we will not have these labels in advance.

For example, we might wish to model each class conditional likelihood as a mixture model, and so we won't know which of these data points were generated by each of the component in our mixture model.

For convenience, let us introduce indicator variables z_{mn}, where $z_{mn} = 1$ indicates that data point $\mathbf{x}_n$ was generated from the mth component of our mixture model.

If the variables z_{mn} are not known, then $\mathbf{z}$ are known as latent or hidden variables, and we cannot simply apply the maximum likelihood estimators straightforwardly.

Non-Gaussian Example

Now let's consider an example where we have data and wrongly assume that it is drawn from a Gaussian distribution.

In particular, let us draw some data from a mixture of two Gaussian distributions, defined by

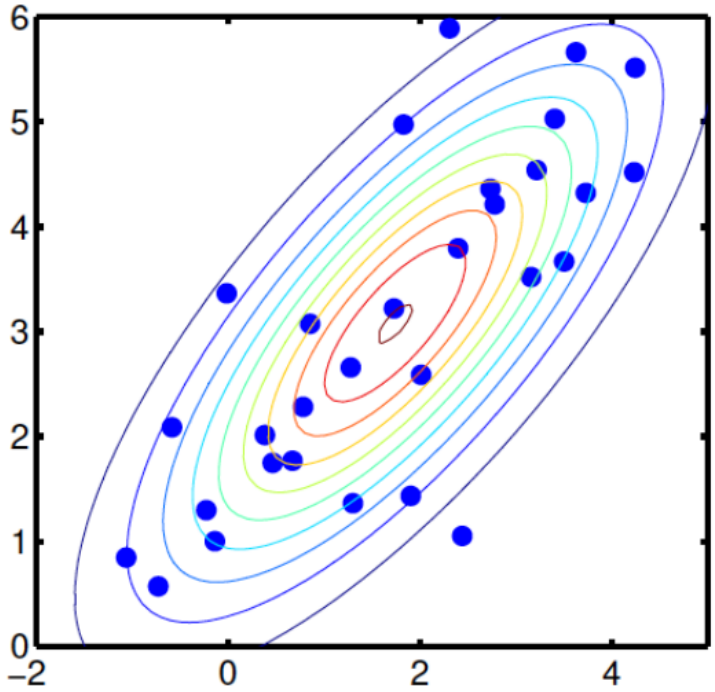
$$\boldsymbol{\mu}_1 = \begin{pmatrix} 0.5 \\ 2 \end{pmatrix} \qquad \boldsymbol{\Sigma}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

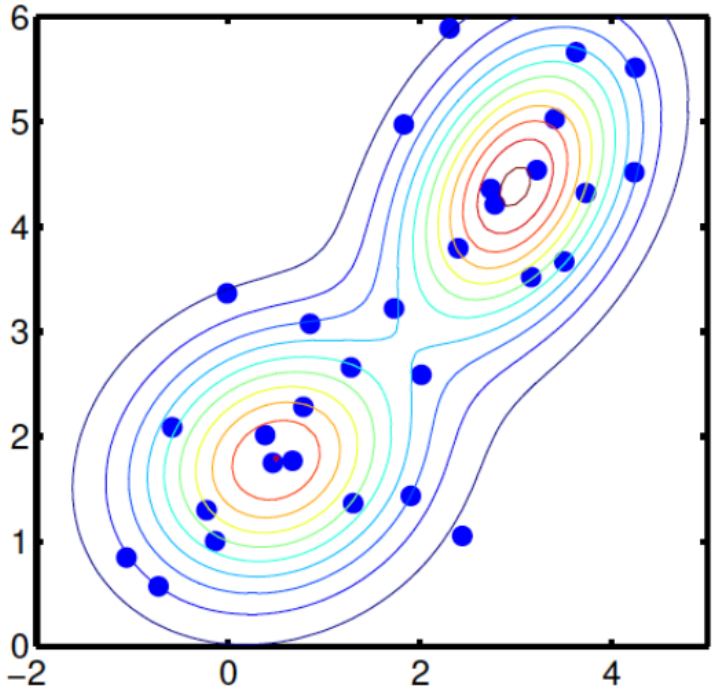
$$\boldsymbol{\mu}_2 = \begin{pmatrix} 3 \\ 4 \end{pmatrix} \qquad \boldsymbol{\Sigma}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

There are therefore two underlying generating process for the data we observe.

Non-Gaussian Example

Here we can see the data with the contours of the maximum likelihood estimated *single* Gaussian model, and the true mixture of two Gaussian distributions from which the data was actually sampled.





Mixture Models

Let's consider a more flexible model that comprises a mixture of Gaussian distributions.

In the case of two Gaussians, this can be represented as

$$\begin{aligned} p(\mathbf{x}|\boldsymbol{\theta}) &= \pi p(\mathbf{x}|\boldsymbol{\theta}_1) + (1-\pi)p(\mathbf{x}|\boldsymbol{\theta}_2) \\ &= \pi \mathcal{N}_{\mathbf{x}}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1-\pi)\mathcal{N}_{\mathbf{x}}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \end{aligned}$$

where our set of parameters is now,

$$\boldsymbol{\theta} = [\pi, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2] = [\pi, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2]$$

The parameter π denotes the probability that the data point $\mathbf{x}$ was generated from $p(\mathbf{x}|\boldsymbol{\theta}_1)$, and so there is probability $1-\pi$ that the point was generated from $p(\mathbf{x}|\boldsymbol{\theta}_2)$.

