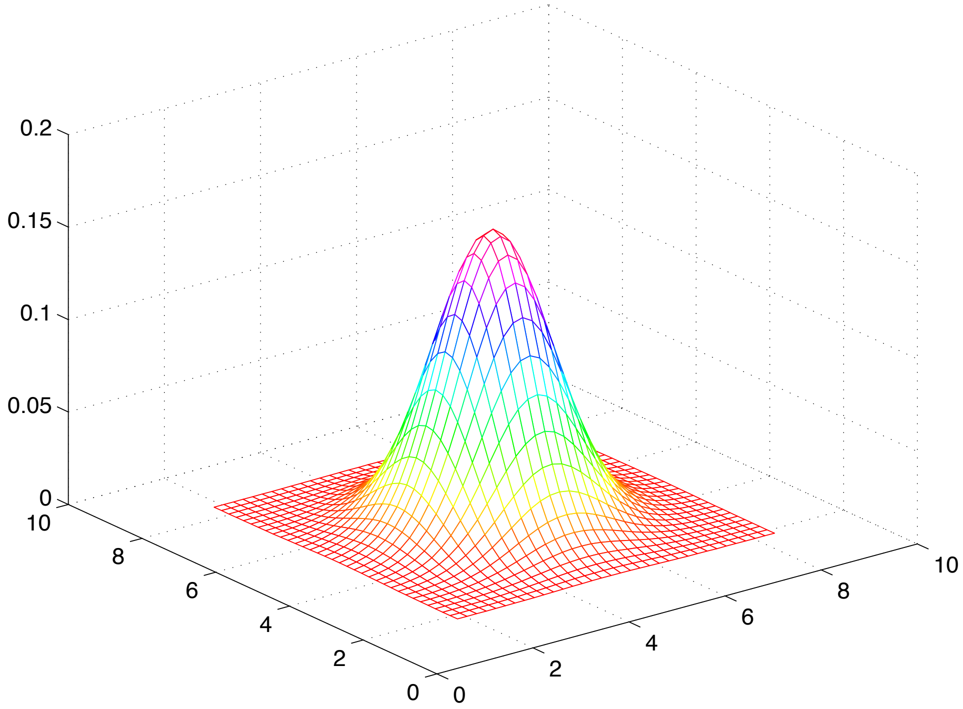
University of Cambridge
Engineering Part IIB

Module 4F10: Statistical Pattern Processing

Handout 2: Multivariate Gaussians



Mark Gales
mjfg@eng.cam.ac.uk

Michaelmas 2015

Generative Model Decision Boundaries

The previous lecture discussed Bayes' decision rule and how it may be used with generative models to yield a classifier and decision boundaries. In generative models the joint distribution is modelled as

$$p(\boldsymbol{x}, \omega) = p(\boldsymbol{x}|\omega)P(\omega)$$

The decision boundary will depend on

- $p(\boldsymbol{x}|\omega)$: the class-conditional PDF
- $P(\omega)$: the prior distribution

(continuous observation feature vectors, $\boldsymbol{x}$, are considered)

A large of number of class-conditional PDFs could be used

- univariate Gaussian: parameters μ, σ^2
$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$
- uniform: parameters a, b
$$p(x) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & \text{otherwise} \end{cases}$$
- etc etc

This lecture will look at the multivariate Gaussian:

- $p(\boldsymbol{x}|\omega_i) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$
- nature of the distribution and decision boundaries
- estimating the model parameters

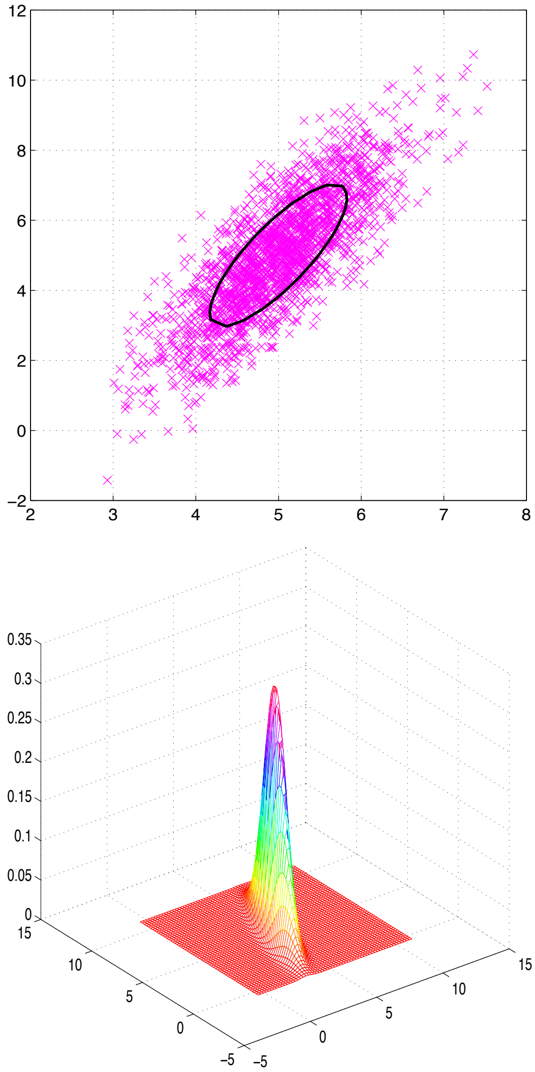
Multivariate Gaussian Distribution

$$p\left(\boldsymbol{x}\right)=\frac{1}{\left(2\pi\right)^{d/2}\left|\boldsymbol{\Sigma}\right|^{1/2}}\exp\left(-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)'\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)\right)$$

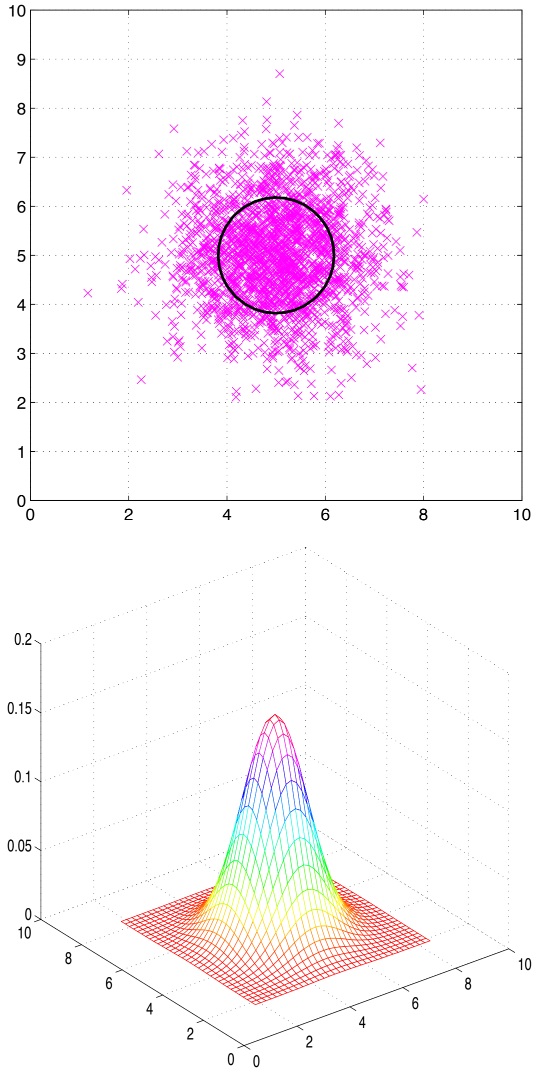
The distribution is characterised by:

- the mean vector $\boldsymbol{\mu}$
- the covariance matrix $\boldsymbol{\Sigma}$

$$\boldsymbol{\Sigma}=\begin{pmatrix}3 & 1\\ 1 & 0.5\end{pmatrix}$$



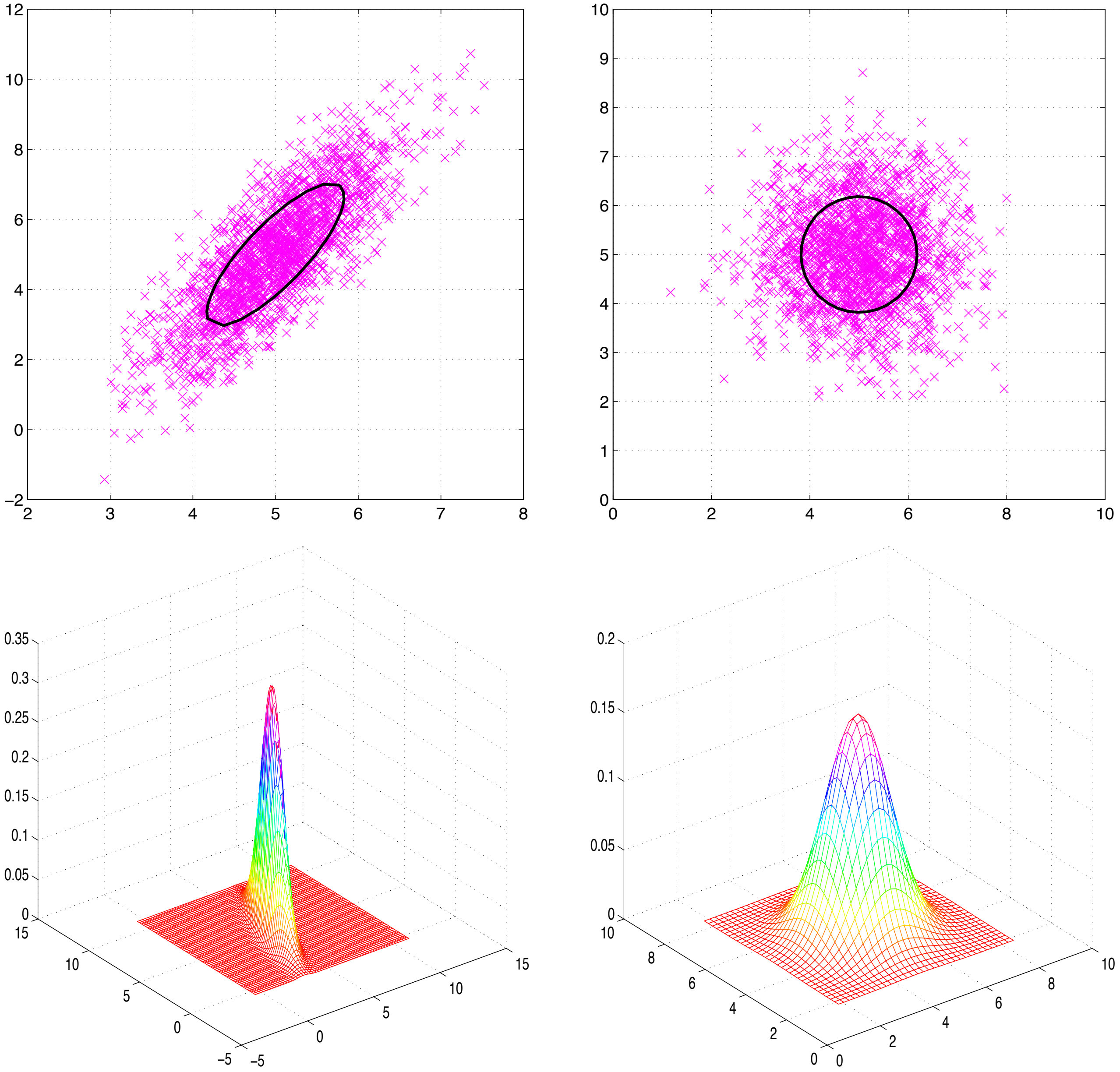
$$\boldsymbol{\Sigma}=\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$$



Multivariate Gaussian Distribution

$$p\left(\boldsymbol{x}\right) = \frac{1}{\left(2\pi\right)^{d/2}\left|\boldsymbol{\Sigma}\right|^{1/2}}\exp\left(-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)'\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)\right)$$

- $\boldsymbol{\mu}$: mean vector
- $\boldsymbol{\Sigma}$: covariance matrix


12
10
8
6
4
2
0
−2
2
3
4
5
6
7
8
10
9
8
7
6
5
4
3
2
1
0
0
2
4
6
8
10
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
15
10
5
0
−5
−5
0
5
10
15
0.2
0.15
0.1
0.05
0
10
8
6
4
2
0
0
2
4
6
8
10

Properties

- The mean and covariance matrix are defined as

$$\begin{aligned}\boldsymbol{\mu} &= \mathcal{E}\left\{\boldsymbol{x}\right\} \\ \boldsymbol{\Sigma} &= \mathcal{E}\left\{(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})'\right\}\end{aligned}$$

  The matrix is clearly symmetric and for d dimensions is described by $d(d+1)/2$ parameters.

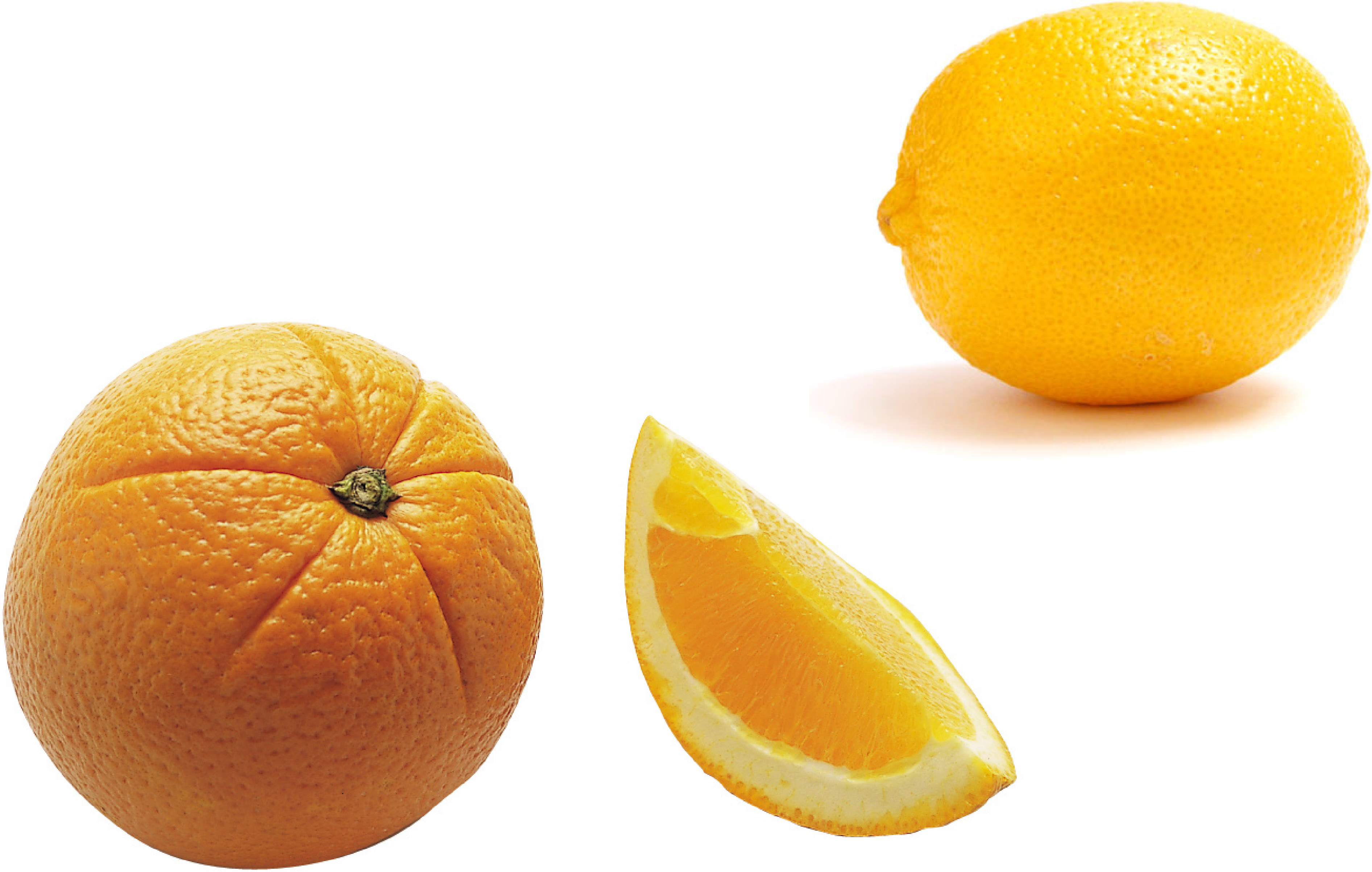
- The diagonal elements of the covariance matrix σ_{ii} are the variances in the individual dimensions σ_i^2, the off-diagonal elements determine the correlation.
- If all off-diagonal elements are zero, the covariance matrix is uncorrelated, this is equivalent to a univariate Gaussian in each dimension

$$p(\boldsymbol{x}) = \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right)$$

