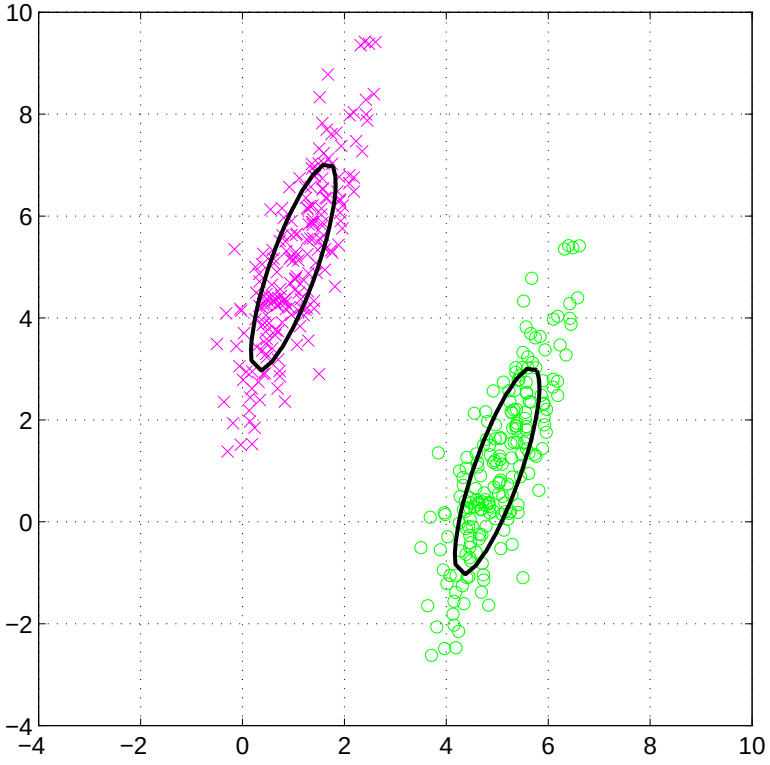
University of Cambridge Engineering Part IIB

Module 4F10: Statistical Pattern Processing

Handout 2: Bayes' Classifier with Gaussians



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

Bayes' Decision Rule with Gaussian densities

Bayes' minimum error decision rule can be written in terms of Discriminant Functions for each class i as a function of the features so that the decision rule is

$$\text{choose } \omega_j \text{ where } g_j(\boldsymbol{x}) = \max_i g_i(\boldsymbol{x})$$

where

$$g_i(\boldsymbol{x}) = \ln p(\boldsymbol{x}|\omega_i) + \ln P(\omega_i)$$

Note that taking the log does not affect the decision boundaries between the classes since log is a monotonic function.

We will use discriminant functions to investigate the Gaussian classifier. The class-conditional pdf's of the classes are Gaussian which we will abbreviate as $p(\boldsymbol{x}|\omega_i) = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. Substituting the full multivariate Gaussian formula, noting there is no need to include a constant term $\frac{d}{2}\log(2\pi)$ for the discriminant functions, in general we have

$$\begin{aligned} g_i(\boldsymbol{x}) \;=\; & -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)'\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i) \\ & -\frac{1}{2}\ln(|\boldsymbol{\Sigma}_i|) + \ln P(\omega_i) \end{aligned}$$

For the two class case, consider a single discriminant function $g_1(\boldsymbol{x}) - g_2(\boldsymbol{x})$ and the value compared to a threshold for decision:

$$\begin{aligned} -(\boldsymbol{x}-\boldsymbol{\mu}_1)'\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) + (\boldsymbol{x}-\boldsymbol{\mu}_2)'\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2) & \\ + \ln\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|} + 2\ln\frac{P_1}{P_2} \;\begin{matrix}\omega_1\\ \gtrless \\ \omega_2\end{matrix}\; 0 & \end{aligned}$$

This decision rule specifies a quadratic classifier and the decision boundaries are quadratic functions of the input features. Note that this point is also when the class posteriors are both equal to $1/2$. The decision boundaries occur when the left hand side of the above equation is equal to zero.

i.e.

$$(\boldsymbol{x}-\boldsymbol{\mu}_1)^{'}\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)-(\boldsymbol{x}-\boldsymbol{\mu}_2)^{'}\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2) - \ln\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|}-2\ln\frac{P_1}{P_2}=0$$

$$\boldsymbol{x}^{'}(\boldsymbol{\Sigma}_1^{-1}-\boldsymbol{\Sigma}_2^{-1})\boldsymbol{x}+2(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu_2}-\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu_1})^{'}\boldsymbol{x} +\boldsymbol{\mu}_1^{'}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1-\boldsymbol{\mu}_2^{'}\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2-\ln\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|}-2\ln\frac{P_1}{P_2}=0$$

i.e. of the form

$$\boldsymbol{x}^{'}\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}^{'}\boldsymbol{x}+c=0$$

which gives the equation of the decision surface.