Mixture Models

The most general case is where there are M Gaussian distributions in the mixture model, in which case the probability density can be expressed as,

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{m=1}^{M} \pi_m p(\mathbf{x}|\boldsymbol{\theta}_m)$$

where the parameter set is now defined as,

$$\boldsymbol{\theta} = [\pi_1, \ldots, \pi_M, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M]$$

and $\sum_{m=1}^{M} \pi_m = 1$, since π_m describes the probability of the data point being generated by the mth component.

Mixture Models

We therefore need to estimate the full set of parameters for this mixture model.

Given some data points $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ we assume the mixture model $p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{m=1}^{M} \pi_m p(\mathbf{x}|\boldsymbol{\theta}_m)$.

We need estimates of each π_m and so if we have labels we could just count how many points in $\mathcal{D}$ come from each of the m components, then normalise by the total number N.

i.e. if N_m points in $\mathcal{D}$ come from component m then we use the estimate,

$$\hat{\pi}_m = \frac{N_m}{N}$$

Similarly, for the mean and covariance parameters we can simply use the maximum likelihood expressions we have derived already, for each of the m components.

EM Algorithm

However... often we will not have these labels in advance.

For example, we might wish to model each class conditional likelihood as a mixture model, and so we won't know which of these data points were generated by each of the component in our mixture model.

For convenience, let us introduce indicator variables z_{mn}, where $z_{mn} = 1$ indicates that data point $\mathbf{x}_n$ was generated from the mth component of our mixture model.

If the variables z_{mn} are not known, then $\mathbf{z}$ are known as latent or hidden variables, and we cannot simply apply the maximum likelihood estimators straightforwardly.

Mixture Models

Non-Gaussian Example

Now let's consider an example where we have data and wrongly assume that it is drawn from a Gaussian distribution.

In particular, let us draw some data from a mixture of two Gaussian distributions, defined by

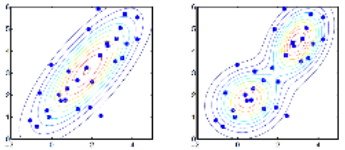
$$\mu_1 = \begin{pmatrix} 0.5 \\ 2 \end{pmatrix} \qquad \Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\mu_2 = \begin{pmatrix} 3 \\ 4 \end{pmatrix} \qquad \Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

There are therefore two underlying generating process for the data we observe.

Non-Gaussian Example

Here we can see the data with the contours of the maximum likelihood estimated *single* Gaussian model, and the true mixture of two Gaussian distributions from which the data was actually sampled.



Mixture Models

Let's consider a more flexible model that comprises a mixture of Gaussian distributions.

In the case of two Gaussians, this can be represented as

$$\begin{aligned} p(\mathbf{x}|\boldsymbol{\theta}) &= \pi p(\mathbf{x}|\boldsymbol{\theta}_1) + (1-\pi)p(\mathbf{x}|\boldsymbol{\theta}_2) \\ &= \pi \mathcal{N}_{\mathbf{x}}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1-\pi)\mathcal{N}_{\mathbf{x}}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \end{aligned}$$

where our set of parameters is now,

$$\boldsymbol{\theta} = [\pi, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2] = [\pi, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2]$$

The parameter π denotes the probability that the data point $\mathbf{x}$ was generated from $p(\mathbf{x}|\boldsymbol{\theta}_1)$, and so there is probability $1-\pi$ that the point was generated from $p(\mathbf{x}|\boldsymbol{\theta}_2)$.

Mixture Models

The most general case is where there are M Gaussian distributions in the mixture model, in which case the probability density can be expressed as,

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{m=1}^{M} \pi_m p(\mathbf{x}|\boldsymbol{\theta}_m)$$

where the parameter set is now defined as,

$$\boldsymbol{\theta} = [\pi_1, \ldots, \pi_M, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M]$$

and $\sum_{m=1}^{M} \pi_m = 1$, since π_m describes the probability of the data point being generated by the mth component.

Mixture Models

We therefore need to estimate the full set of parameters for this mixture model.

Given some data points $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ we assume the mixture model $p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{m=1}^{M} \pi_m p(\mathbf{x}|\boldsymbol{\theta}_m)$.

We need estimates of each π_m and so if we have labels we could just count how many points in $\mathcal{D}$ come from each of the m components, then normalise by the total number N.

i.e. if N_m points in $\mathcal{D}$ come from component m then we use the estimate,

$$\hat{\pi}_m = \frac{N_m}{N}$$

Similarly, for the mean and covariance parameters we can simply use the maximum likelihood expressions we have derived already, for each of the m components.

EM Algorithm

However... often we will not have these labels in advance.

For example, we might wish to model each class conditional likelihood as a mixture model, and so we won't know which of these data points were generated by each of the component in our mixture model.

For convenience, let us introduce indicator variables z_{mn}, where $z_{mn} = 1$ indicates that data point $\mathbf{x}_n$ was generated from the mth component of our mixture model.

If the variables z_{mn} are not known, then $\mathbf{z}$ are known as latent or hidden variables, and we cannot simply apply the maximum likelihood estimators straightforwardly.

Expectation Maximisation

EM Algorithm

Let us consider the joint distribution of all data points, $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, and latent variables, $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$, where each $\mathbf{z}_n = \{z_{1n}, \ldots, z_{Mn}\}$ are the indicator variables associated with nth data point.