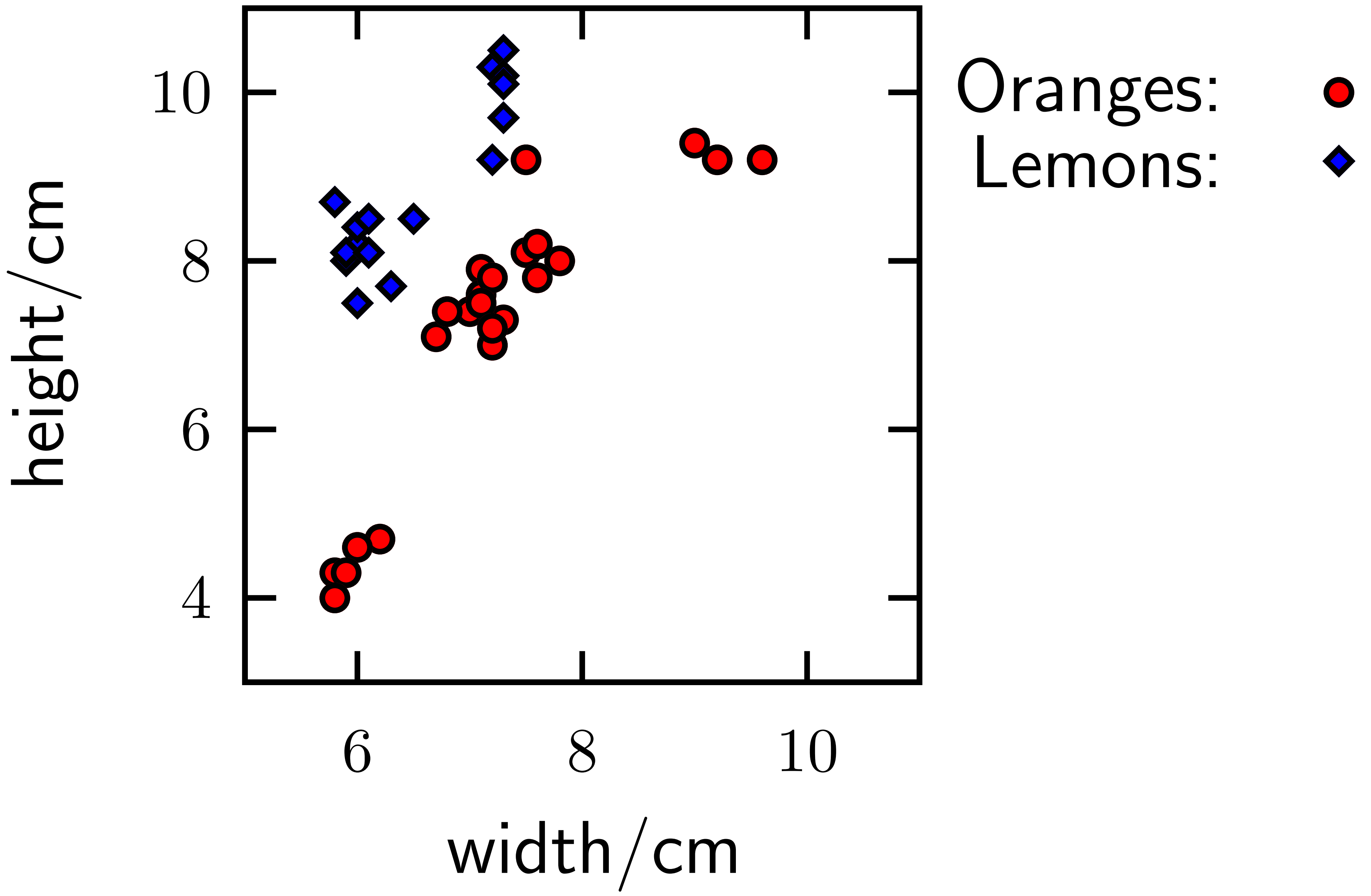
- For a full covariance matrix correlations cause the contours of equal probability density, which are ellipses, to be angled to the axes of the feature space .
- An important property that we will return to is the effect of a linear transformation on a Gaussian distribution. Given that the distribution of vectors $\boldsymbol{x}$ is Gaussian and that $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$ (and $\boldsymbol{A}$ is non-singular) then

$$\begin{aligned}\boldsymbol{\mu}_{\mathrm{y}} &= \boldsymbol{A}\boldsymbol{\mu}_{\mathrm{x}} + \boldsymbol{b} \\ \boldsymbol{\Sigma}_{\mathrm{y}} &= \boldsymbol{A}\boldsymbol{\Sigma}_{\mathrm{x}}\boldsymbol{A}'\end{aligned}$$

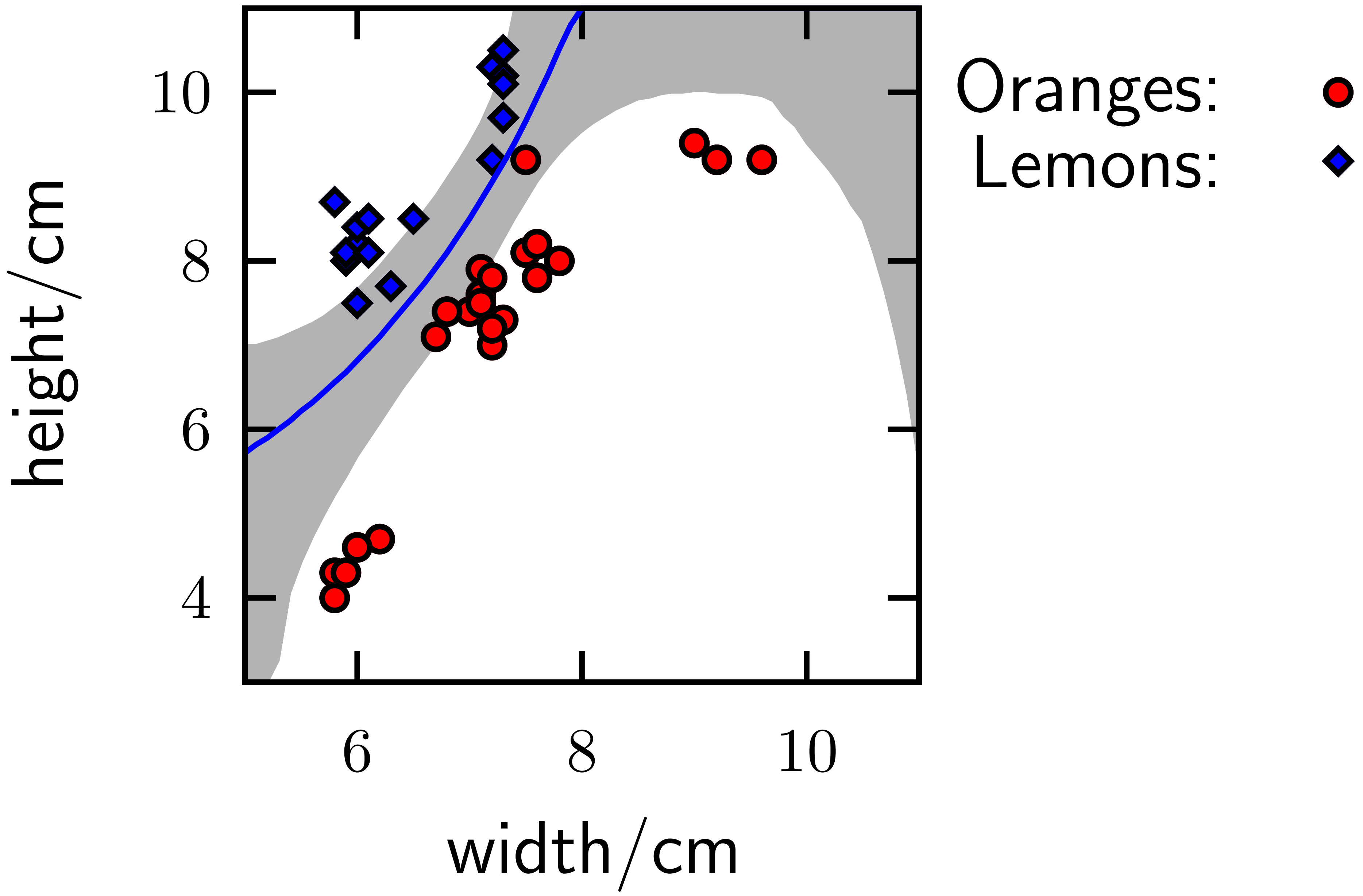
Oranges and Lemons
Thanks to Iain Murray



