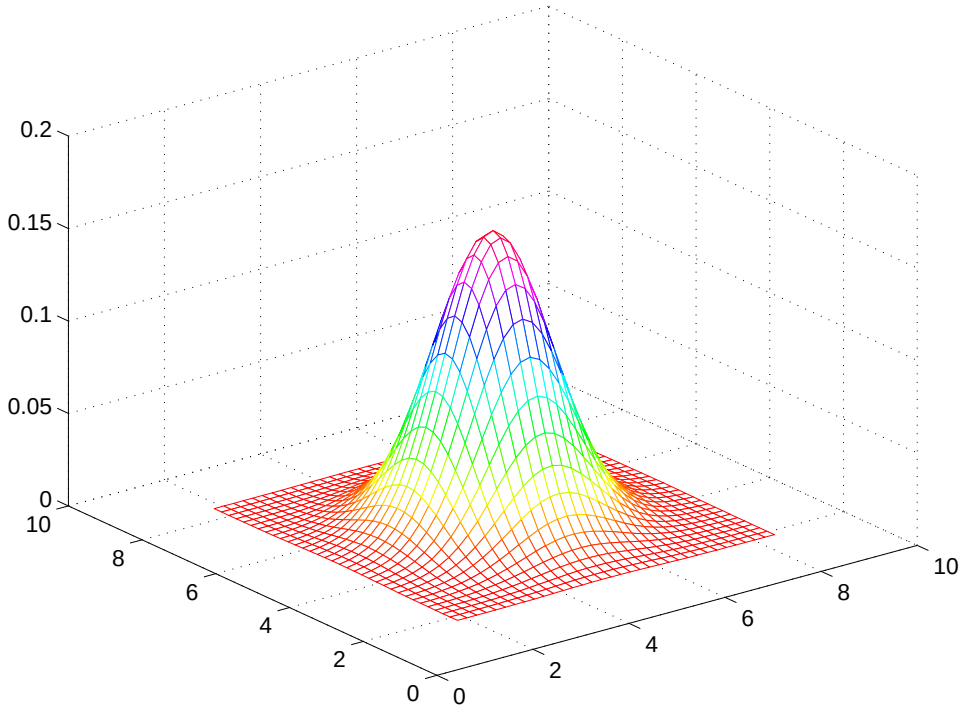
University of Cambridge
Engineering Part IIB

Module 4F10: Statistical Pattern Processing

Handout 1: Introduction & Bayes' Decision Theory



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

Syllabus

1. Introduction & Bayes' Decision Theory (1L)

- Statistical pattern processing
- Bayesian decision theory
- Generalisation

2. Multivariate Gaussian Distributors & Decision Boundaries (1L)

- Decision boundaries for Mulitvariate Gaussians
- Maximum likelihood estimation
- Classification cost
- ROC curves

3. Gaussian Mixture Models (1L)

- Mixture models
- Parameter estimation
- EM for discrete random variables

4. Expectation Maximisation (1L)

- Latent variables both continuous and discrete
- Proof of EM
- Factor analysis

5. Linear Classifiers (1L)

- Single layer perceptron
- Perceptron learning algorithm
- Fisher's linear discriminant analysis
- Limitations

6. Multi-Layer Perceptrons (2L)

- Basic structure
- Posterior distribution modelling
- Regression
- Error back propagation learning
- Second order optimisation methods

7. Support Vector Machines (2L)

- Maximum margin classifiers
- Handling non-separable data
- Training SVMs
- Non-linear SVMs
- Kernel functions

8.Gaussian Processes (2L)

- Gaussian processes
- Covariance functions
- Non-linear regression
- Gaussian processes for classification

9. Classification and Regression Trees (1L)

- Decision trees
- Query selection
- Multivariate decision trees

10. Non-Parametric Techniques (1L)

- Parzen windows
- Nearest neighbour rule

- K-nearest neighbours

11. Application: Speaker Verification and Identification (1L)

- Speaker recognition/verification task
- GMMs and MAP adaptation
- SVM-based verification

Total of 14L + 2 Examples Classes

Lecturers: Prof. Phil Woodland & Dr. Mark Gales

Assessment by exam (1.5h): 3 questions from 5.

A number of books cover parts of the course material.

- C.M.Bishop, *Pattern Recognition and Neural Networks* OUP, 1995, `CUED: NOF 55`
- R.O.Duda, P.E.Hart & D.G. Stork *Pattern Classification*, Wiley, 2001, `CUED: NOF 64`
- D.J.C. Mackay, *Information Theory, Inference and Learning Algorithms*, CUP, 2004. (Also available online) `CUED: NO 277`
- C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer 2006.

What is Statistical Pattern Processing?

