UNIVERSITY
of
GLASGOW

Machine Learning

Lecture. 11.

Mark Girolami

girolami@dcs.gla.ac.uk

Department of Computing Science
University of Glasgow

Density Estimation

UNIVERSITY
of
GLASGOW

- Fundamental ML problem

Density Estimation

- Fundamental ML problem
- Parametric Probability Density Estimation

Density Estimation

UNIVERSITY
of
GLASGOW

- Fundamental ML problem
- Parametric Probability Density Estimation
- Maximum Likelihood Estimation

Density Estimation

UNIVERSITY
of
GLASGOW

- Fundamental ML problem
- Parametric Probability Density Estimation
- Maximum Likelihood Estimation
- Mixture Density Models

Density Estimation

- Fundamental ML problem
- Parametric Probability Density Estimation
- Maximum Likelihood Estimation
- Mixture Density Models
- EM Algorithm

Density Estimation

- Class conditional density $p(\mathbf{x}|C = k)$ denote by functional parametric form $p(\mathbf{x}|\boldsymbol{\theta}_k)$

Density Estimation

- Class conditional density $p(\mathbf{x}|C = k)$ denote by functional parametric form $p(\mathbf{x}|\boldsymbol{\theta}_k)$
- Given data and labels require estimation of each set of $\boldsymbol{\theta}_k$

Density Estimation

- Class conditional density $p(\mathbf{x}|C = k)$ denote by functional parametric form $p(\mathbf{x}|\boldsymbol{\theta}_k)$
- Given data and labels require estimation of each set of $\boldsymbol{\theta}_k$
- Employ likelihood function and estimate parameters which maximise the likelihood

MLE for Gaussian

- N_k examples from class k, assume D features are distributed as Multivariate Gaussian. Likelihood is $\mathcal{L}_k$

MLE for Gaussian

- N_k examples from class k, assume D features are distributed as Multivariate Gaussian. Likelihood is $\mathcal{L}_k$

$$\prod_{n=1}^{N_k} p(\mathbf{x}_n|\boldsymbol{\theta}_k) = \prod_{n=1}^{N_k} p(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$= \prod_{n=1}^{N_k} \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}_k|}} \exp\left\{-\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k)\right\}$$

MLE for Gaussian

- N_k examples from class k, assume D features are distributed as Multivariate Gaussian. Likelihood is $\mathcal{L}_k$

$$\prod_{n=1}^{N_k} p(\mathbf{x}_n|\boldsymbol{\theta}_k) = \prod_{n=1}^{N_k} p(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$= \prod_{n=1}^{N_k} \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}_k|}} \exp\left\{-\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k)\right\}$$

Work with logarithm of likelihood $\log \mathcal{L}_k$ and drop constant $-\frac{N_k D}{2}\log 2\pi$

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

MLE for Gaussian

- In this case each $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ and so we take derivatives of $\log \mathcal{L}_k$

MLE for Gaussian

- In this case each $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ and so we take derivatives of $\log \mathcal{L}_k$
- Expand the quadratic term and drop all terms independent of $\boldsymbol{\mu}_k$ then

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

MLE for Gaussian

- In this case each $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ and so we take derivatives of $\log \mathcal{L}_k$
- Expand the quadratic term and drop all terms independent of $\boldsymbol{\mu}_k$ then

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

- Vector derivatives obtain

$$\frac{\partial}{\partial \boldsymbol{\mu}_k} \log \mathcal{L}_k = \sum_{n=1}^{N_k} \{ \boldsymbol{\Sigma}_k^{-1} \mathbf{x}_n - \boldsymbol{\Sigma}_k^{-1} \boldsymbol{\mu}_k \}$$

MLE for Gaussian

UNIVERSITY
of
GLASGOW

- Setting the gradient to zero we then obtain

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

MLE for Gaussian

UNIVERSITY
of
GLASGOW

- Setting the gradient to zero we then obtain

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

- Now we can multiply both sides by the matrix $\boldsymbol{\Sigma}_k$ to obtain $\sum_{n=1}^{N_k} \mathbf{x}_n = N_k \boldsymbol{\mu}_k$

MLE for Gaussian

UNIVERSITY *of* GLASGOW

- Setting the gradient to zero we then obtain

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

- Now we can multiply both sides by the matrix $\mathbf{\Sigma}_k$ to obtain $\sum_{n=1}^{N_k} \mathbf{x}_n = N_k \boldsymbol{\mu}_k$
- Maximum-Likelihood estimate for the mean of the class-conditional Multivariate Gaussian as