Special Case: $\boldsymbol{\Sigma}_i = \sigma^2 \boldsymbol{I}$

For this case the covariance matrices are common, diagonal with all dimensions having equal variance, and so again dropping terms independent of class and using the expression for a Gaussian with d uncorrelated features leads to

$$\begin{aligned} g_i(\boldsymbol{x}) &= \ln \prod_{j=1}^{d} \exp(-\frac{(x_i - \mu_{ij})^2}{2\sigma^2}) + \ln P(\omega_i) \\ g_i(\boldsymbol{x}) &= -\sum_{j=1}^{d} \frac{(x_j - \mu_{ij})^2}{2\sigma^2} + \ln P(\omega_i) \end{aligned}$$

or alternatively

$$g_i(\boldsymbol{x}) = -\frac{||\boldsymbol{x} - \boldsymbol{\mu_i}||^2}{2\sigma^2} + \ln P(\omega_i)$$

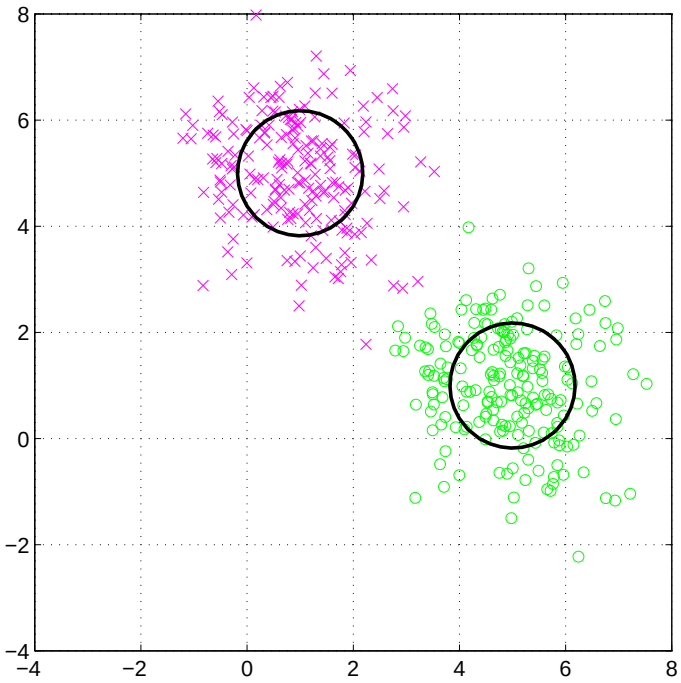
where $||\boldsymbol{y}||$ denotes the Euclidean Norm i.e.

$$||\boldsymbol{x} - \boldsymbol{\mu_i}||^2 = (\boldsymbol{x} - \boldsymbol{\mu_i})'(\boldsymbol{x} - \boldsymbol{\mu_i})$$

Note that the discriminants can also be simply derived from the general Gaussian case by noting that here

$$\boldsymbol{\Sigma}^{-1} = \frac{1}{\sigma^2}\boldsymbol{I}$$

and the class-conditional pdfs are circular.



- The quadratic term in the general Gaussian classifier decision boundary disappears since the covariance matrices are equal for all classes.
- The decision boundary is linear and orthogonal to the line joining the means *i.e.* this is a linear classifier, and the boundary is a hyperplane in $d-1$ dimensions.
- In the case of equal class priors *i.e.* $P(\omega_1) = P(\omega_2)$, the decision boundary passes half-way between the means. The form of the boundary can be simply found from the general case as:

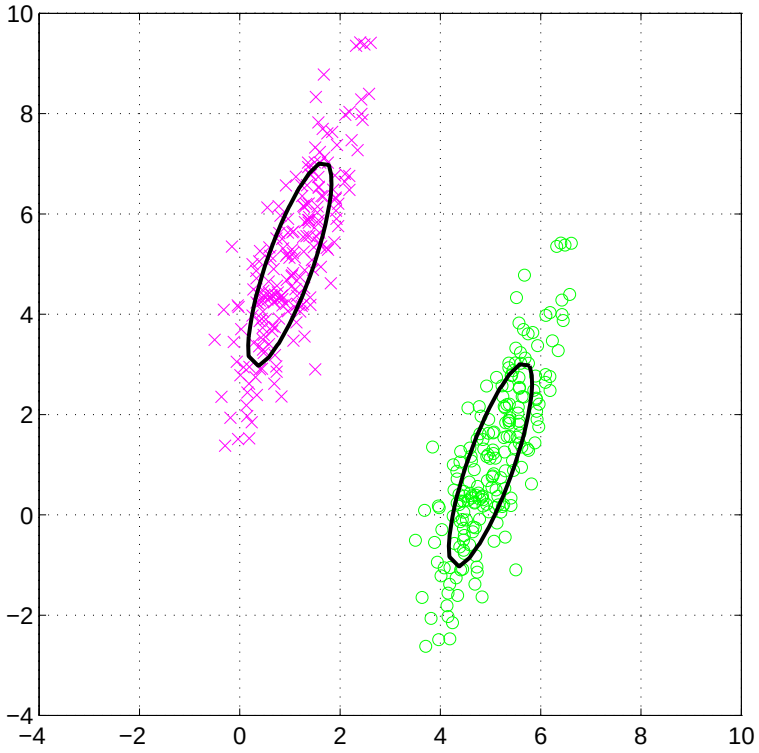
$$(\boldsymbol{\mu_2} - \boldsymbol{\mu_1})' \left[\boldsymbol{x} - 1/2\,(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)\right] = 0$$

- In the case of equal priors then the classifier is equivalent to computing the minimum Euclidean distance from the class mean.

Special Case: $\Sigma_i = \Sigma$

Here the covariance matrices are common but full.

$$g_i(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{\mu}_i)' \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_i) + 2 \ln P(\omega_i)$$