The main area of Statistical Pattern Processing discussed in this course is **classification** of **patterns** into different **classes**. The patterns can represent many different types of object.

Typical areas of application include

- Object recognition /classification (e.g. face recognition)
- Speech recognition / speaker identification
- Medical diagnosis
- Financial analysis

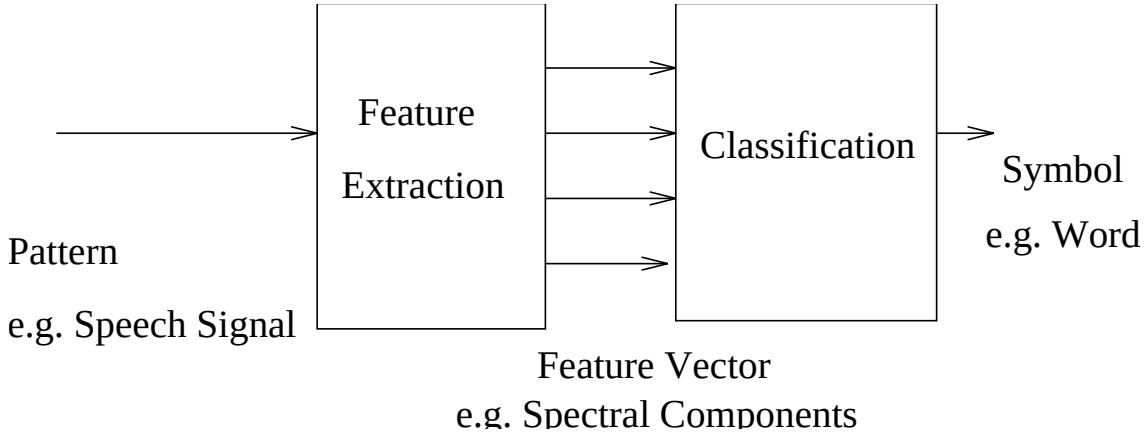
A key issue in all pattern recognition systems is **variability**. Patterns arise (often from natural sources) that contain variations. Key issue: are the variations **systematic** (and can be used to distinguish between classes) or are they **noise**.

The variability of classes will be approached by using **probabilistic modelling** of pattern variations.

The standard model for pattern recognition divides the problem into two parts: feature extraction and classification.

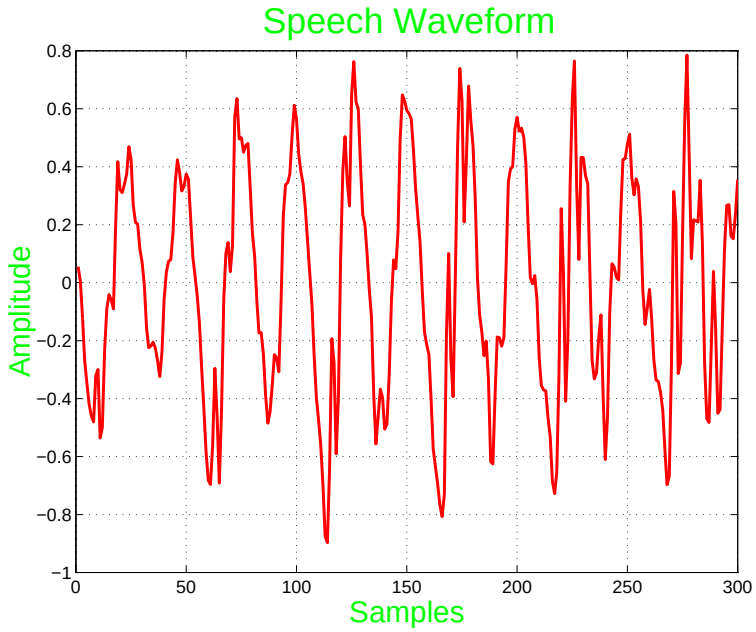
We will also discuss **regression** problems in which the aim is to **predict** a vector of output values from vector of input values.