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

MLE for Gaussian

- Do the same for the ML estimate of the required covariance matrix $\mathbf{\Sigma}_k$

MLE for Gaussian

- Do the same for the ML estimate of the required covariance matrix $\boldsymbol{\Sigma}_k$
- From Section 2.1.2 of the Matrix Cookbook we have the following equality

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

MLE for Gaussian

- Do the same for the ML estimate of the required covariance matrix $\boldsymbol{\Sigma}_k$
- From Section 2.1.2 of the Matrix Cookbook we have the following equality

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

- So

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

MLE for Gaussian

UNIVERSITY *of* GLASGOW

- Cookbook, Section 2.2 expression third from bottom, shows that $\frac{\partial}{\partial \mathbf{X}} \mathbf{a}^\mathsf{T} \mathbf{X}^{-1} \mathbf{b} = -\mathbf{X}^{-1} \mathbf{a} \mathbf{b}^\mathsf{T} \mathbf{X}^{-1}$

MLE for Gaussian

- Cookbook, Section 2.2 expression third from bottom, shows that $\frac{\partial}{\partial \mathbf{X}} \mathbf{a}^\mathsf{T} \mathbf{X}^{-1} \mathbf{b} = -\mathbf{X}^{-1} \mathbf{a} \mathbf{b}^\mathsf{T} \mathbf{X}^{-1}$ using this expression then $\frac{\partial}{\partial \boldsymbol{\Sigma}_k} \sum_{n=1}^{N_k} (\mathbf{x}_n - \boldsymbol{\mu}_k)^\mathsf{T} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k)$ equals

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

MLE for Gaussian

- Cookbook, Section 2.2 expression third from bottom, shows that $\frac{\partial}{\partial \mathbf{X}} \mathbf{a}^{\mathsf{T}} \mathbf{X}^{-1} \mathbf{b} = -\mathbf{X}^{-1} \mathbf{a} \mathbf{b}^{\mathsf{T}} \mathbf{X}^{-1}$ using this expression then $\frac{\partial}{\partial \boldsymbol{\Sigma}_k} \sum_{n=1}^{N_k} (\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathsf{T}} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k)$ equals

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

  Plugging everything together then we obtain

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

MLE for Gaussian

- Setting gradient to zero, replacing mean vectors with their ML estimates, after a little manipulation the estimate for the class-conditioned covariance is, as we would expect

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

MLE for Gaussian

- Setting gradient to zero, replacing mean vectors with their ML estimates, after a little manipulation the estimate for the class-conditioned covariance is, as we would expect

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

- ML estimation method can be adopted for any parametric form of probability density or distribution function. Of course we can also adopt a Bayesian approach by setting appropriate priors for the mean and covariance terms - we will resist this temptation for the time being

Illustrative Examples

- Matlab script `gauss_density_est.m` generates random sample drawn from 2D Gaussian with parameters

$$\boldsymbol{\mu} = \begin{bmatrix} 1.0 \\ 3.0 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1.5 & 0.6 \\ 0.6 & 0.4 \end{bmatrix}$$

Illustrative Examples

- Matlab script `gauss_density_est.m` generates random sample drawn from 2D Gaussian with parameters

$$\boldsymbol{\mu} = \begin{bmatrix} 1.0 \\ 3.0 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1.5 & 0.6 \\ 0.6 & 0.4 \end{bmatrix}$$

- Use sample to obtain estimates for the required parameters clearly sample size $N \to \infty$ then estimates converge to true values