A two-dimensional space



Supervised learning



Binary Decision Boundary

$$\frac{P(\omega_1|\boldsymbol{x})}{P(\omega_2|\boldsymbol{x})} \underset{\omega_2}{\overset{\omega_1}{\substack{> \\ <}}} 1, \qquad \frac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)} \underset{\omega_2}{\overset{\omega_1}{\substack{> \\ <}}} \frac{P(\omega_2)}{P(\omega_1)}$$

$$\log(p(\boldsymbol{x}|\omega_1)) - \log(p(\boldsymbol{x}|\omega_2)) =$$
$$\log(P(\omega_2)) - \log(P(\omega_1))$$

Binary Decision Boundaries

For a two class problem, what is the form of the decision boundary when multivariate Gaussian distributions are used for the class-conditional PDFs?

Here the minimum probability of error decision boundary will be computed thus

$$\frac{P(\omega_1|\boldsymbol{x})}{P(\omega_2|\boldsymbol{x})} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} 1, \qquad \frac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{P(\omega_2)}{P(\omega_1)}$$

Normally logs are taken of both sides, so a point, $\boldsymbol{x}$, on the decision boundary satisfies

$$\log(p(\boldsymbol{x}|\omega_1)) - \log(p(\boldsymbol{x}|\omega_2)) = \log(P(\omega_2)) - \log(P(\omega_1))$$

Substituting in

$$p(\boldsymbol{x}|\omega_1) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \qquad p(\boldsymbol{x}|\omega_2) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$

Yields the following quadratic equation

$$-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)'\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)'\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2) + \frac{1}{2}\log\left(\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|}\right) = \log\left(\frac{P(\omega_2)}{P(\omega_1)}\right)$$

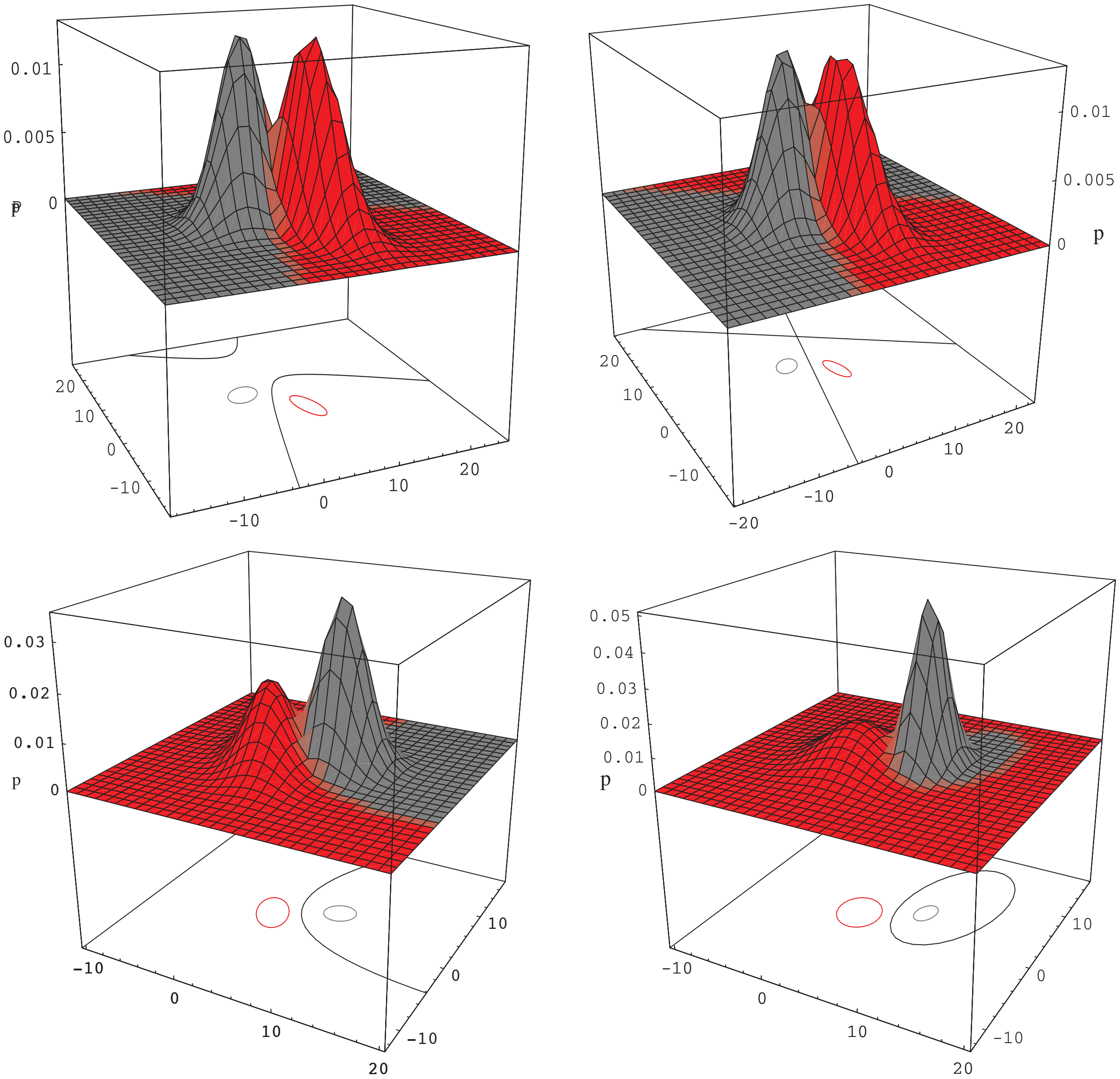
Which can be expressed as

$$\boldsymbol{x}'(\boldsymbol{\Sigma}_1^{-1} - \boldsymbol{\Sigma}_2^{-1})\boldsymbol{x} + 2(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu_2} - \boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu_1})'\boldsymbol{x} + \boldsymbol{\mu}_1'\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2'\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2 - \log\left(\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|}\right) - 2\log\left(\frac{P(\omega_1)}{P(\omega_2)}\right) = 0$$