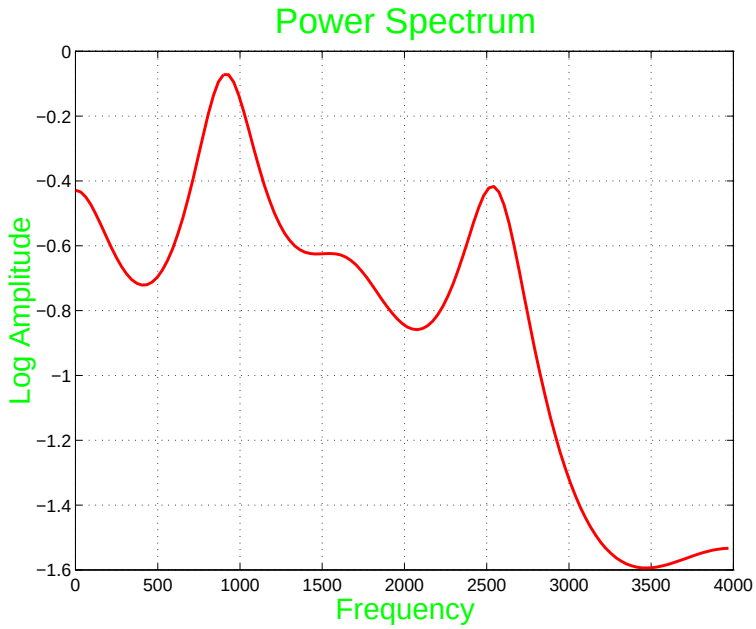
Basic Model



- Initial **feature extraction** produces a **vector** of features that contain all the information for subsequent processing (such as classification).
- Ideally, for classification, only the features that contain discriminatory information are used.
- Often features to measure are determined by an "expert", although techniques exist for choosing suitable features.
- The classifier processes the vector of features and chooses a particular class.
- Normally the classifier is "trained" using a set of data for which there are labelled pairs of feature vectors / class identifiers available. It should be noted that test performance on the classifier training data is biased and distinct training/test sets are needed.

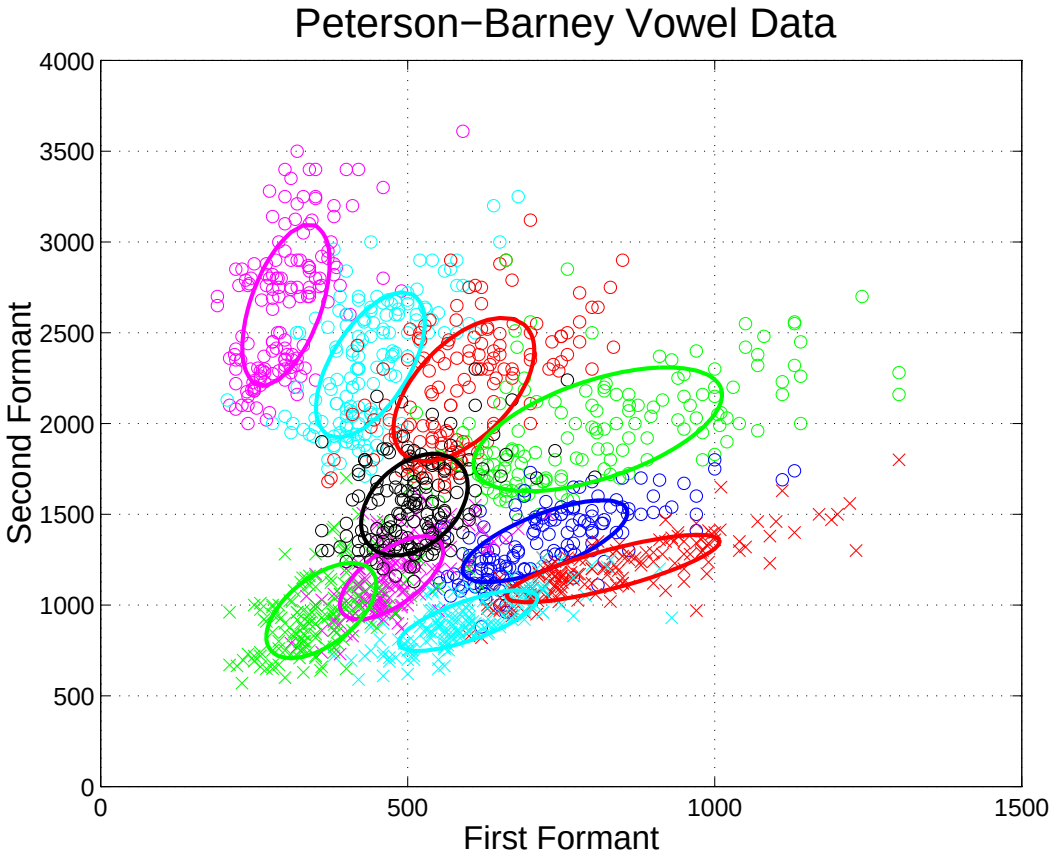
A Speech Classification Example: Features





- Features for vowel classification may be the spectral shape or frequencies of peaks (formants)