Given a set of parameters $\theta = \{\theta_1, \ldots, \theta_M\}$ for each of the components, we can marginalise over all possible allocations of the data to the mixture components,

$$p(\mathbf{X}|\theta) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\theta)$$

EM Algorithm

(Note: log is a concave function)

Using Jensen's Inequality ($\log E\{f(X)\} \geq E\{\log f(X)\}$) we can therefore form a lower bound on the log likelihood,

$$\begin{aligned}
\log p(\mathbf{X}|\theta) &= \log \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\theta) \\
&= \log \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}) \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{p(\mathbf{Z}|\mathbf{X})} \\
&\geq \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}) \log \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{p(\mathbf{Z}|\mathbf{X})} \\
&= \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}) \log p(\mathbf{X}, \mathbf{Z}|\theta) - \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}) \log p(\mathbf{Z}|\mathbf{X})
\end{aligned}$$

EM Algorithm

We can now write this lower bound on the log likelihood in full, where we need to take a summation over all data points and all mixture components,

$$\begin{aligned}
\sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}) \log \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{p(\mathbf{Z}|\mathbf{X})} &= \sum_{m,n}^{M,N} p(m|\mathbf{x}_n) \log \frac{p(\mathbf{x}_n|\theta_m)p(m)}{p(m|\mathbf{x}_n)} \\
&= \sum_{m=1}^{M} \sum_{n=1}^{N} p(m|\mathbf{x}_n) \log p(\mathbf{x}_n|\theta_m)p(m) \\
&\quad - \sum_{m=1}^{M} \sum_{n=1}^{N} p(m|\mathbf{x}_n) \log p(m|\mathbf{x}_n)
\end{aligned}$$

where $p(m|\mathbf{x}_n)$ is the probability that $z_{mn} = 1$ and $p(m)$ is the probability that $z_{mn} = 1$ for any n.

EM Algorithm

The Expectation Maximisation (EM) algorithm is a general algorithm for *maximising* the likelihood of the complete data, $\mathbf{X}$ and $\mathbf{Z}$, so as to obtain estimates of the component parameters θ_m.

Given the some initial parameter values, we must first calculate the probability density of the latent variables, $p(\mathbf{Z}|\mathbf{X})$, with respect to which the *expectation* is taken.

After we have a set of latent variables, we can then perform a *maximisation* step to obtain the next estimate of the parameter values.

We repeat these two steps until some convergence criterion is satisfied.

Expectation Step

We can take functional derivatives of the lower bound expression, $\mathcal{L}_B$, with respect to $p(m|\mathbf{x}_n)$,

$$\frac{\partial \mathcal{L}_B}{\partial p(m|\mathbf{x}_n)} = \log p(m|\mathbf{x}_n) - \log p(\mathbf{x}_n|\theta_m)p(m) - 1$$

Setting this to zero we see that $p(m|\mathbf{x}_n) \propto p(\mathbf{x}_n|\theta_m)p(m)$ and so by normalising this probability appropriately we obtain,

$$p(m|\mathbf{x}_n) = \frac{p(\mathbf{x}_n|\theta_m)p(m)}{\sum_{m'} p(\mathbf{x}_n|\theta_{m'})p(m')}$$

This is just the posterior distribution over the mixture components m that generated the data $\mathbf{x}_n$.

Once we have maximised the bound with respect to the distribution of the indicator variables, which is used for the *expectation*, we can then *maximise* the bound with respect to the parameter values.

Maximisation Step

The only terms in the bound $\mathcal{L}_B$ that depend on the component parameters are

$$\sum_{m=1}^{M} \sum_{n=1}^{N} p(m|\mathbf{x}_n) \log p(\mathbf{x}_n|\theta_m)p(m)$$

which we can maximise with respect to the component parameters θ_m.

Example

As an example, let's assume each $p(\mathbf{x}_n|\theta_m)$ is a multivariate Gaussian distribution. Then the elements of the lower bound are,

$$\begin{aligned}
\mathcal{L}_B &= -\frac{1}{2} \sum_{m=1}^{M} \sum_{n=1}^{N} p(m|\mathbf{x}_n) \log |\Sigma_m| \\
&\quad -\frac{1}{2} \sum_{m=1}^{M} \sum_{n=1}^{N} p(m|\mathbf{x}_n)(\mathbf{x}_n - \mu_m)^T \Sigma_m^{-1} (\mathbf{x}_n - \mu_m) \\
&\quad +\frac{1}{2} \sum_{m=1}^{M} \sum_{n=1}^{N} p(m|\mathbf{x}_n) \log p(m)
\end{aligned}$$

Example

We can take derivatives with respect to the parameter μ_m and set equal to zero, to obtain

$$\hat{\mu}_m = \frac{\sum_{n=1}^{N} p(m|\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} p(m|\mathbf{x}_n)}$$

We can compare this estimator to the case where we have perfect knowledge of the indicator variables, z_{mn},

$$\hat{\mu}_m = \frac{\sum_{n=1}^{N} z_{mn}\mathbf{x}_n}{\sum_{n=1}^{N} z_{mn}}$$

So if we don't know the true indicator values, we simply use their posterior probabilities, which we calculated in the expectation step.

Example

Similarly, we can obtain the maximum value of the covariance parameters,

$$\Sigma_m = \frac{\sum_{n=1}^{N} p(m|\mathbf{x}_n)(\mathbf{x}_n - \hat{\mu}_m)(\mathbf{x}_n - \hat{\mu}_m)^T}{\sum_{n=1}^{N} p(m|\mathbf{x}_n)}$$

where again we see that we are using the posterior probabilit the indicator variables, since we don't observe them directly.

And we can get an updated estimate of $p(m)$ by taking derivatives, setting to zero, solving and normalising, to obtain,

$$p(m) = \frac{1}{N} \sum_{n=1}^{N} p(m|\mathbf{x}_n)$$

EM Algorithm

Let us consider the joint distribution of all data points, $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, and latent variables, $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$, where each $\mathbf{z}_n = \{z_{1n}, \ldots, z_{Mn}\}$ are the indicator variables associated with nth data point.