Illustrative Examples

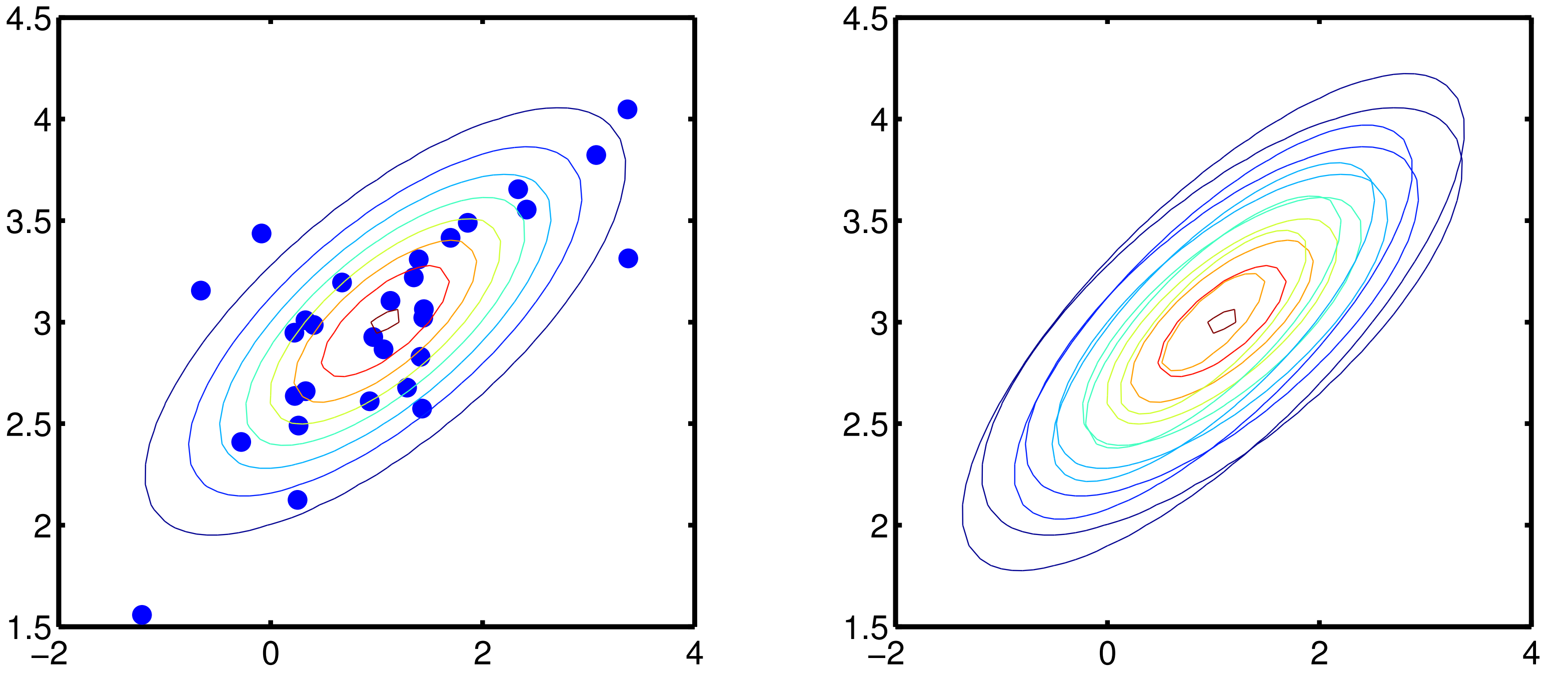

Figure 1: The left plot shows a random sample of 30 points drawn from a 2D Gaussian, the iso-contours of estimated probability density are superimposed on the plot. The iso-contours of probability density for the Gaussian with the actual parameter values are given on the right hand plot superimposed upon the iso-contours of estimated density.

Non-Gaussian Example

- Now consider an example of data for which we, wrongly, assume that the density is also Gaussian.

Non-Gaussian Example

- Now consider an example of data for which we, wrongly, assume that the density is also Gaussian.
- The density is a mixture of two Gaussians with mean and covariances of

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 0.5 \\ 2.0 \end{bmatrix} \quad \mathbf{C}_1 = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 3.0 \\ 4.0 \end{bmatrix} \quad \mathbf{C}_2 = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

Non-Gaussian Example

- Now consider an example of data for which we, wrongly, assume that the density is also Gaussian.
- The density is a mixture of two Gaussians with mean and covariances of

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 0.5 \\ 2.0 \end{bmatrix} \quad \mathbf{C}_1 = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 3.0 \\ 4.0 \end{bmatrix} \quad \mathbf{C}_2 = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

- Data considered as coming from two sub-populations, or there are two distinct generating processes each responsible for producing the data we observe.