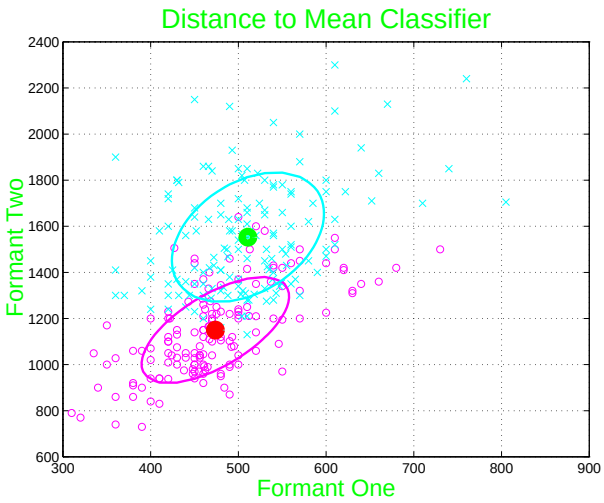
Vowel Distributions Using Formants



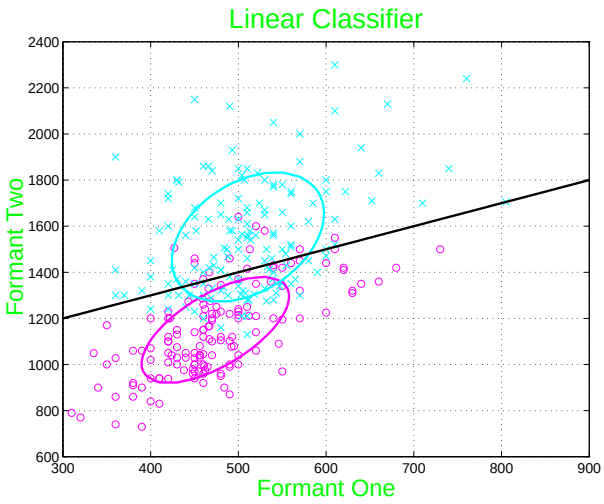
- Vowel classes are reasonably separated (but some overlap!) using these features: could draw **decision boundaries**
- It will be important to be able to calculate the **probability** of a particular class ω_i given a feature vector x i.e. $P(\omega_i|x)$

Simple Classifiers

- Min Distance to Class Mean



- Linear Discriminant.



The min distance to class mean constructs a **linear decision boundary**. There are many other ways to also construct linear boundaries.

- If the **distribution** of the features (for continuous valued features the **probability density function**) can be modelled these types of classifiers (and other more complex types) can be constructed.

Some Basic Probability (Revision!!)

- Discrete random variable x takes one value from the set

$$\mathcal{X} = \omega_1, \ldots, \omega_K$$

We can compute a set of probabilities

$$p_j = \Pr(x = \omega_j), \quad j = 1, \ldots, K$$

We use a probability mass function $P(x)$, to describe the set of probabilities. The PMF satisfies

$$\sum_{x \in \mathcal{X}} P(x) = 1, \quad P(x) \geq 0$$

- Continuous random variable: scalar x or a vector $\mathbf{x}$. Described by its probability density function (PDF), $p(x)$. The PDF satisfies

$$\int_{-\infty}^{\infty} p(x)dx = 1, \quad p(x) \geq 0$$

- For random variables x, y, z need

conditional distribution: $p\left(x|y\right) = \frac{p(x,y)}{p(y)}$

joint distribution $p\left(x,\, y\right)$

marginal distribution $p\left(x\right) = \int_{-\infty}^{\infty} p(x,y)dy$

chain rule $p(x,y,z) = p(x|y,z)\, p(y|z)\, p(z)$

Bayes' Rule

Since $p(x, y) = p(y, x)$ the formula for conditional probability leads to

$$\begin{aligned} p\left(x|y\right) p\left(y\right) &= p\left(y|x\right) p\left(x\right) \\ p\left(y|x\right) &= \frac{p\left(x|y\right) p\left(y\right)}{p\left(x\right)} \end{aligned}$$

This last form is known as **Bayes' Rule**. It will also be particularly useful to us in the form

$$P(\omega_j|x) = \frac{p(x|\omega_j)P(\omega_j)}{p(x)}$$

Bayes' rule here computes the **posterior probability** of a particular class, $P(\omega_j|x)$) using the **likelihood** of the data computed from the class conditional density $p(x|\omega_j)$.

The term $P(\omega_j)$ is known as the the **prior** probability of the class ω_j. This is the probability of the class before any data is observed.