Given a set of parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$ for each of the components, we can marginalise over all possible allocations of the data to the mixture components,

$$p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$$

EM Algorithm

(Note: log is a concave function)

Using Jensen's Inequality ($\log E\{f(X)\} \geq E\{\log f(X)\}$) we can therefore form a lower bound on the log likelihood,

$$
\begin{aligned}
\log p(\mathbf{X}|\boldsymbol{\theta}) &= \log \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) \\
&= \log \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}) \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{p(\mathbf{Z}|\mathbf{X})} \\
&\geq \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}) \log \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{p(\mathbf{Z}|\mathbf{X})} \\
&= \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}) \log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) - \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}) \log p(\mathbf{Z}|\mathbf{X})
\end{aligned}
$$

EM Algorithm

We can now write this lower bound on the log likelihood in full, where we need to take a summation over all data points and all mixture components,

$$\begin{aligned}\sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}) \log \frac{p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})}{p(\mathbf{Z}|\mathbf{X})} &= \sum_{m,n}^{M,N} p(m|\mathbf{x}_n) \log \frac{p(\mathbf{x}_n|\boldsymbol{\theta}_m)p(m)}{p(m|\mathbf{x}_n)} \\ &= \sum_{m=1}^{M}\sum_{n=1}^{N} p(m|\mathbf{x}_n) \log p(\mathbf{x}_n|\boldsymbol{\theta}_m)p(m) \\ &\quad - \sum_{m=1}^{M}\sum_{n=1}^{N} p(m|\mathbf{x}_n) \log p(m|\mathbf{x}_n)\end{aligned}$$

where $p(m|\mathbf{x}_n)$ is the probability that $z_{mn} = 1$ and $p(m)$ is the probability that $z_{mn} = 1$ for any n.

EM Algorithm

The Expectation Maximisation (EM) algorithm is a general algorithm for *maximising* the likelihood of the complete data, $\mathbf{X}$ and $\mathbf{Z}$, so as to obtain estimates of the component parameters $\boldsymbol{\theta}_m$.

Given the some initial parameter values, we must first calculate the probability density of the latent variables, $p(\mathbf{Z}|\mathbf{X})$, with respect to which the *expectation* is taken.

After we have a set of latent variables, we can then perform a *maximisation* step to obtain the next estimate of the parameter values.

We repeat these two steps until some convergence criterion is satisfied.

Expectation Step

We can take functional derivatives of the lower bound expression, $\mathcal{L}_B$, with respect to $p(m|\mathbf{x}_n)$,

$$\frac{\partial \mathcal{L}_B}{\partial p(m|\mathbf{x}_n)} = \log p(m|\mathbf{x}_n) - \log p(\mathbf{x}_n|\boldsymbol{\theta}_m)p(m) - 1$$

Setting this to zero we see that $p(m|\mathbf{x}_n) \propto p(\mathbf{x}_n|\boldsymbol{\theta}_m)p(m)$ and so by normalising this probability appropriately we obtain,

$$p(m|\mathbf{x}_n) = \frac{p(\mathbf{x}_n|\boldsymbol{\theta}_m)p(m)}{\sum_{m'} p(\mathbf{x}_n|\boldsymbol{\theta}_{m'})p(m')}$$

This is just the posterior distribution over the mixture components m that generated the data $\mathbf{x}_n$.

Once we have maximised the bound with respect to the distribution of the indicator variables, which is used for the *expectation*, we can then *maximise* the bound with respect to the parameter values.

Maximisation Step

The only terms in the bound $\mathcal{L}_B$ that depend on the component parameters are

$$\sum_{m=1}^{M} \sum_{n=1}^{N} p(m|\mathbf{x}_n) \log p(\mathbf{x}_n|\boldsymbol{\theta}_m) p(m)$$

which we can maximise with respect to the component parameters $\boldsymbol{\theta}_m$.

Example

As an example, let's assume each $p(\mathbf{x}_n|\boldsymbol{\theta}_m)$ is a multivariate Gaussian distribution. Then the elements of the lower bound are,

$$\begin{aligned}
\mathcal{L}_B &= -\frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{N} p(m|\mathbf{x}_n)\log|\boldsymbol{\Sigma}_m| \\
&\quad -\frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{N} p(m|\mathbf{x}_n)(\mathbf{x}_n-\boldsymbol{\mu}_m)^T\boldsymbol{\Sigma}_m^{-1}(\mathbf{x}_n-\boldsymbol{\mu}_m) \\
&\quad +\frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{N} p(m|\mathbf{x}_n)\log p(m)
\end{aligned}$$

Example

We can take derivatives with respect to the parameter $\boldsymbol{\mu}_m$ and set equal to zero, to obtain

$$\hat{\boldsymbol{\mu}}_m = \frac{\sum_{n=1}^{N} p(m|\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} p(m|\mathbf{x}_n)}$$

We can compare this estimator to the case where we have perfect knowledge of the indicator variables, z_{mn},

$$\hat{\boldsymbol{\mu}}_m = \frac{\sum_{n=1}^{N} z_{mn}\mathbf{x}_n}{\sum_{n=1}^{N} z_{mn}}$$

So if we don't know the true indicator values, we simply use their posterior probabilities, which we calculated in the expectation step.

Example

Similarly, we can obtain the maximum value of the covariance parameters,

$$\hat{\boldsymbol{\Sigma}}_m = \frac{\sum_{n=1}^{N} p(m|\mathbf{x}_n)(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_m)(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_m)^T}{\sum_{n=1}^{N} p(m|\mathbf{x}_n)}$$

where again we see that we are using the posterior probabilit the indicator variables, since we don't observe them directly.