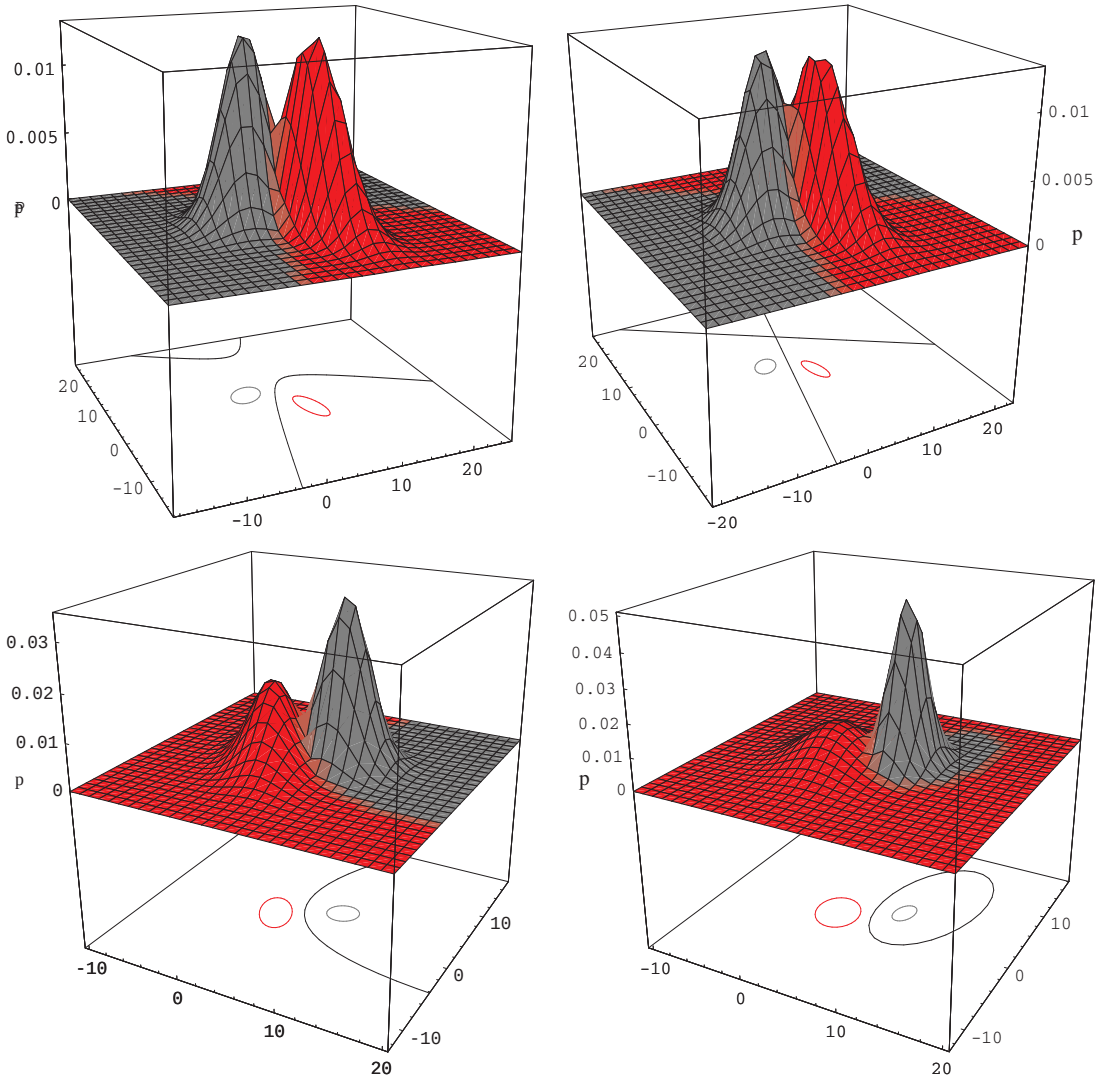


- Here the classifier computes a weighted distance called the Mahalanobis distance from the input data $\boldsymbol{x}$ to the mean.
- The squared Mahalanobis distance $(\boldsymbol{x} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})$ both weights the effect of individual features (by their inverse variance) and accounts for inter-feature correlations.

- For equal class priors this is a "nearest-the-mean" classifier with distance calculated using the Mahalanobis distance.
- Common covariance matrices, and therefore linear decision boundaries.
- In the case of common covariances a classifier can be more simply constructed by transforming the input space so that the features are decorrelated. This means that a full covariance calculation need not be performed for every class and has the advantage of reducing computation also allows the classes to be represented by Gaussians with diagonal covariance matrices.

Examples of General Case

Arbitrary Gaussian distributions can lead to general hyper-quadratic boundaries. The following figures (from DHS) indicate this. Note that the boundaries can of course be straight lines and the regions may not be simply connected.



Example Decision Boundary

Assume two classes with

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 3 \\ 6 \end{bmatrix}; \; \boldsymbol{\Sigma}_1 = \begin{bmatrix} 1/2 & 0 \\ 0 & 2 \end{bmatrix} \quad \boldsymbol{\mu}_2 = \begin{bmatrix} 3 \\ -2 \end{bmatrix} \; \boldsymbol{\Sigma}_2 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

The inverse covariance matrices are then

$$\boldsymbol{\Sigma}_1{}^{-1} = \begin{bmatrix} 2 & 0 \\ 0 & 1/2 \end{bmatrix} \quad \boldsymbol{\Sigma}_2{}^{-1} = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix}$$

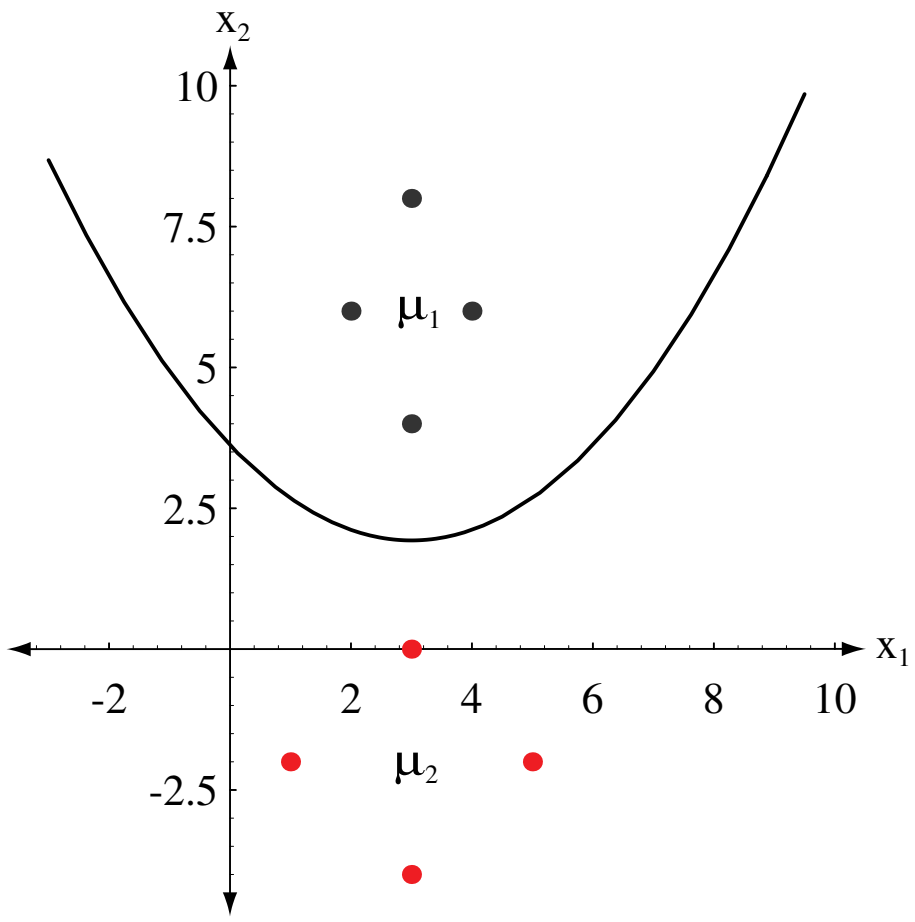
Substituting into the general expression for Gaussian boundaries yields:

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 3/2 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$
$$+2 \begin{bmatrix} -9/2 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + 36 - 6.5 - \ln 4 = 0$$

$$1.5x_1^2 - 9x_1 - 8x_2 + 28.11 = 0$$

$$x_2 = 3.514 - 1.125x_1 + 0.1875x_1^2$$

which is a parabola with a minimum at (3,1.83). This is illustrated (from DHS) below. The graph shows 4 sample points from each class, the means and the decision boundary. Note that the boundary does not pass through the mid-point between the means.


x2
10
7.5
5
2.5
-2.5
μ1
μ2
x1
-2
2
4
6
8
10

Maximum Likelihood Estimation

We need to estimate the vector parameters of the class conditional pdfs $\boldsymbol{\theta}$ from training data. The underlying assumption for ML estimates is that the parameter values are fixed but unknown. Assume that the parameters are to be estimated from a training/design data set, $\mathcal{D}$, with n example patterns

$$\mathcal{D} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$$

and note $\boldsymbol{\theta}$ depends on $\mathcal{D}$.

If these training vectors are drawn independently *i.e.* are independent and indentically distributed or IID, the joint probability density of the training set is given by

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{k=1}^{n} p(\boldsymbol{x}_k|\boldsymbol{\theta})$$

$p(\mathcal{D}|\boldsymbol{\theta})$ viewed as a function of $\boldsymbol{\theta}$ is called the *likelihood* of $\boldsymbol{\theta}$ given $\mathcal{D}$.

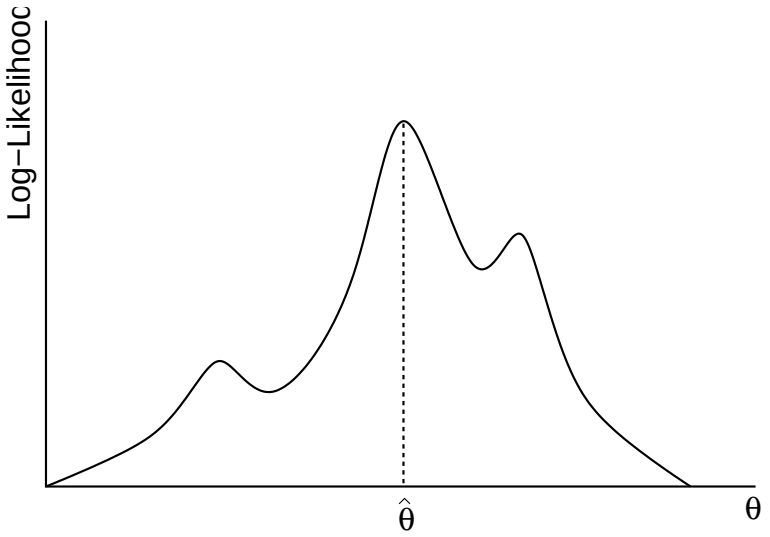
In ML estimation, the value of $\boldsymbol{\theta}$ is chosen which is most likely to give rise to the observed training data.

Often the log likelihood function, $l(\boldsymbol{\theta})$, is maximised instead for convenience *i.e.*

$$l(\boldsymbol{\theta}) = \ln p(\mathcal{D}|\boldsymbol{\theta}) = \sum_{k=1}^{n} \ln p(\boldsymbol{x}_k|\boldsymbol{\theta})$$

This value can either be maximised by iterative techniques (*e.g.* gradient descent and expectation-maximisation algorithms : see later in the course) or in some cases by a direct closed form solution exists. Either way we need to differentiate the log likelihood function with respect to the unknown parameters and equate to zero.