Non-Gaussian Example

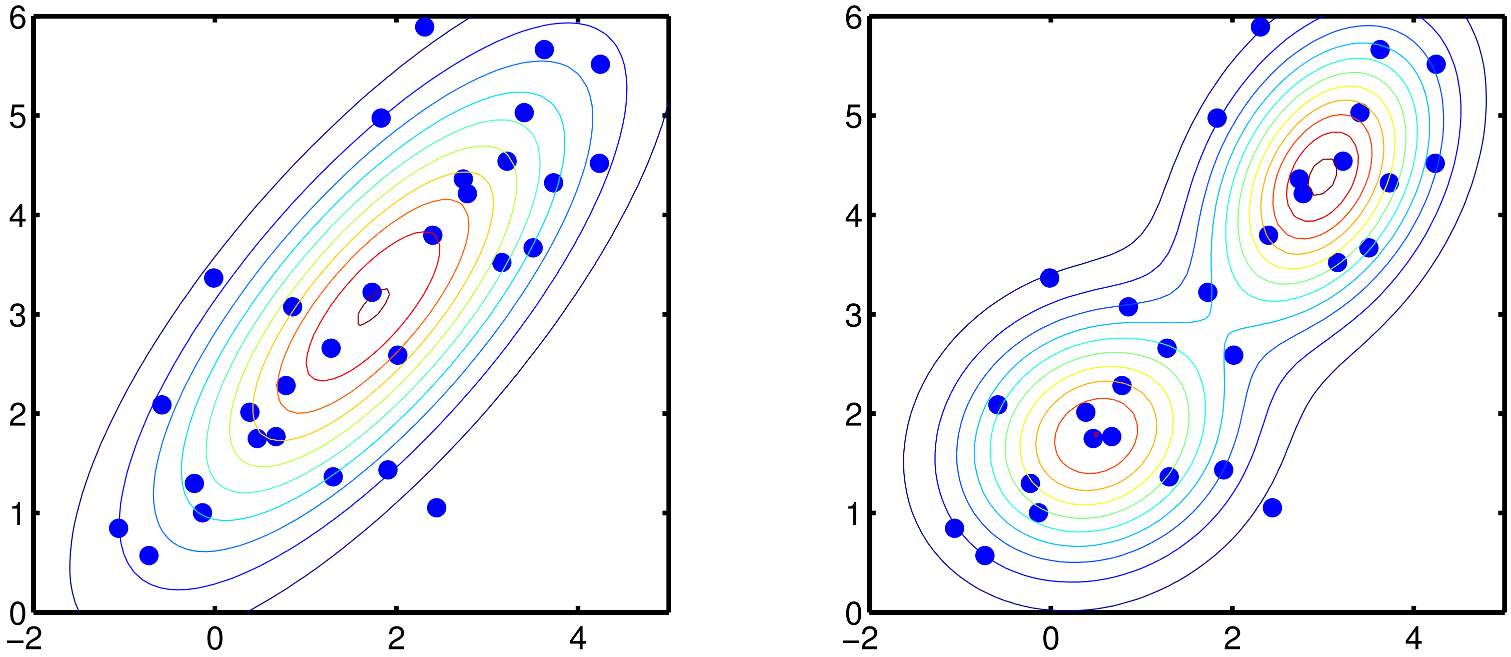

Figure 2: The left plot shows a random sample of 30 points drawn from 2 distinct 2D Gaussians, the iso-contours of estimated probability density under the assumption that the density is a single 2D Gaussian, are superimposed on the plot. The right plot shows the same random sample with the iso-contours of estimated probability density where the true functional form has been employed i.e. two 2D Gaussians.

Density Estimation

- Average likelihood of points spread uniformly across the regions shown in the figures assuming a single Gaussian is -3.261.

Density Estimation

- Average likelihood of points spread uniformly across the regions shown in the figures assuming a single Gaussian is -3.261.
- Average likelihood of points spread uniformly across the regions shown in the figures assuming a mixture of two Gaussians is -3.123.

Density Estimation

- Average likelihood of points spread uniformly across the regions shown in the figures assuming a single Gaussian is -3.261.
- Average likelihood of points spread uniformly across the regions shown in the figures assuming a mixture of two Gaussians is -3.123.
- This is higher than that achieved when assuming a single Gaussian and so provides a superior predictive generative model of the data.

Density Estimation

- Average likelihood of points spread uniformly across the regions shown in the figures assuming a single Gaussian is -3.261.
- Average likelihood of points spread uniformly across the regions shown in the figures assuming a mixture of two Gaussians is -3.123.
- This is higher than that achieved when assuming a single Gaussian and so provides a superior predictive generative model of the data.
- A Matlab script `mix_gauss_density.m` is available on the course website to allow you to replicate these results.