i.e. of the form

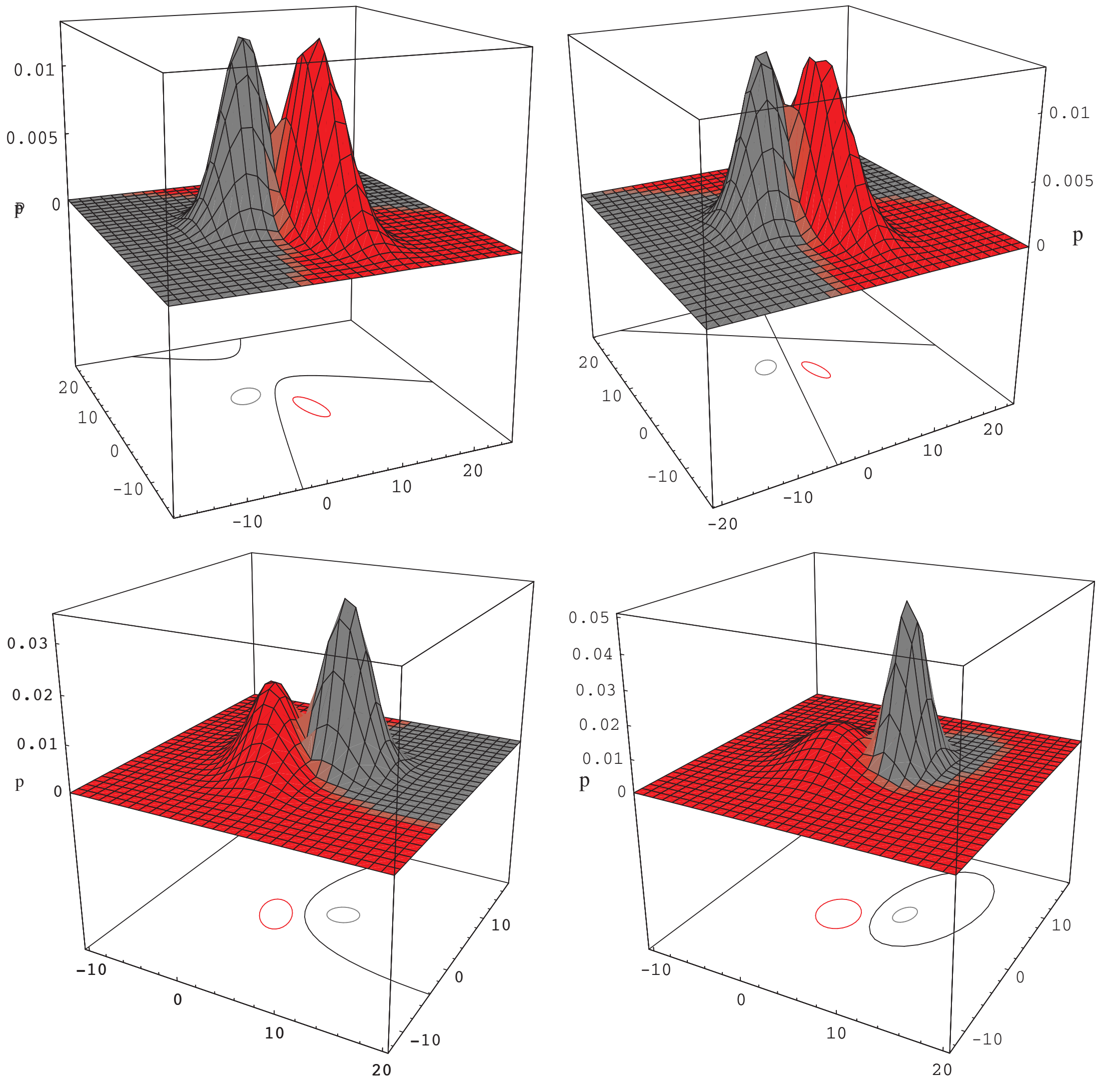
$$\boldsymbol{x}'\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}'\boldsymbol{x} + c = 0$$

which gives the equation of the decision boundary.


0.01
0.005
p
0
20
10
0
-10
-10
0
10
20
0.01
0.005
0
p
20
10
0
-10
-20
-10
0
10
20
0.03
0.02
0.01
p
0
-10
0
10
20
10
0
-10
0.05
0.04
0.03
0.02
0.01
p
0
-10
0
10
20
10
0
-10

Examples of General Case

Arbitrary Gaussian distributions can lead to general hyperquadratic 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

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

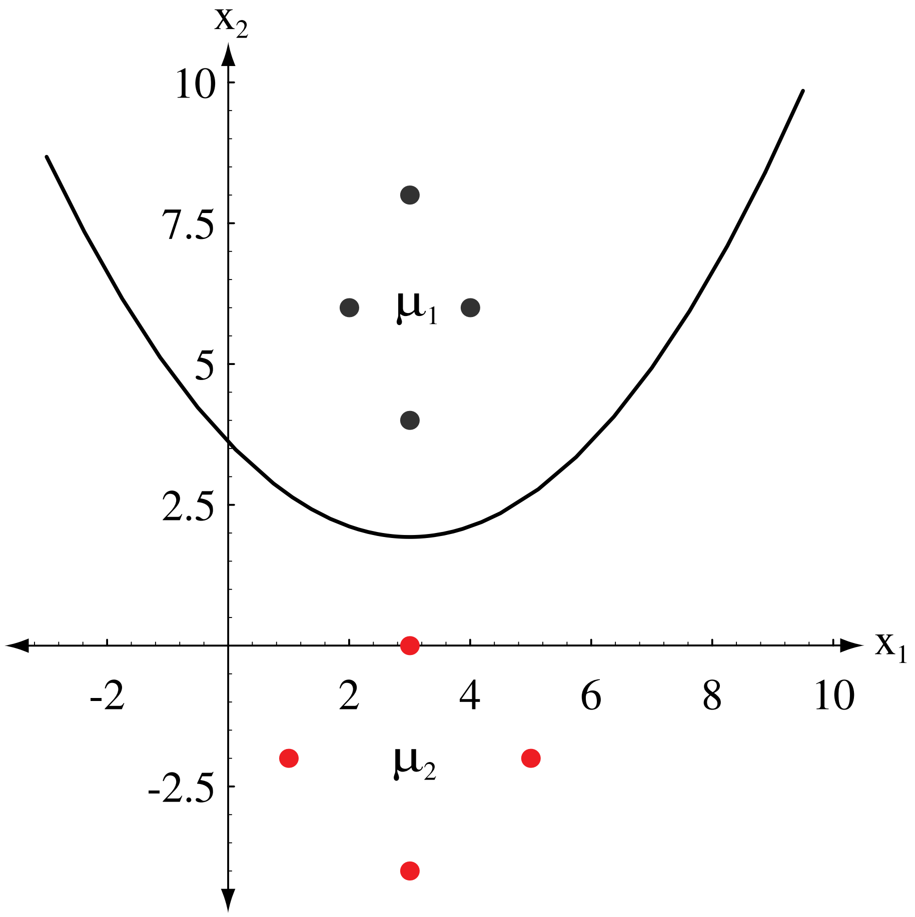
Substituting into the general expression for Gaussian boundaries yields:

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

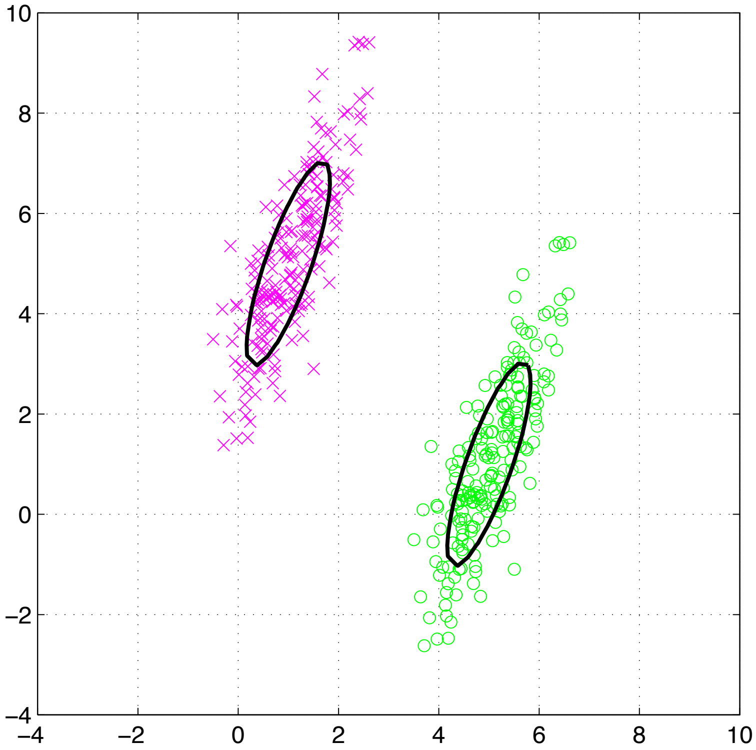
$$1.5x_1^2 - 9x_1 - 8x_2 + 28.11 = 0$$

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

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.


x_2
10
7.5
5
2.5
-2.5
x_1
-2
2
4
6
8
10
μ_1
μ_2

Constrained Case: $\Sigma_i = \Sigma$



Constraining both class-conditional PDF covariance matrices to be the same simplifies the decision boundary

$$2(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)'\boldsymbol{\Sigma}^{-1}\boldsymbol{x} + \boldsymbol{\mu}_1'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 - 2\log\left(\frac{P(\omega_1)}{P(\omega_2)}\right) = 0$$

This is a linear decision boundary

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

- Here the classifier computes a weighted distance called the Mahalanobis distance from the input data $\boldsymbol{x}$ to the mean.