Maximising the Likelihood



To find the maximum likelihood value we need to find the point where

$$\nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta}) = \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} \log(p(\boldsymbol{x}_i|\boldsymbol{\theta})) = \mathbf{0}$$

i.e. the gradient with respect to $\boldsymbol{\theta}$ is zero. Note

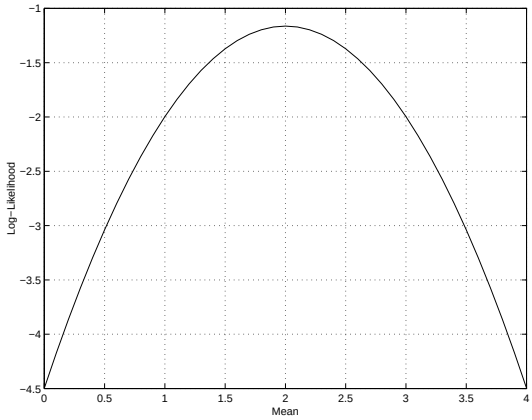
$$\nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial l(\boldsymbol{\theta})}{\partial \theta_1} \\ \vdots \\ \frac{\partial l(\boldsymbol{\theta})}{\partial \theta_P} \end{bmatrix}$$

if there are P model parameters to estimate. The gradient will be zero at *maxima* (desired), *minima* and *saddle-points*.

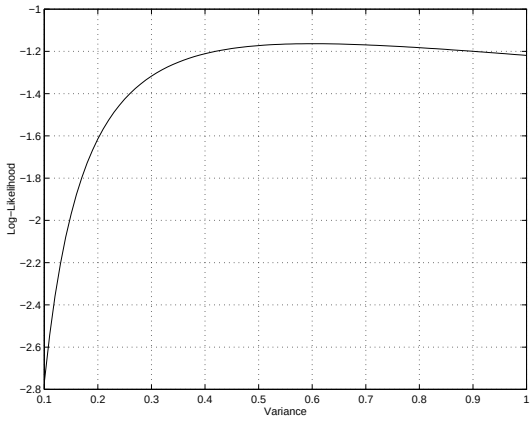
The dependence of the *grad* on the model parameters, $\boldsymbol{\theta}$, will be assumed. It will now be simply written as ∇ for clarity.

Log-Likelihood Functions

As an example consider estimating the parameters of a univariate Gaussian distribution with data generated from a Gaussian distribution with mean=2.0 and variance=0.6.



The variation of log-likelihood with the mean is shown above (assuming that the correct variance is known).



Similarly the variation with the variance (assuming that the correct mean is known).

Mean of a Gaussian distribution

Now we would like to obtain an analytical expression for the estimate of the mean of a Gaussian distribution. Consider a single dimensional observation ($d = 1$). Consider estimating the mean, so

$$\theta = \mu$$

First the log-likelihood may be written as

$$l(\mu) = \sum_{i=1}^{n} \log(p(x_i|\mu)) = \sum_{i=1}^{n} \left(-\frac{1}{2}\log(2\pi\sigma^2) - \frac{(x_i - \mu)^2}{2\sigma^2} \right)$$

Differentiating this gives

$$\nabla l(\mu) = \frac{\partial}{\partial \mu} l(\mu) = \sum_{i=1}^{n} \frac{(x_i - \mu)}{\sigma^2}$$

We now want to find the value of the model parameters that the gradient is 0. Thus

$$\sum_{i=1}^{n} \frac{(x_i - \mu)}{\sigma^2} = 0$$

So (much as expected!) the ML estimate of the mean $\hat{\mu}$ is

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Similarly the ML estimate of the variance can be derived.

Multivariate Gaussian Case

For the general case the set of model parameters associated with a Gaussian distribution are

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\mu} \\ \text{vec}(\boldsymbol{\Sigma}) \end{bmatrix}$$

We will not go into the details of the derivation here (do this as an exercise), but it can be shown that the ML solutions for the mean ($\hat{\boldsymbol{\mu}}$) and the covariance matrix ($\hat{\boldsymbol{\Sigma}}$) are

$$\hat{\boldsymbol{\mu}} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i$$

and

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})'$$

Note that when deriving ML estimates for multivariate distributions, the following matrix calculus equalities are useful:

$$\begin{aligned} \frac{\partial}{\partial \mathbf{A}} (\mathbf{b}'\mathbf{A}\mathbf{c}) &= \mathbf{b}\mathbf{c}' \\ \frac{\partial}{\partial \mathbf{a}} (\mathbf{a}'\mathbf{B}\mathbf{a}) &= 2\mathbf{B}\mathbf{a} \\ \frac{\partial}{\partial \mathbf{a}} (\mathbf{a}'\mathbf{B}\mathbf{c}) &= \mathbf{B}\mathbf{c} \\ \frac{\partial}{\partial \mathbf{A}} (\log(|\mathbf{A}|)) &= \mathbf{A}^{-1} \end{aligned}$$