Mixture Models

- The probability density function for the case of two Gaussians can be represented as

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

Mixture Models

- The probability density function for the case of two Gaussians can be represented as

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

- where $\boldsymbol{\theta} = \{\pi, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$ and each set of parameters is defined by $\boldsymbol{\theta}_1 = \{\boldsymbol{\mu}_1, \mathrm{C}_1\}$ and $\boldsymbol{\theta}_2 = \{\boldsymbol{\mu}_2, \mathrm{C}_2\}$

Mixture Models

UNIVERSITY
of
GLASGOW

- The probability density function for the case of two Gaussians can be represented as

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

- where $\boldsymbol{\theta} = \{\pi, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$ and each set of parameters is defined by $\boldsymbol{\theta}_1 = \{\boldsymbol{\mu}_1, \mathrm{C}_1\}$ and $\boldsymbol{\theta}_2 = \{\boldsymbol{\mu}_2, \mathrm{C}_2\}$
- The parameter π is the probability that a point $\mathbf{x}$ will be generated from $p(\mathbf{x}|\boldsymbol{\theta}_1)$ and so the probability that the point will be generated from $p(\mathbf{x}|\boldsymbol{\theta}_2)$ is $1-\pi$

Mixture Models

- In the more general case where there are M components, of arbitrary parametric form the probability density will be expressed as

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

Mixture Models

UNIVERSITY
of
GLASGOW

- In the more general case where there are M components, of arbitrary parametric form the probability density will be expressed as

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

- where the whole parameter set is defined as $\boldsymbol{\theta} = \{\pi_1 \cdots \pi_M, \boldsymbol{\theta}_1 \cdots \boldsymbol{\theta}_M\}$ and $\sum_{m=1}^{M} \pi_m = 1$ as each π_m is the probability that the mth component of the mixture will produce a data point so it must sum to one to be a valid probability over the M selection events

Mixture Models

- Given data $\mathcal{D} = \{\mathbf{x}_1 \cdots \mathbf{x}_N\}$ assuming mixture model $p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{m=1}^{M} \pi_m p(\mathbf{x}|\boldsymbol{\theta}_m)$ estimate parameters

Mixture Models

UNIVERSITY
of
GLASGOW

- Given data $\mathcal{D} = \{\mathbf{x}_1 \cdots \mathbf{x}_N\}$ assuming mixture model $p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{m=1}^{M} \pi_m p(\mathbf{x}|\boldsymbol{\theta}_m)$ estimate parameters
- Require estimates of each π_m, probability of data being generated by each m, just count how many points from $\mathcal{D}$ coming from each of M components then normalise by N. Count N_m points in $\mathcal{D}$ drawn from component m then

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

where each N_m can be obtained from $N_m = \sum_{n=1}^{N} z_{mn}$ where each $z_{mn} = 1$ if the nth point was drawn from component m and $z_{mn} = 0$ otherwise.

Mixture Models

- What of the specific parameters of each of the components $\boldsymbol{\theta}_m$? This is also easy as all we need to do is obtain the estimates $\widehat{\boldsymbol{\theta}}_m$ which maximise the likelihood of the data points which were drawn from component m under the parametric form $p(\mathbf{x}|\boldsymbol{\theta}_m)$.

Mixture Models

- What of the specific parameters of each of the components $\boldsymbol{\theta}_m$? This is also easy as all we need to do is obtain the estimates $\widehat{\boldsymbol{\theta}}_m$ which maximise the likelihood of the data points which were drawn from component m under the parametric form $p(\mathbf{x}|\boldsymbol{\theta}_m)$.
- For example if the mixture components were Gaussians then the Maximum-Likelihood estimate for the component mean vectors would simply be

$$\widehat{\boldsymbol{\mu}}_m = \frac{\sum_{n=1}^{N} z_{nm}\mathbf{x}_n}{\sum_{n=1}^{N} z_{nm}} = \frac{1}{N_m}\sum_{n\in m}\mathbf{x}_n$$

Mixture Models

- The expression for the covariance matrices for each component would follow simply as

$$\widehat{\mathbf{\Sigma}}_m = \frac{1}{N_m}\sum_{n=1}^{N} z_{mn}(\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_m)(\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_m)^{\mathsf{T}}$$

and we are then finished.

Mixture Models

- The expression for the covariance matrices for each component would follow simply as

$$\widehat{\boldsymbol{\Sigma}}_m = \frac{1}{N_m}\sum_{n=1}^{N} z_{mn}(\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_m)(\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_m)^{\mathsf{T}}$$

  and we are then finished.

- There is one small difficulty which we have overlooked, we do not have values for the indictor variables z_{mn} on which we have relied.

Mixture Models

UNIVERSITY
of
GLASGOW

- The expression for the covariance matrices for each component would follow simply as

$$\widehat{\mathbf{\Sigma}}_m = \frac{1}{N_m} \sum_{n=1}^{N} z_{mn} (\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_m)(\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_m)^{\mathsf{T}}$$

  and we are then finished.