And we can get an updated estimate of $p(m)$ by taking derivatives, setting to zero, solving and normalising, to obtain,

$$p(m) = \frac{1}{N} \sum_{n=1}^{N} p(m|\mathbf{x}_n)$$

Expectation Maximisation

EM Algorithm

Let us consider the joint distribution of all data points, $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, and latent variables, $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$, where each $\mathbf{z}_n = \{z_{1n}, \ldots, z_{Mn}\}$ are the indicator variables associated with nth data point.

Given a set of parameters $\theta = \{\theta_1, \ldots, \theta_M\}$ for each of the components, we can marginalise over all possible allocations of the data to the mixture components,

$$p(\mathbf{X}|\theta) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\theta)$$

EM Algorithm

(Note: log is a concave function)

Using Jensen's Inequality ($\log E\{f(X)\} \geq E\{\log f(X)\}$) we can therefore form a lower bound on the log likelihood,

$$\begin{aligned}
\log p(\mathbf{X}|\theta) &= \log \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\theta) \\
&= \log \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}) \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{p(\mathbf{Z}|\mathbf{X})} \\
&\geq \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}) \log \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{p(\mathbf{Z}|\mathbf{X})} \\
&= \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}) \log p(\mathbf{X}, \mathbf{Z}|\theta) - \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}) \log p(\mathbf{Z}|\mathbf{X})
\end{aligned}$$

EM Algorithm

We can now write this lower bound on the log likelihood in full, where we need to take a summation over all data points and all mixture components,

$$\begin{aligned}
\sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}) \log \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{p(\mathbf{Z}|\mathbf{X})} &= \sum_{m,n}^{M,N} p(m|\mathbf{x}_n) \log \frac{p(\mathbf{x}_n|\theta_m)p(m)}{p(m|\mathbf{x}_n)} \\
&= \sum_{m=1}^{M} \sum_{n=1}^{N} p(m|\mathbf{x}_n) \log p(\mathbf{x}_n|\theta_m)p(m) \\
&\quad - \sum_{m=1}^{M} \sum_{n=1}^{N} p(m|\mathbf{x}_n) \log p(m|\mathbf{x}_n)
\end{aligned}$$

where $p(m|\mathbf{x}_n)$ is the probability that $z_{mn} = 1$ and $p(m)$ is the probability that $z_{mn} = 1$ for any n.

EM Algorithm

The Expectation Maximisation (EM) algorithm is a general algorithm for *maximising* the likelihood of the complete data, $\mathbf{X}$ and $\mathbf{Z}$, so as to obtain estimates of the component parameters θ_m.

Given the some initial parameter values, we must first calculate the probability density of the latent variables, $p(\mathbf{Z}|\mathbf{X})$, with respect to which the *expectation* is taken.

After we have a set of latent variables, we can then perform a *maximisation* step to obtain the next estimate of the parameter values.

We repeat these two steps until some convergence criterion is satisfied.

Expectation Step

We can take functional derivatives of the lower bound expression, $\mathcal{L}_B$, with respect to $p(m|\mathbf{x}_n)$,

$$\frac{\partial \mathcal{L}_B}{\partial p(m|\mathbf{x}_n)} = \log p(m|\mathbf{x}_n) - \log p(\mathbf{x}_n|\theta_m)p(m) - 1$$

Setting this to zero we see that $p(m|\mathbf{x}_n) \propto p(\mathbf{x}_n|\theta_m)p(m)$ and so by normalising this probability appropriately we obtain,

$$p(m|\mathbf{x}_n) = \frac{p(\mathbf{x}_n|\theta_m)p(m)}{\sum_{m'} p(\mathbf{x}_n|\theta_{m'})p(m')}$$

This is just the posterior distribution over the mixture components m that generated the data $\mathbf{x}_n$.

Once we have maximised the bound with respect to the distribution of the indicator variables, which is used for the *expectation*, we can then *maximise* the bound with respect to the parameter values.

Maximisation Step

The only terms in the bound $\mathcal{L}_B$ that depend on the component parameters are

$$\sum_{m=1}^{M} \sum_{n=1}^{N} p(m|\mathbf{x}_n) \log p(\mathbf{x}_n|\theta_m)p(m)$$

which we can maximise with respect to the component parameters θ_m.

Example

As an example, let's assume each $p(\mathbf{x}_n|\theta_m)$ is a multivariate Gaussian distribution. Then the elements of the lower bound are,

$$\begin{aligned}
\mathcal{L}_B &= -\frac{1}{2} \sum_{m=1}^{M} \sum_{n=1}^{N} p(m|\mathbf{x}_n) \log |\Sigma_m| \\
&\quad -\frac{1}{2} \sum_{m=1}^{M} \sum_{n=1}^{N} p(m|\mathbf{x}_n)(\mathbf{x}_n - \mu_m)^T \Sigma_m^{-1} (\mathbf{x}_n - \mu_m) \\
&\quad +\frac{1}{2} \sum_{m=1}^{M} \sum_{n=1}^{N} p(m|\mathbf{x}_n) \log p(m)
\end{aligned}$$

Example

We can take derivatives with respect to the parameter μ_m and set equal to zero, to obtain

$$\hat{\mu}_m = \frac{\sum_{n=1}^{N} p(m|\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} p(m|\mathbf{x}_n)}$$

We can compare this estimator to the case where we have perfect knowledge of the indicator variables, z_{mn},

$$\hat{\mu}_m = \frac{\sum_{n=1}^{N} z_{mn}\mathbf{x}_n}{\sum_{n=1}^{N} z_{mn}}$$

So if we don't know the true indicator values, we simply use their posterior probabilities, which we calculated in the expectation step.