Posterior for $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}$

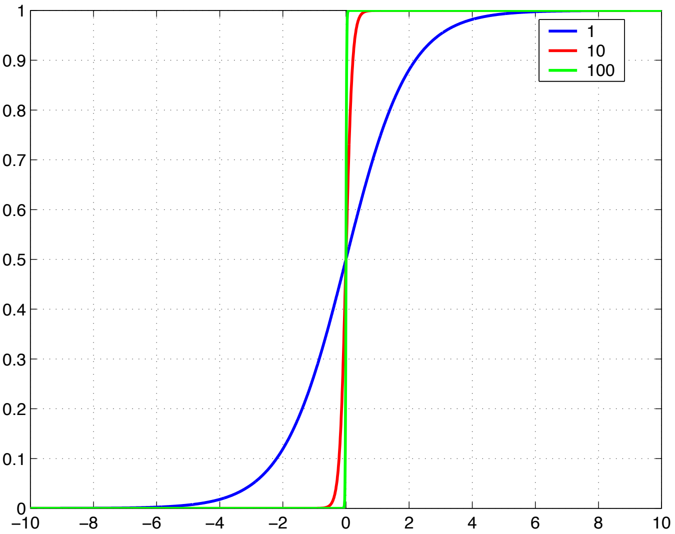
Interesting to look at the posteriors

$$P(\omega_1|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_1)P(\omega_1)}{p(\boldsymbol{x}|\omega_1)P(\omega_1) + p(\boldsymbol{x}|\omega_2)P(\omega_2)} = \frac{1}{1 + \left(\frac{p(\boldsymbol{x}|\omega_2)P(\omega_2)}{p(\boldsymbol{x}|\omega_1)P(\omega_1)}\right)}$$

Simply comparing to the decision working (previous slide)

$$\begin{aligned} P(\omega_1|\boldsymbol{x}) &= \frac{1}{1 + \exp(\boldsymbol{b}'\boldsymbol{x} + c)} \\ \boldsymbol{b} &= 2\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) \end{aligned}$$

This looks like multivariate sigmoid $\frac{1}{1+\exp(-\rho x)}$



From 3F3 this can be compared to logistic regression/classification

$$P(\omega_1|\boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{b}'\boldsymbol{x} - c)} = \frac{1}{1 + \exp(-\tilde{\boldsymbol{b}}'\tilde{\boldsymbol{x}})}$$

- $\tilde{\boldsymbol{x}} = [\boldsymbol{x}' \;\; 1]'$, $\tilde{\boldsymbol{b}} = [\boldsymbol{b}' \;\; c]'$,
- one is a generative model, one is discriminative
- training criteria for the two differ

Training Generative Models

So far the parameters are assumed to be known. In practice this is seldom the case, need to estimate

- class-conditional PDF, $p(\boldsymbol{x}|\omega)$
- class priors, $P(\omega)$

Supervised training, so all the training examples associated with a particular class can be extracted and used to train these models.

The performance of a generative model based classifier is highly dependent on how good the models are. The classifier is the minimum error classifier if

- form of the class-conditional PDF is correct
- training sample set is infinite
- training algorithm finds the correct parameters
- correct prior is used

None of these are usually true!, but things still work (sometimes ...)

Priors can be simply estimated using, for example $\frac{n_1}{n_1+n_2}$

- how to find the parameters of the PDF

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 identically distributed or IID, the joint probability density of the training set is given by

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

$p(\mathcal{D}|\boldsymbol{\theta})$ 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, $\mathcal{L}(\boldsymbol{\theta})$, is maximised instead for convenience *i.e.*

$$\mathcal{L}(\boldsymbol{\theta}) = \log p(\mathcal{D}|\boldsymbol{\theta}) = \sum_{i=1}^{n} \log p(\boldsymbol{x}_i|\boldsymbol{\theta})$$

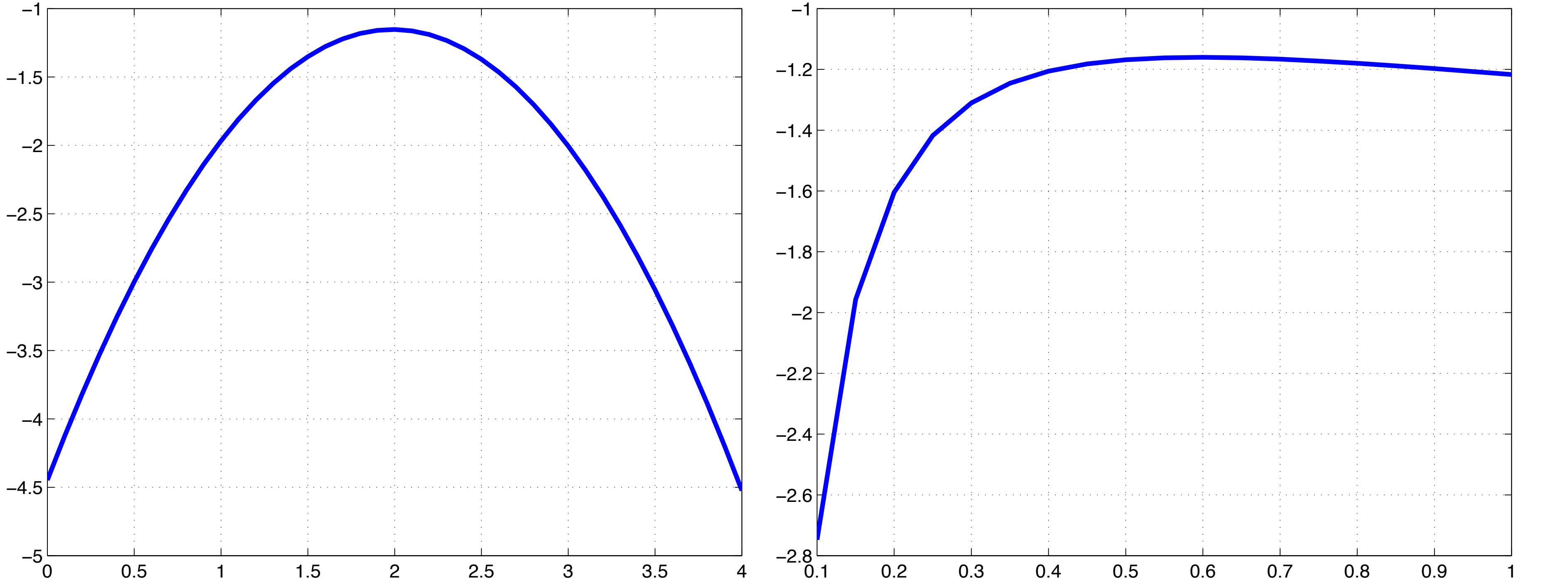
This value can either be maximised by iterative techniques (*e.g.* gradient descent and expectation-maximisation algorithms : see later in the course) or in some cases by a direct closed form solution exists. Either way we need to differentiate the log likelihood function with respect to the unknown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