The denominator of this form of Bayes' Rule can be computed as

$$p(x) = \sum_j p(x|\omega_j)P(\omega_j)$$

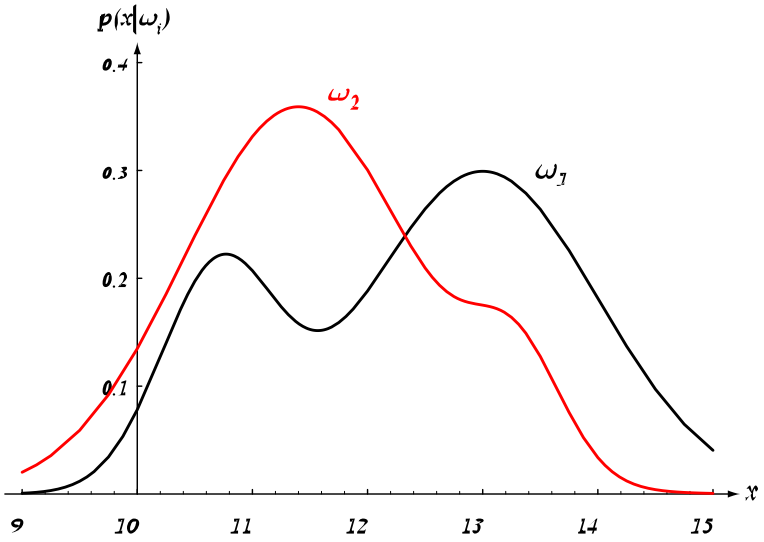
This is sometimes termed the **evidence** and is the probability density of the data independent of class.

Bayes' Rule is sometimes remembered as

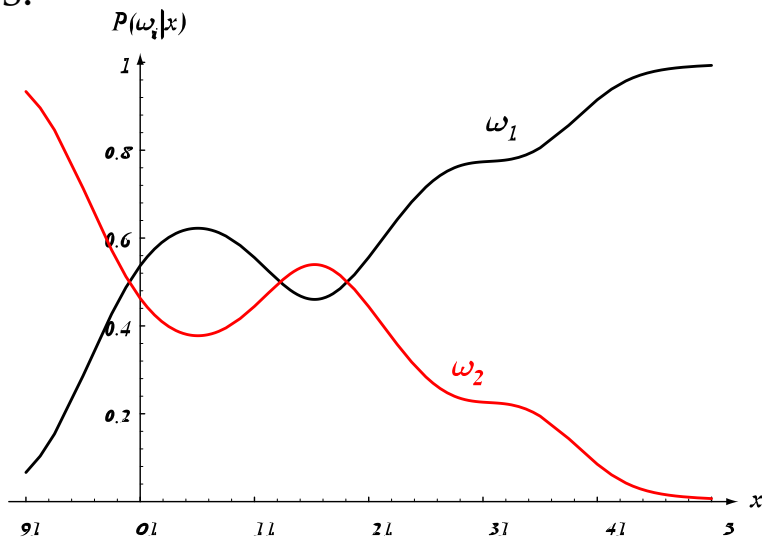
$$\text{posterior} \propto \text{likelihood} \times \text{prior}$$

The figures below (from DHS) give hypothetical class-conditional pdfs for two classes, and with $P(\omega_1) = 2/3$ and $P(\omega_2) = 1/3$, the posterior distribution.

pdfs:



posteriors:



Bayesian Decision Theory

Our goal in creating a decision rule here is to minimise an average probability of error which is calculated by integrating the joint probability of error and the feature over the space of $\boldsymbol{x}$.

$$\begin{aligned} P(\text{error}) &= \int P(\text{error}, \boldsymbol{x}) d\boldsymbol{x} \\ &= \int P(\text{error}|\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} \end{aligned}$$

For a two class problem, the conditional probability of error, (*i.e.* the error probability, given a value for the feature vector), can be written as

$$P(\text{error}|\boldsymbol{x}) = \begin{cases} P(\omega_1|\boldsymbol{x}) & \text{if we decide } \omega_2 \\ P(\omega_2|\boldsymbol{x}) & \text{if we decide } \omega_1 \end{cases}$$

A decision rule that can minimise this conditional probability of error and apply it to every example, then we will be minimising the average probability of error. This leads to Bayes' decision rule, which for a two class problem is

$$\text{Decide} \begin{cases} \text{Class}\omega_1 & \text{if } P(\omega_1 |\boldsymbol{x}) > P(\omega_2 |\boldsymbol{x}); \\ \text{Class}\omega_2 & \text{Otherwise} \end{cases}$$