Example

Similarly, we can obtain the maximum value of the covariance parameters,

$$\Sigma_m = \frac{\sum_{n=1}^{N} p(m|\mathbf{x}_n)(\mathbf{x}_n - \hat{\mu}_m)(\mathbf{x}_n - \hat{\mu}_m)^T}{\sum_{n=1}^{N} p(m|\mathbf{x}_n)}$$

where again we see that we are using the posterior probabilit the indicator variables, since we don't observe them directly.

And we can get an updated estimate of $p(m)$ by taking derivatives, setting to zero, solving and normalising, to obtain,

$$p(m) = \frac{1}{N} \sum_{n=1}^{N} p(m|\mathbf{x}_n)$$

Examples

Overview of EM

In summary, the *E* step consists of

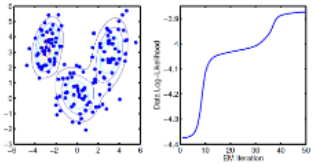
$$p(m|\mathbf{x}_n) = \frac{p(\mathbf{x}_n|\boldsymbol{\theta}_m)p(m)}{\sum_{m'} p(\mathbf{x}_n|\boldsymbol{\theta}_{m'})p(m')}$$

and the *M* step consists of

$$\begin{aligned}
\hat{\boldsymbol{\mu}}_m &= \frac{\sum_{n=1}^{N} p(m|\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} p(m|\mathbf{x}_n)} \\
\hat{\boldsymbol{\Sigma}}_m &= \frac{\sum_{n=1}^{N} p(m|\mathbf{x}_n)(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_m)(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_m)^T}{\sum_{n=1}^{N} p(m|\mathbf{x}_n)} \\
p(m) &= \frac{1}{N}\sum_{n=1}^{N} p(m|\mathbf{x}_n)
\end{aligned}$$

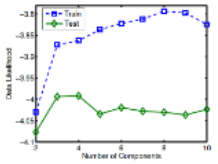
Gaussian Mixture Example

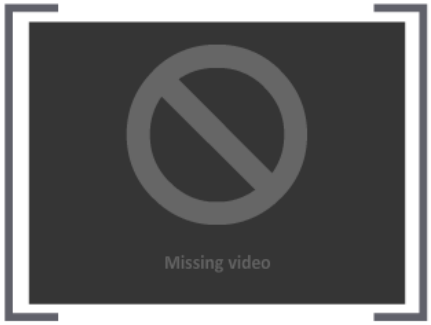
Given 150 data points drawn from a mixture of three Gaussian distributions, with unit variance and means $[0, 0]$, $[3, 3]$ and $[-3, 3]$, we can use the EM algorithm to learn the generating mixture of distributions and their parameters.



Gaussian Mixture Example

Of course we also won't know how many components there actually are in the most appropriate mixture distribution. Once again we can use cross validation.





Overview of EM

In summary, the *E* step consists of

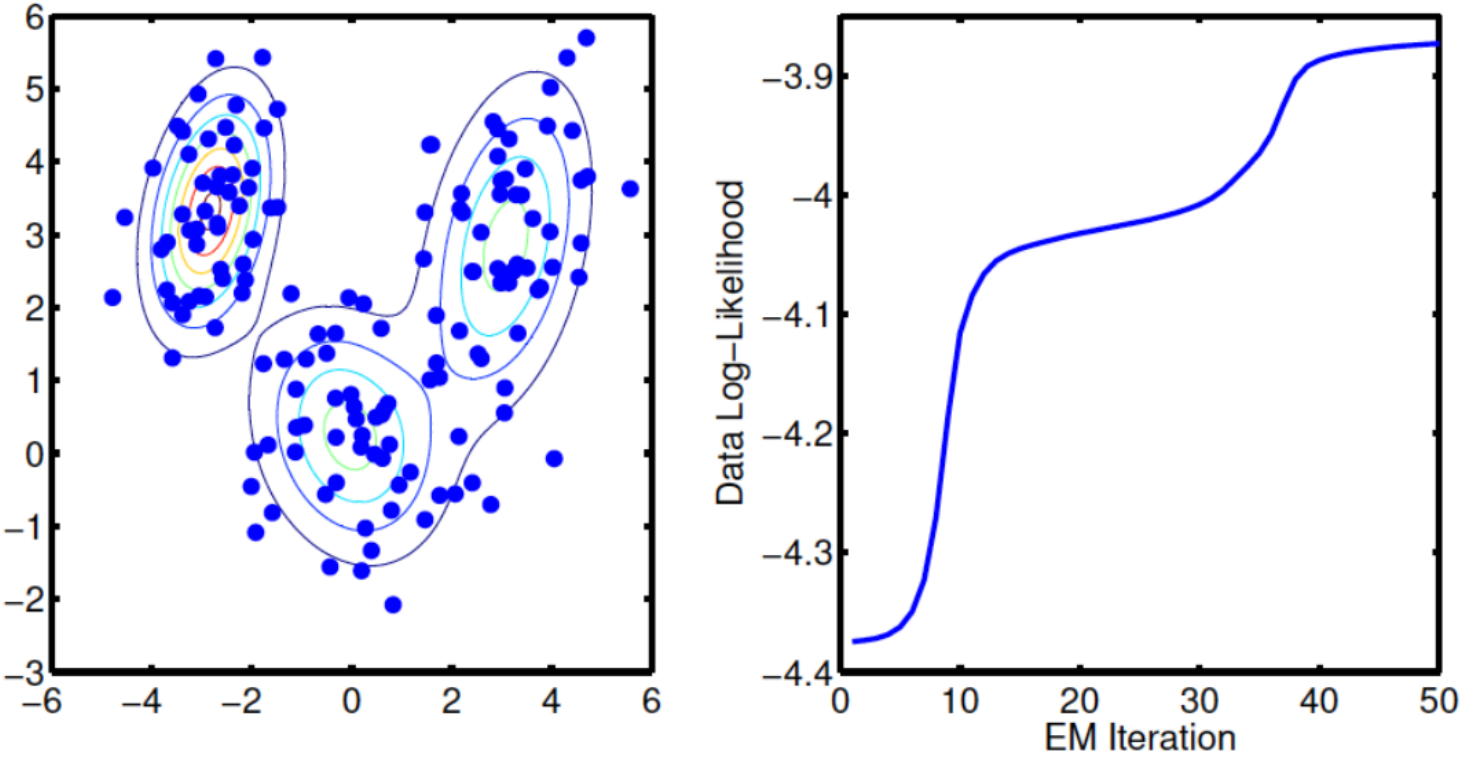
$$p(m|\mathbf{x}_n) = \frac{p(\mathbf{x}_n|\boldsymbol{\theta}_m)p(m)}{\sum_{m'} p(\mathbf{x}_n|\boldsymbol{\theta}_{m'})p(m')}$$