- There is one small difficulty which we have overlooked, we do not have values for the indictor variables z_{mn} on which we have relied.
- This is a major difficulty as the fact that the variables z_{mn} are hidden or latent then our ML estimates cannot follow in the straightforward manner we had anticipated.

The EM Algorithm

- The problem is that we assumed knowledge of the values for the allocation or indicator variables z_{mn}

The EM Algorithm

- The problem is that we assumed knowledge of the values for the allocation or indicator variables z_{mn}
- Need the joint likelihood of data $\mathbf{X} = \{\mathbf{x}_1 \cdots \mathbf{x}_N\}$ and indicator variables $\mathbf{Z} = \{\mathbf{z}_1 \cdots \mathbf{z}_N\}$ where each $\mathbf{z}_n = \{z_{1n} \cdots z_{Mn}\}$

The EM Algorithm

UNIVERSITY
of
GLASGOW

- The problem is that we assumed knowledge of the values for the allocation or indicator variables z_{mn}
- Need the joint likelihood of data $\mathbf{X} = \{\mathbf{x}_1 \cdots \mathbf{x}_N\}$ and indicator variables $\mathbf{Z} = \{\mathbf{z}_1 \cdots \mathbf{z}_N\}$ where each $\mathbf{z}_n = \{z_{1n} \cdots z_{Mn}\}$
- Given $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1 \cdots \boldsymbol{\theta}_M\}$ we can marginalise over all possible component allocations

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

where the summation is over all possible values which $\mathbf{Z}$ may take on.

The EM Algorithm

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

The EM Algorithm

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

Use inequality $\log E\{f(X)\} \geq E\{\log f(X)\}$ so can write

The EM Algorithm

UNIVERSITY
of
GLASGOW

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

Use inequality $\log E\{f(X)\} \geq E\{\log f(X)\}$ so can write

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

The EM Algorithm

As $\mathbf{x}_n$ drawn iid from m exclusively then summation over all $\mathbf{Z}$ equals a summation over all n and m i.e. $\mathcal{L}_B$ equals

$$\sum_{\mathbf{Z}} P(\mathbf{Z}|\mathbf{X}) \log \frac{p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})}{P(\mathbf{Z}|\mathbf{X})} = \sum_{m,n}^{M,N} P(m|\mathbf{x}_n) \log \frac{p(\mathbf{x}_n|\boldsymbol{\theta}_m)P(m}{P(m|\mathbf{x}_n)}$$

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

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

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

The EM Algorithm

- The Expectation Maximisation (EM) algorithm is a general purpose method to *Maximise* the likelihood of the complete data $(\mathbf{X}\ \&\ \mathbf{Z})$ so as to obtain estimates of the component parameters $\boldsymbol{\theta}_m$.

The EM Algorithm

UNIVERSITY
of
GLASGOW

- The Expectation Maximisation (EM) algorithm is a general purpose method to *Maximise* the likelihood of the complete data $(\mathbf{X}$ & $\mathbf{Z})$ so as to obtain estimates of the component parameters $\boldsymbol{\theta}_m$.
- Before performing the *Maximisiation* step we require to obtain the *Expected* values of a set of hidden binary allocation variables z_{mn}.

The EM Algorithm

- The Expectation Maximisation (EM) algorithm is a general purpose method to *Maximise* the likelihood of the complete data $(\mathbf{X} \,\&\, \mathbf{Z})$ so as to obtain estimates of the component parameters $\boldsymbol{\theta}_m$.
- Before performing the *Maximisiation* step we require to obtain the *Expected* values of a set of hidden binary allocation variables z_{mn}.
- Once we have obtained the *Expected* values of the latent variables we then perform the *Maximisation* step to obtain our current parameter estimates.

The EM Algorithm

UNIVERSITY
of
GLASGOW

- The Expectation Maximisation (EM) algorithm is a general purpose method to *Maximise* the likelihood of the complete data $(\mathbf{X}\ \&\ \mathbf{Z})$ so as to obtain estimates of the component parameters $\boldsymbol{\theta}_m$.
- Before performing the *Maximisiation* step we require to obtain the *Expected* values of a set of hidden binary allocation variables z_{mn}.
- Once we have obtained the *Expected* values of the latent variables we then perform the *Maximisation* step to obtain our current parameter estimates.
- This EM interleaving is continued until some convergence criterion is achieved.