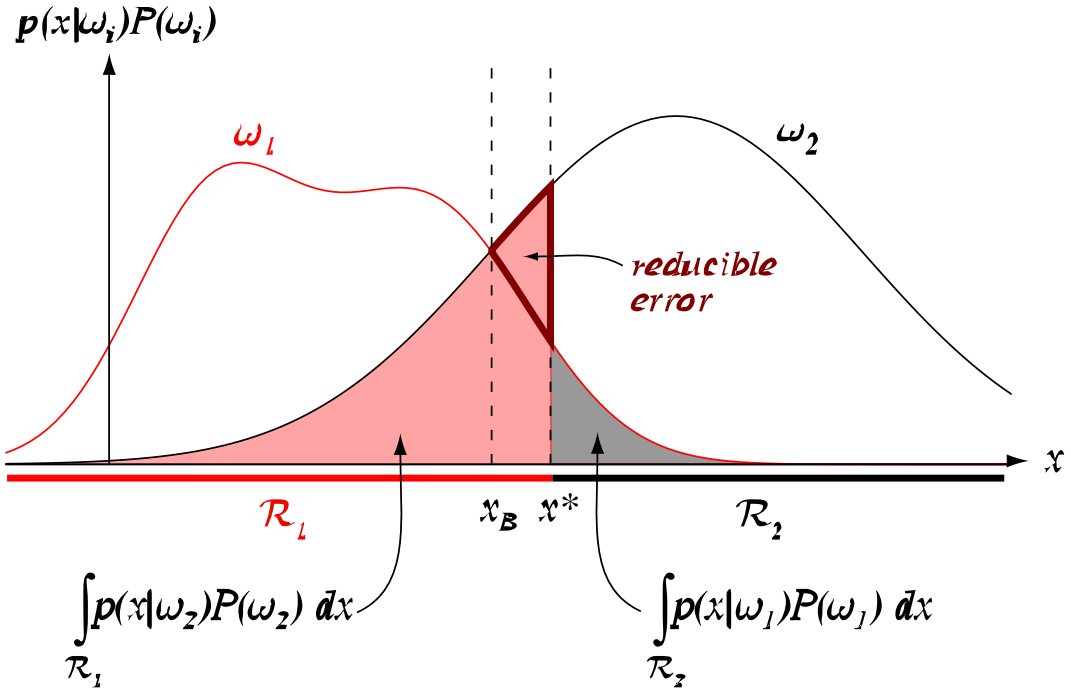
Note that to find the overall probability of error for the two-class problem, there are two regions defined by the decision rule, decide for class ω_1 in $\mathcal{R}_1$ and ω_2 in $\mathcal{R}_2$ and

$$P(\text{error}) = P(\boldsymbol{x} \in \mathcal{R}_2, \omega_1) + P(\boldsymbol{x} \in \mathcal{R}_1, \omega_2)$$

$$
\begin{aligned}
&= P(\boldsymbol{x} \in \mathcal{R}_2|\omega_1)P(\omega_1) + P(\boldsymbol{x} \in \mathcal{R}_1|\omega_2)P(\omega_2) \\
&= \int_{\mathcal{R}_2} p(\boldsymbol{x}|\omega_1)P(\omega_1)d\boldsymbol{x} + \int_{\mathcal{R}_1} p(\boldsymbol{x}|\omega_2)P(\omega_2)d\boldsymbol{x}
\end{aligned}
$$

It is impossible to find a closed form solution for $P(\text{error})$ except in some fairly simple cases (which includes the important one Gaussian distributions with equal priors).

The error regions for a two-class problem are shown below (from DHS). The decision boundary x^* is set to x_B for minimum error.



For the two-class case the Bayes' minimum average error decision rule could be written as a ratio of posterior probabilities:

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

or alternatively the **likelihood ratio** used & included the class

priors in the threshold comparison

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

For multi-class problems, we calculate all the C posterior probabilities

$$P(\omega_1|\boldsymbol{x}), P(\omega_2|\boldsymbol{x}), ..., \ P(\omega_C|\boldsymbol{x}),$$

find the maximum of these and assign the vector $\boldsymbol{x}$ to the corresponding class.

Applying Bayes' Rule to this formula we need to find the class ω_j which gives

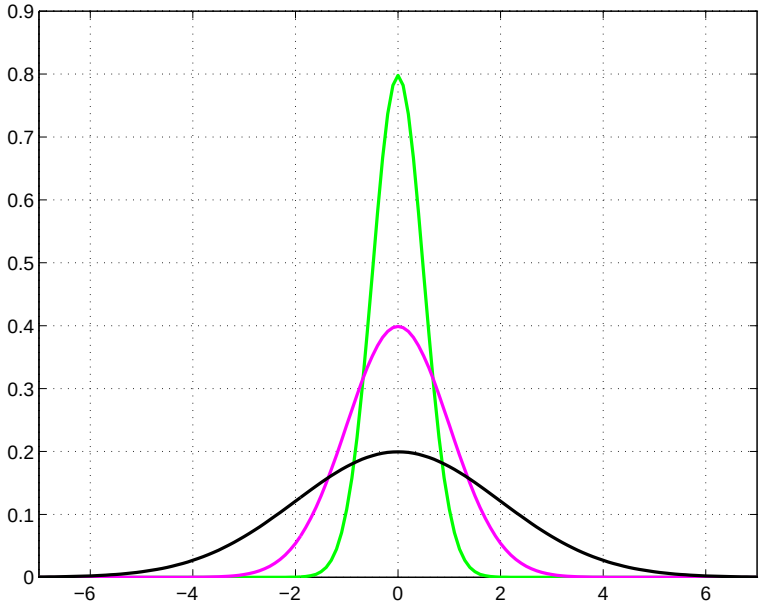
$$\max_j p(\boldsymbol{x}|\omega_j)P(\omega_j)$$

since the rhs denominator of Bayes' rule is independent of class and this is a frequent statement of Bayes' decision rule for minimum error.