Maximum Likelihood Criterion

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

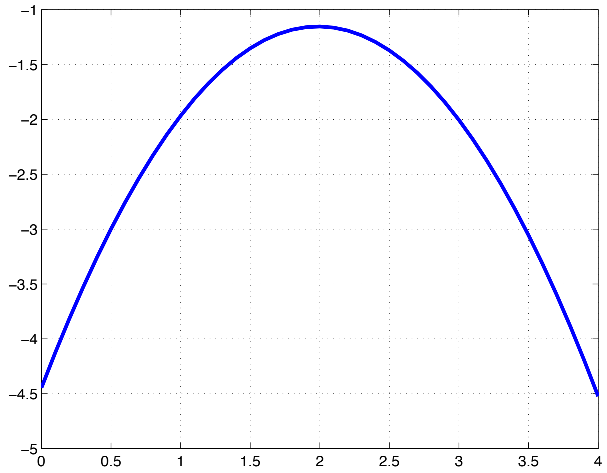
$$\mathcal{L}(\boldsymbol{\theta}) = \log p(\mathcal{D}|\boldsymbol{\theta}) = \sum_{i=1}^{n} \log p(\boldsymbol{x}_i|\boldsymbol{\theta})$$

Gaussian Log-Likelihood

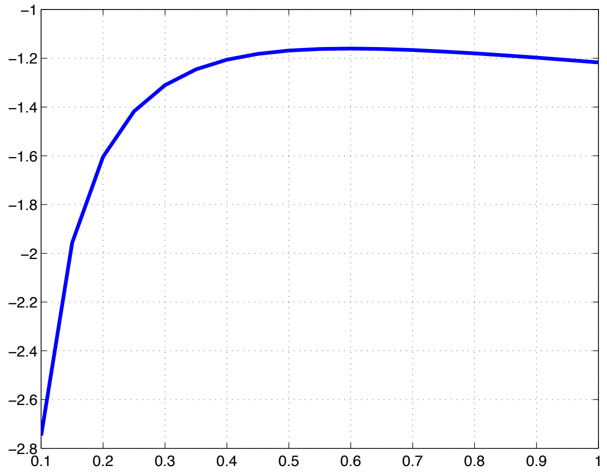


Gaussian Log-Likelihood Functions

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

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

Differentiating this gives

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

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

$$\boldsymbol{\theta} = \left[\begin{array}{c} \boldsymbol{\mu} \\ \mathrm{vec}(\boldsymbol{\Sigma}) \end{array} \right]$$

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 (given for reference only):

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

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

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

There is a difference between this and the ML solution ($\frac{1}{n}$ and $\frac{1}{n-1}$). In the limit as $n \to \infty$ the two values are the same.

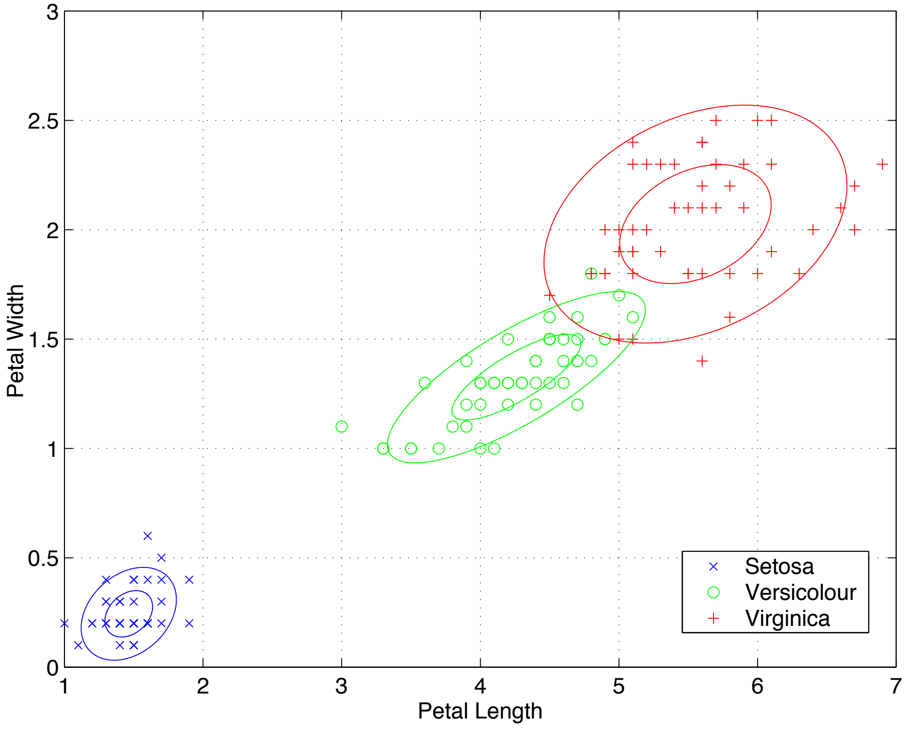
So which is correct/wrong? Neither - they're just different.

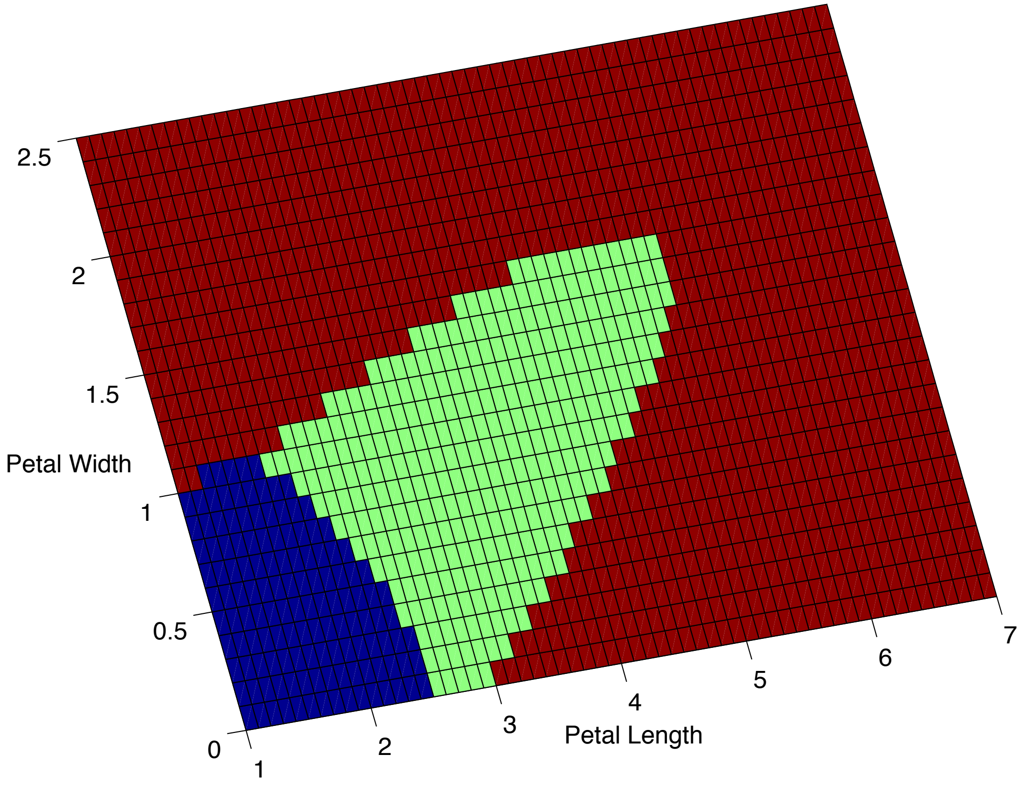
There are two important statistical properties illustrated here.

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.