and the *M* step consists of

$$\begin{aligned}
\hat{\boldsymbol{\mu}}_m &= \frac{\sum_{n=1}^{N} p(m|\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} p(m|\mathbf{x}_n)} \\
\hat{\boldsymbol{\Sigma}}_m &= \frac{\sum_{n=1}^{N} p(m|\mathbf{x}_n)(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_m)(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_m)^T}{\sum_{n=1}^{N} p(m|\mathbf{x}_n)} \\
p(m) &= \frac{1}{N}\sum_{n=1}^{N} p(m|\mathbf{x}_n)
\end{aligned}$$

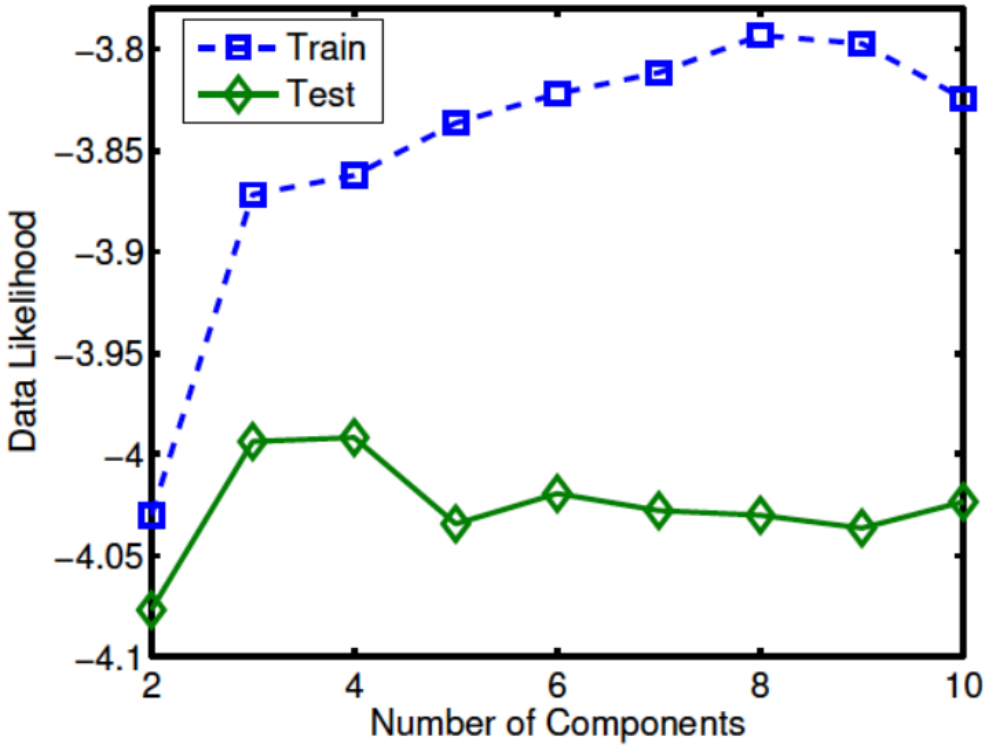
Gaussian Mixture Example

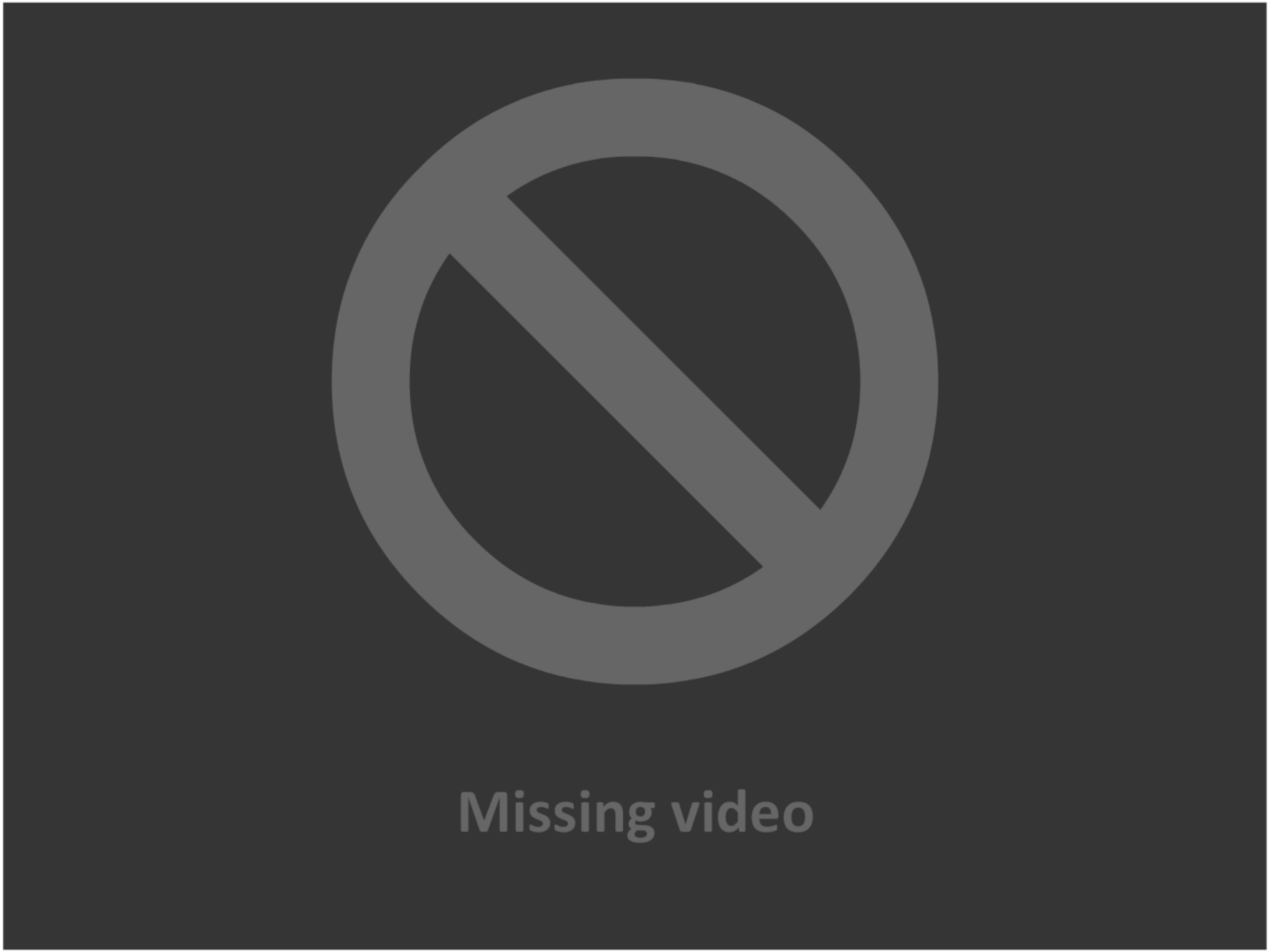
Given 150 data points drawn from a mixture of three Gaussian distributions, with unit variance and means $[0, 0]$, $[3, 3]$ and $[-3, 3]$, we can use the EM algorithm to learn the generating mixture of distributions and their parameters.



Gaussian Mixture Example

Of course we also won't know how many components there actually are in the most appropriate mixture distribution. Once again we can use cross validation.




Missing video

Examples

Overview of EM

In summary, the *E* step consists of

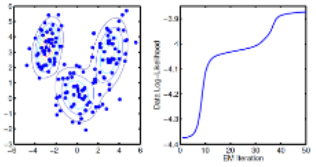
$$p(m|\mathbf{x}_n) = \frac{p(\mathbf{x}_n|\boldsymbol{\theta}_m)p(m)}{\sum_{m'} p(\mathbf{x}_n|\boldsymbol{\theta}_{m'})p(m')}$$