Gaussian (Normal) Distribution

- Univariate Gaussian Distribution

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$



- The Gaussian is the most important distribution that we will work with, and in many important cases class conditional densities are approximately Gaussian
- Samples drawn from a Gaussian density tend to be clustered around the mean μ, the spread of the samples is proportional to σ.
- Unimodal and symmetric about the mean

- Defined fully by the mean and standard deviation (or the variance)
- One simple way to estimate the parameters of a Gaussian distribution is to set the parameters the **sample mean** and **standard deviation** from a training set.

$$\begin{aligned} \mu &= \mathcal{E}[x] \\ &= \int x\, p(x)\, dx \\ \sigma^2 &= \mathcal{E}[(x-\mu)^2] \\ &= \int (x-\mu)^2\, p(x)\, dx \end{aligned}$$

- Usually the feature representations we work with are d-dimensional vectors. It is possible to model these using a univariate Gaussian in each dimension as

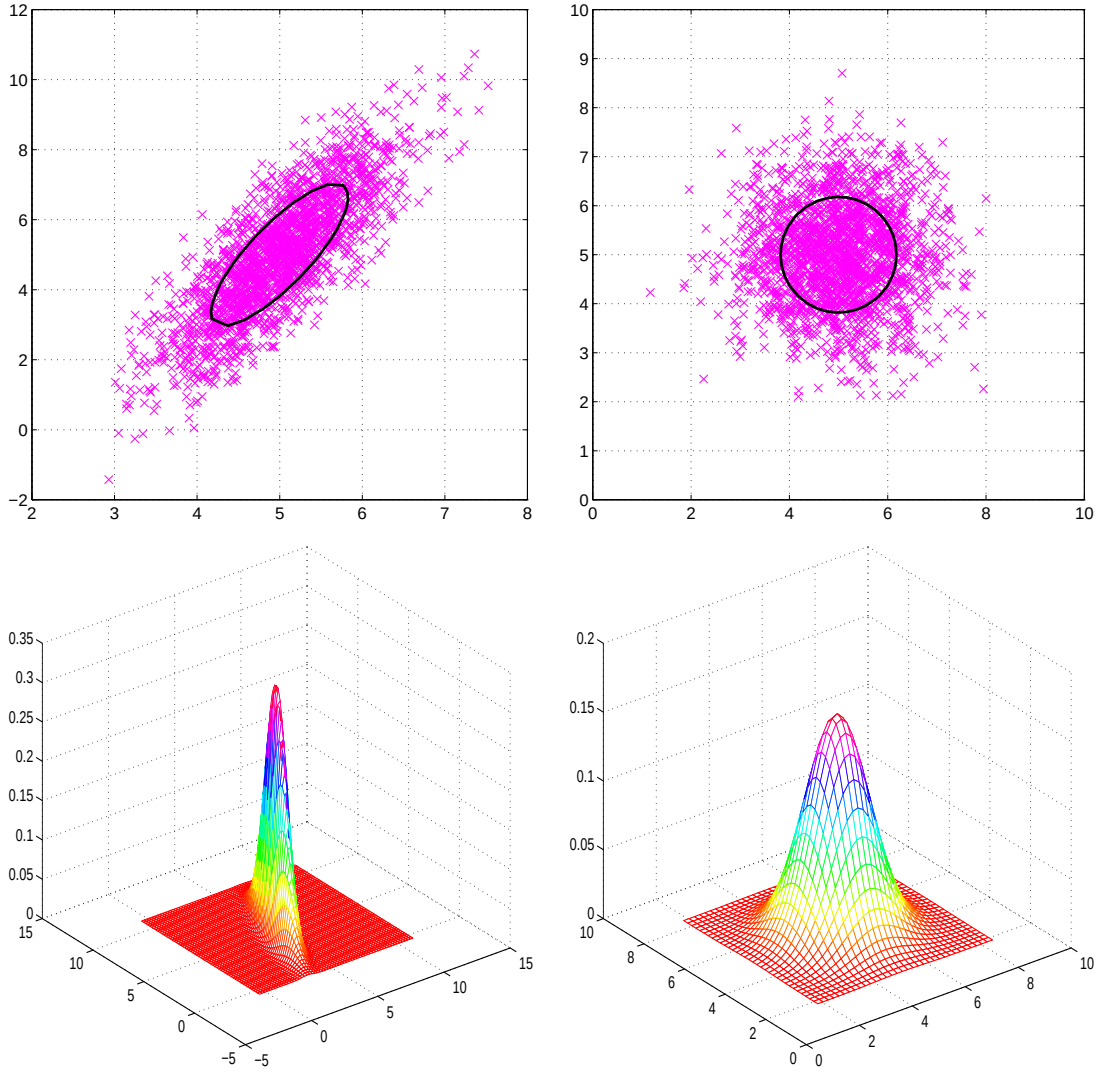
$$p\left(\boldsymbol{x}\right) = \prod_{i=1}^{d} \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left(-\frac{\left(\boldsymbol{x}_i - \mu_i\right)^2}{2\,\sigma_i^2}\right)$$