Expectation Step

- Taking functional derivatives of the lower-bound with respect to $P(m|\mathbf{x}_n)$ then

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

Expectation Step

UNIVERSITY
of
GLASGOW

- Taking functional derivatives of the lower-bound with respect to $P(m|\mathbf{x}_n)$ then

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

- Setting to zero we see that $P(m|\mathbf{x}_n) \propto p(\mathbf{x}_n|\boldsymbol{\theta}_m)P(m)$ and normalising appropriately yields the distribution of the form

$$P(m|\mathbf{x}_n) = \frac{p(\mathbf{x}_n|\boldsymbol{\theta}_m)P(m)}{\sum_{m'=1}^{M} p(\mathbf{x}_n|\boldsymbol{\theta}_{m'})P(m')}$$

Expectation Step

- You should now be able to see that this is the posterior distribution over the mixture components m which generated $\mathbf{x}_n$, or the expected value of the binary variable z_{mn}.

Expectation Step

- You should now be able to see that this is the posterior distribution over the mixture components m which generated $\mathbf{x}_n$, or the expected value of the binary variable z_{mn}.
- Now that we have maximised the bound with respect to the *Expected* value of the indicator variable we need to *Maximise* the bound with respect to the parameter values.

Maximisation Step

- The only terms in the bound $\mathcal{L}_B$ which are dependent on the component parameters are

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

  in which case we maximise the above with respect to each $\boldsymbol{\theta}_m$.

Maximisation Step

- As an example assume that each $p(\mathbf{x}_n|\boldsymbol{\theta}_m)$ is a multivariate Gaussian, then expanding and retaining the elements dependent on the parameters we obtain

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

Maximisation Step

UNIVERSITY *of* GLASGOW

- Taking derivatives wrt $\boldsymbol{\mu}_m$ and solving yields

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

Maximisation Step

- Taking derivatives wrt $\boldsymbol{\mu}_m$ and solving yields

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

- Nice result, compare with the estimator when we have perfect knowledge of the allocation variables z_{mn} i.e.

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

  so in the absence of the values z_{mn} we employ the expected values, or the posterior probabilities $P(m|\mathbf{x}_n)$ which are obtained in the *Expectation* step

Maximisation Step

- Leaving you to have some fun with the derivation of the estimator for the covariance matrices we obtain

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

  again we can see that we have replaced perfect knowledge of the allocation variables with our current estimates of the posteriors $P(m|\mathbf{x}_n)$

Maximisation Step

UNIVERSITY *of* GLASGOW

- Finally we need an estimate for $P(m)$ taking derivatives then we observe that

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

This needs to be properly normalised and so

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

EM Algorithm

E Step

$$P(m|\mathbf{x}_n) = \frac{p(\mathbf{x}_n|\boldsymbol{\theta}_m)P(m)}{\sum_{m'=1}^{M} p(\mathbf{x}_n|\boldsymbol{\theta}_{m'})P(m')}$$

M Step

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

Experiments

- Data with equal probability from three 2D Gaussians with a common unit variance i.e. $\mathbf{I}$ means of $[0, 0], [3, 3], [-3, 3]$ see Matlab file `Gauss_Mix_Data.mat`

Experiments

- Data with equal probability from three 2D Gaussians with a common unit variance i.e. $\mathbf{I}$ means of $[0,0],[3,3],[-3,3]$ see Matlab file `Gauss_Mix_Data.mat`
- There is also a 1500×2 dimensional data set drawn from the same distribution which can be used to obtain values of likelihood on independent test data

Experiments

- Data with equal probability from three 2D Gaussians with a common unit variance i.e. $\mathbf{I}$ means of $[0, 0], [3, 3], [-3, 3]$ see Matlab file `Gauss_Mix_Data.mat`
- There is also a 1500×2 dimensional data set drawn from the same distribution which can be used to obtain values of likelihood on independent test data
- Wish to estimate probability density for data. There is, as always, one slight snag, our EM algorithm requires the number of components in the mixture. For now lets assume that we have a good idea what this value is