Iris Data


3
2.5
2
1.5
1
0.5
0
Petal Width
1
2
3
4
5
6
7
Petal Length
× Setosa
○ Versicolour
+ Virginica

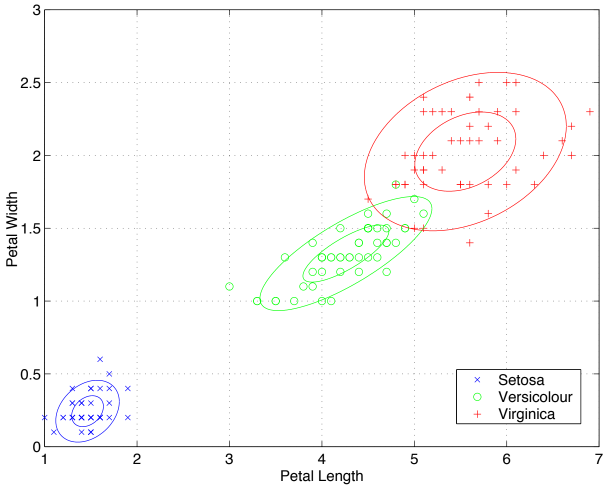

2.5
2
1.5
Petal Width
1
0.5
0
1
2
3
4
5
6
7
Petal Length

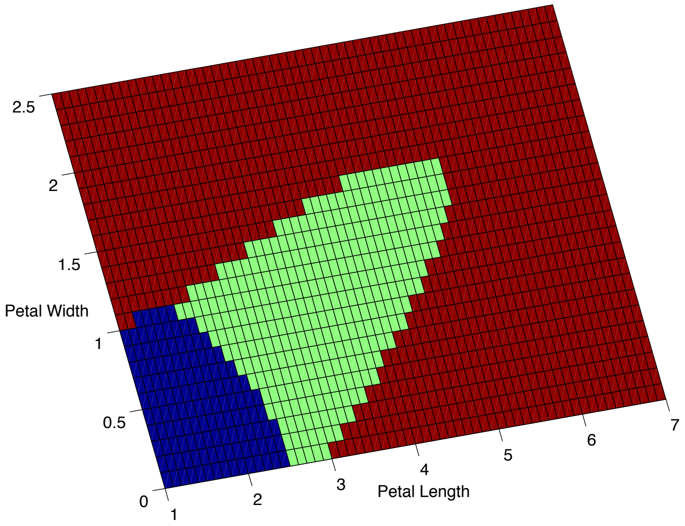
Iris data

Famous (standard) database from machine learning/pattern recognition literature. Measurements taken from three forms of iris

- sepal length and width
- petal length and width

Only petal information considered here.





Use multivariate Gaussians to model each class. Plots show

- data points
- lines at 1 and 2 standard deviations from the mean
- regions assigned using Bayes' decision rule
    - all priors are assumed to be equal

Logistic Regression/Classification

When the covariance matrices of the class-condition Gaussian PDFs are constrained to be the same, the posteriors look like logistic regression - but very different training.

The criterion for training the logistic regression parameters $\tilde{\boldsymbol{b}}$ aims to maximise the likelihood of producing the class labels rather than the observations.

$$\begin{aligned}\mathcal{L}(\tilde{\boldsymbol{b}}) &= \sum_{i=1}^{N} \log(P(y_i|\boldsymbol{x}_i, \tilde{\boldsymbol{b}})) \\ &= \sum_{i=1}^{N} \left[y_i \log\left(\frac{1}{1+\exp(-\tilde{\boldsymbol{b}}'\tilde{\boldsymbol{x}}_i)}\right) + (1-y_i)\log\left(\frac{1}{1+\exp(\tilde{\boldsymbol{b}}'\tilde{\boldsymbol{x}}_i)}\right)\right]\end{aligned}$$

where

$$y_i = \begin{cases} 1, & \boldsymbol{x}_i \text{ generated by class } \omega_1 \\ 0, & \boldsymbol{x}_i \text{ generated by class } \omega_2 \end{cases}$$

where (noting $P(\omega_1|\boldsymbol{x}) + P(\omega_2|\boldsymbol{x}) = 1$)

$$\begin{aligned}P(\omega_1|\boldsymbol{x}) &= \frac{1}{1+\exp(-\tilde{\boldsymbol{b}}'\tilde{\boldsymbol{x}})} \\ P(\omega_2|\boldsymbol{x}) &= \frac{1}{1+\exp(\tilde{\boldsymbol{b}}'\tilde{\boldsymbol{x}})}\end{aligned}$$

Optimised using gradient descent, Newton's method etc (discussed later in the course).

MAP Estimation

It is sometimes useful to use a prior over the model parameters

- high dimensional observation feature space
- limited training data

Both related to curse of dimensionality and how well the classifier will generalise.

Consider a prior on the multivariate Gaussian mean, $\boldsymbol{\mu}$, of the form

$$p(\boldsymbol{\mu}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}_{\rm p}|^{\frac{1}{2}}} \exp\left(-(\boldsymbol{\mu}-\boldsymbol{\mu}_{\rm p})'\boldsymbol{\Sigma}_{\rm p}^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu}_{\rm p})\right)$$

where $\boldsymbol{\mu}_{\rm p}$ and $\boldsymbol{\Sigma}_{\rm p}$ are the parameters of the prior.
The MAP criterion for the mean is

$$\mathcal{F}(\boldsymbol{\mu}) = \mathcal{L}(\boldsymbol{\mu}) + \log(p(\boldsymbol{\mu}))$$

Differentiating and equating $\nabla\mathcal{F}(\boldsymbol{\mu}) = \mathbf{0}$

$$\hat{\boldsymbol{\mu}} = \left(n\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}_{\rm p}^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_{\rm p}^{-1}\boldsymbol{\mu}_{\rm p} + \boldsymbol{\Sigma}^{-1}\sum_{i=1}^{n}\boldsymbol{x}_i\right)$$

- as $\boldsymbol{\Sigma}_{\rm p} \rightarrow \infty$ tends to ML solution
- as $n \rightarrow \infty$ tends to ML solution
- as $n \rightarrow 0$ tends to prior mean

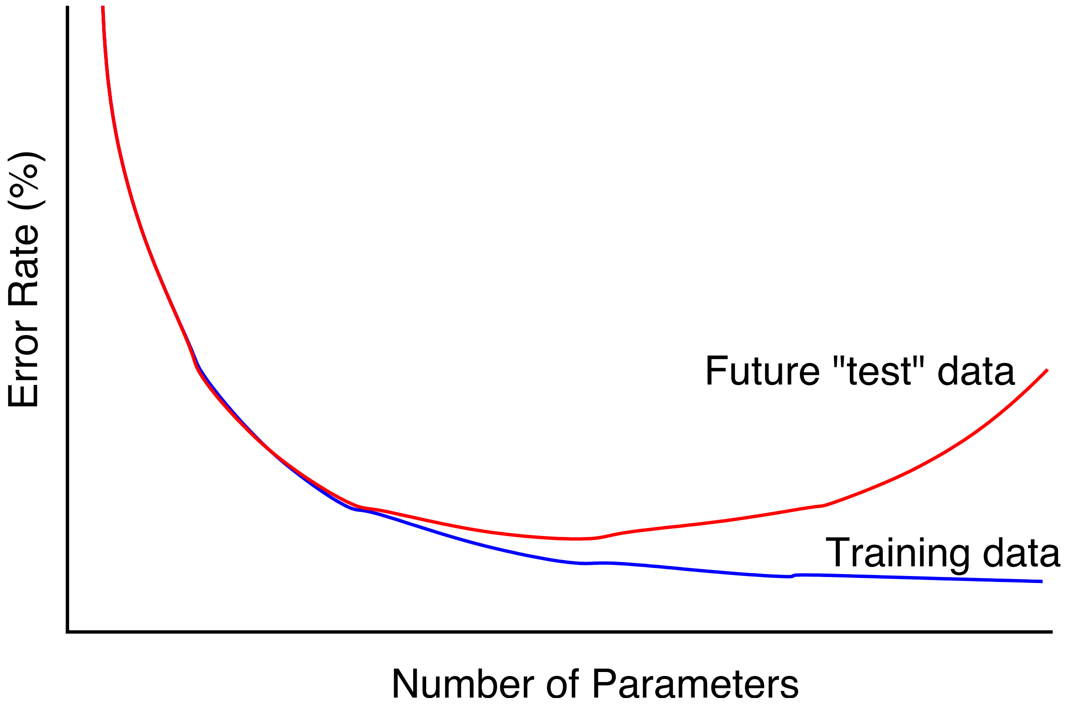
Curse of dimensionality

Given a powerful-enough classifier, or high-enough dimensional observation feature-space, the training data can always be perfectly classified

- think of a look-up-table

BUT we care about performance on held-out data
(we know the labels of the training data!)

Classification of previously unseen data is generalisation.



Often, when designing classifiers, it is convenient to have set of held-out training data that can be used to determine the appropriate complexity of the classifier. This is often called a holdout or validation set.