and the *M* step consists of

$$\begin{aligned}
\hat{\boldsymbol{\mu}}_m &= \frac{\sum_{n=1}^{N} p(m|\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} p(m|\mathbf{x}_n)} \\
\hat{\boldsymbol{\Sigma}}_m &= \frac{\sum_{n=1}^{N} p(m|\mathbf{x}_n)(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_m)(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_m)^T}{\sum_{n=1}^{N} p(m|\mathbf{x}_n)} \\
p(m) &= \frac{1}{N}\sum_{n=1}^{N} p(m|\mathbf{x}_n)
\end{aligned}$$

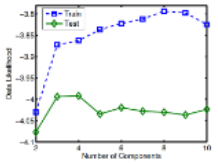
Gaussian Mixture Example

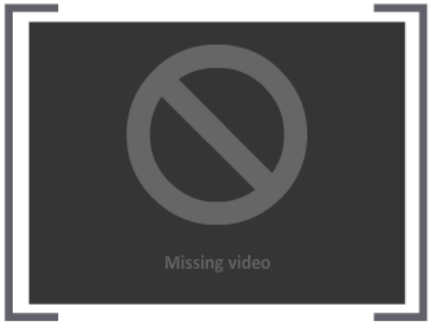
Given 150 data points drawn from a mixture of three Gaussian distributions, with unit variance and means $[0, 0]$, $[3, 3]$ and $[-3, 3]$, we can use the EM algorithm to learn the generating mixture of distributions and their parameters.



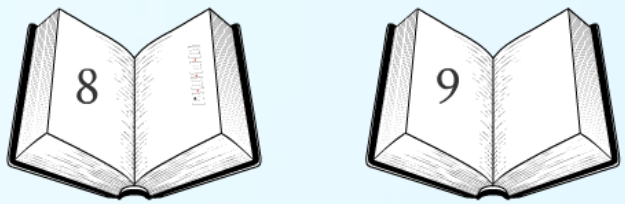
Gaussian Mixture Example

Of course we also won't know how many components there actually are in the most appropriate mixture distribution. Once again we can use cross validation.





Clustering and Density Estimation



M5MS10
Machine Learning

Spring 2018

Lectures 7/8

Dr Ben Calderhead

b.calderhead@imperial.ac.uk