Experiments

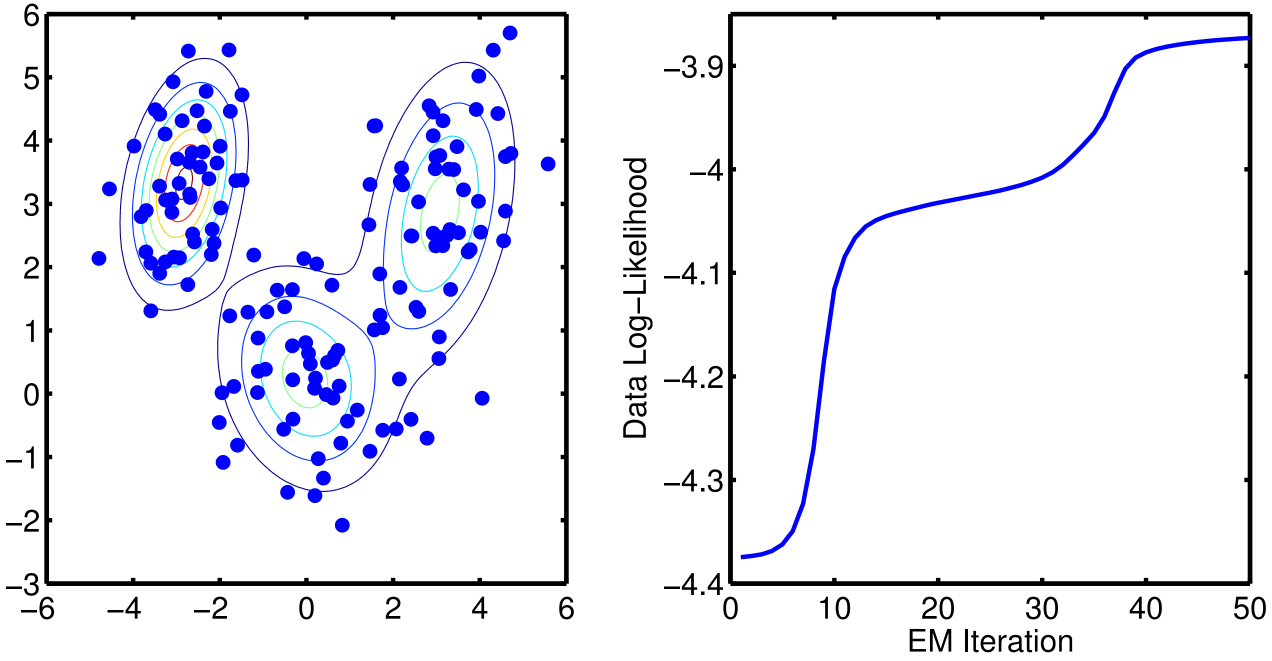

Figure 3: The left plot shows a random sample of 150 points drawn from 3 distinct 2D Gaussians, the iso-contours of estimated probability density under the assumption that the density is a mixture of three 2D Gaussians, are superimposed on the plot. The right plot shows the data likelihood under the mixture model at each EM step, it is clear that the likelihood does not decrease at each step.

Experiments

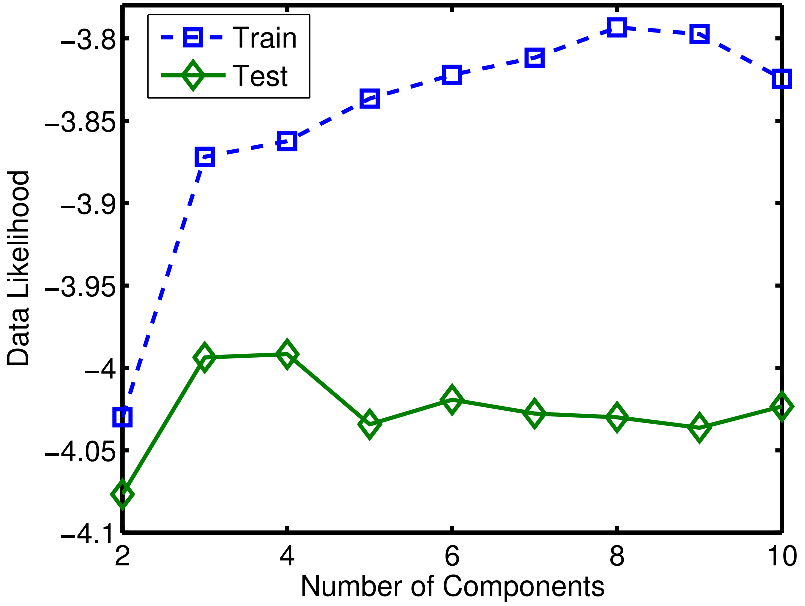

Figure 4: The left plot shows a random sample of 150 points drawn from 3 distinct 2D Gaussians, the iso-contours of estimated probability density under the assumption that the density is a mixture of three 2D Gaussians, are superimposed on the plot. The right plot shows the data likelihood under the mixture model at each EM step, it is clear that the likelihood does not decrease at each step.