Biased Estimators

You will previously have found that the unbiased estimate of the covariance matrix, $\hat{\boldsymbol{\Sigma}}$, with an unknown value of the mean is

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})'$$

There is a difference between this and the ML solution ($\frac{1}{n}$ and $\frac{1}{n-1}$). In the limit as $n \rightarrow \infty$ the two values are the same. So which is correct/wrong? Neither - they're just different.

There are two important statistical properties illustrated here.

1. **Unbiased** estimators: the expected value over a large number of estimates of the parameters is the "true" parameter.
2. **Consistent** estimators: in the limit as the number of points tends to infinity the estimate is the "true" estimate.

It can be shown that the ML estimate of the mean is unbiased, the variance is only consistent.

Cost of Mis-Classification

We have assumed that the goal is to minimise the average probability of classification error. Recall that for the two-class problem, the Bayes minimum average error decision rule can be written as:

$$\frac{P(\omega_1 \mid \boldsymbol{x})}{P(\omega_2 \mid \boldsymbol{x})} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} 1$$

or using the likelihood ratio:

$$\frac{p(x|\omega_1)}{p(x|\omega_2)} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{P(\omega_2)}{P(\omega_1)}$$

Sometimes, the cost (or loss) for misclassification is specified (or can be estimated) and different types of classification error may not have equal cost.

C_{12} Cost of choosing $\omega_1|\boldsymbol{x}$ from ω_2
C_{21} Cost of choosing $\omega_2|\boldsymbol{x}$ from ω_1

and C_{ii} is the cost of correct classification.

The aim now is to minimise the Bayes' Risk which is the expected value of the classification cost.

Let the decision region associated with class ω_j be denoted Ω_j. Consider all the patterns that belong to class ω_1. The expected cost (or risk) for these patterns $\mathcal{R}_1$ is given by

$$\mathcal{R}_1 = \sum_{i=1}^{2} C_{i1} \int_{\Omega_i} p(\boldsymbol{x}|\omega_1) d\boldsymbol{x}$$

The overall cost $\mathcal{R}$ is found as

$$\begin{aligned}\mathcal{R} &= \sum_{j=1}^{2} \mathcal{R}_j P(\omega_j) \\ &= \sum_{j=1}^{2} \sum_{i=1}^{2} C_{ij} \int_{\Omega_i} p(\boldsymbol{x}|\omega_i) d\boldsymbol{x} P(\omega_j) \\ &= \sum_{i=1}^{2} \int_{\Omega_i} \sum_{j=1}^{2} C_{ij} p(\boldsymbol{x}|\omega_j) P(\omega_j) d\boldsymbol{x}\end{aligned}$$

Minimise integrand at all points, choose Ω_1 so

$$\sum_{j=1}^{2} C_{1j} p(\boldsymbol{x}|\omega_j) P(\omega_j) < \sum_{j=1}^{2} C_{2j} p(\boldsymbol{x}|\omega_j) P(\omega_j)$$

In the case that $C_{11} = C_{22} = 0$ we obtain

$$\frac{C_{21} P(\omega_1 \mid \boldsymbol{x})}{C_{12} P(\omega_2 \mid \boldsymbol{x})} \begin{matrix} \omega_1 \\ > \\ < \\ \omega_2 \end{matrix} 1$$

or using the likelihood ratio

$$\frac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)} \begin{matrix} \omega_1 \\ > \\ < \\ \omega_2 \end{matrix} \frac{P(\omega_2) C_{12}}{P(\omega_1) C_{21}}$$

Note that decision rule to minimise the Bayes' Risk is the minimum error rule when $C_{12} = C_{21} = 1$ and correct classification has zero cost.