  However in this case we are assuming that the different feature vector elements are **uncorrelated**. The full **multivariate** distribution can take this into account.

Multivariate Gaussian

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

$$\boldsymbol{\Sigma} = \begin{pmatrix} 3 & 1 \\ 1 & 0.5 \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$



- The distribution is characterised by the mean vector and the covariance matrix Σ

- The mean and covariance matrix are defined as

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

  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 dimesionsions 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.
- 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 (we will look at this in more detail later).
- 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 $p(\boldsymbol{y})$ is Gaussian with

$$\begin{aligned}\boldsymbol{\mu}_y &= \boldsymbol{A}^{'}\boldsymbol{\mu}_x + \boldsymbol{b} \\ \boldsymbol{\Sigma_y} &= \boldsymbol{A}^{'}\boldsymbol{\Sigma}_x\boldsymbol{A}\end{aligned}$$

Training Classifiers & Generalisation

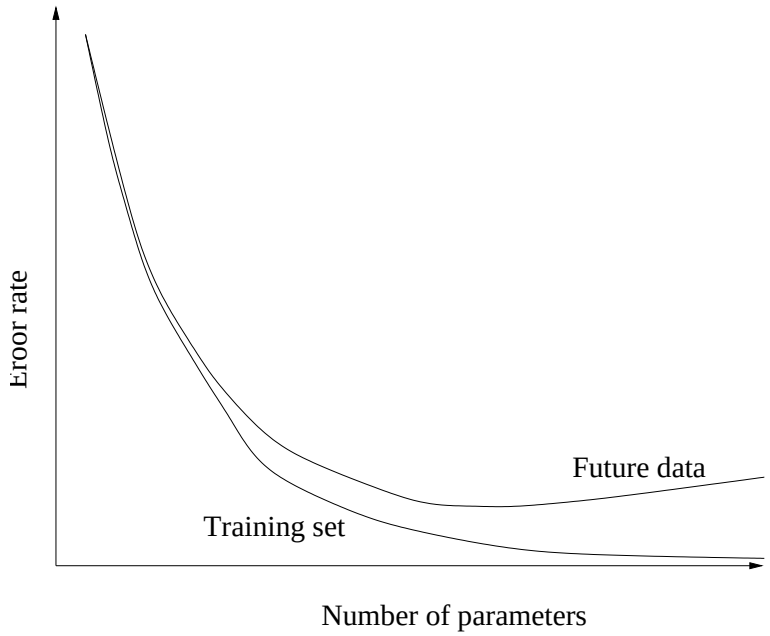
It should be noted that, in practice, there are a number of stages in constructing a good claassifier using the Bayes formulation

1. Decide on the features
2. Have a **training** or **design** set of labelled examples
3. **Assume** a form of class-conditional pdf (e.g. Gaussian)
4. **Estimate** the parameters of the pdfs from the (limited) training data
5. **Estimate** the class priors
6. Estimate the error performance on new **test** data

Note that due to the assumptions and estimates above the classifier may not be the best possible

Note that the aim is normally to get good performance on some previously unseen test data.

Typically as we increase the number of parameters in the class-conditional pdfs (e.g. using full-covariance model rather than diagonal) the training data classification error rate *usually* decreases. However the test (future) set performance has a minimum.



The graph may be split into three regions:

1. **Too Simple**: The models used are relatively simple. The performance on the training and test data is about the same as the models are "well" trained.

2. **Just Right**: This is where the error rate on the test data is at a minimum. This is where we want to be.

3. **Too Complex**: The models perform very well on the training data. However the models perform badly on the test data.

The objective in any pattern classification task is to have the minimum test set (future data) error rate.

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.