ROC curves

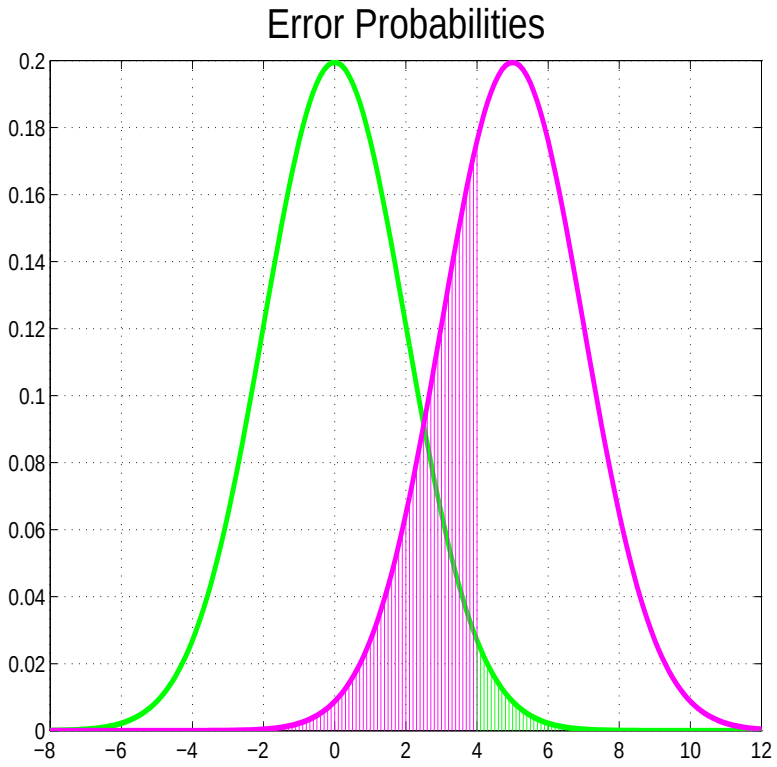
In some problems, such as in medical diagnostics, there is is a "target" class that you want to separate from the rest of the population (*i.e.* it is a detection problem). Four types of outcomes can be identified:

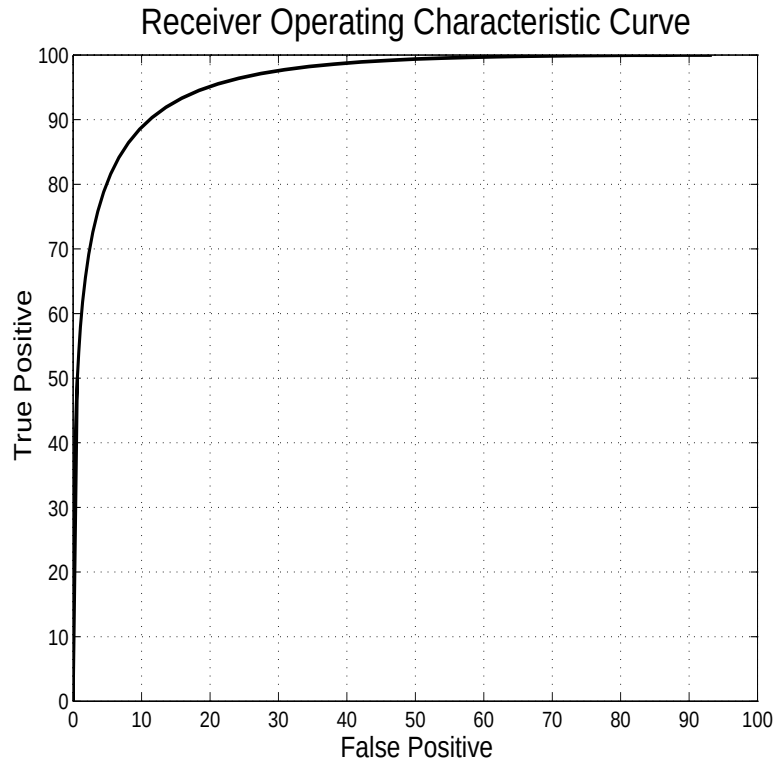
- True Positive (Hit)
- True Negative
- False Positive (False Alarm)
- False Negative

As the decision threshold is changed the ratio of True Positive to False Positive changes. This tradeoff is often plotted in a Receiver Operating Characteristic or ROC curve (originally applied to problems such as radar signal detection).

Changes to the threshold from the point for minimum classification error may improve detection hits significantly (at a cost of more false alarms).

- The example below shows a 1-d example with classes with equal variances and equal priors: the threshold for minimum error would be $(\mu_1 + \mu_2)/2$.


Error Probabilities
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
−8
−6
−4
−2
0
2
4
6
8
10
12


Receiver Operating Characteristic Curve
True Positive
False Positive
100
90
80
70
60
50
40
30
20
10
0
0
10
20
30
40
50
60
70
80
90
100