RANDOM SIGNALS

SIGNAL CLASSIFICATION

P.R. WHITE

GENERAL COMMENTS

FEATURES VECTORS

1-D GAUSSIAN CLASSIFIER

2-D GAUSSIAN CLASSIFIER

SIMPLIFIED GAUSSIAN CLASSIFIER

k-NEAREST NEIGHBOUR CLASSIFIERS

 © ISVR 2003

INTRODUCTION

One of the most widely considered problems in DSP is that of signal classification.

Typical examples are:

Speech Recognition

Automatic Fingerprint/Face/Gait Recognition

Classification of SONAR/RADAR/EM signals

The problem can be posed as given a data set **x** determine which of a finite set of hypotheses is most likely (an *M*-ary hypothesis test).

More formally, to find the maximum of:

$$P(H_1 \mid \mathbf{x}),\ P(H_2 \mid \mathbf{x})\ \ldots..\ P(H_M \mid \mathbf{x})$$

where $P(H_k \mid \mathbf{x})$ is the probability of kth hypothesis given the data **x**.

The hypothesis corresponds to the possible signal sources, e.g. the speech sound (phoneme), the fingers, the animal calls, etc.

 © ISVR 2003

TYPES OF CLASSIFIER

Parametric

Assumes an underlying model for the pdfs of the classes.

The parameters of the class pdfs are estimated from the training data.

By assuming a model one looses generality but reduces the amount of data need to accurately estimate the model parameters.

Non-Parametric

Assumes no model for the class pdfs.

Is very general, but requires more data to train.

We shall primarily concentrate on parametric methods and only touch on non-parametric techniques at the end.

 © ISVR 2003

FEATURE EXTRACTION

Most (if not all) signal classification methods do not act directly on raw data.

The first step in a classification algorithm is usually to extract a few salient features from the raw data. This is called *feature extraction*.

The features aim to act as a summary of the data.

The aim is to reduce the dimension of the problem, without significant loss of information.

The most suitable set of features depends strongly upon the application under consideration.

Throughout this lecture the notation **x** will be used interchangeably for both the raw data and the features.

Indeed one can regard the features as replacing the data.

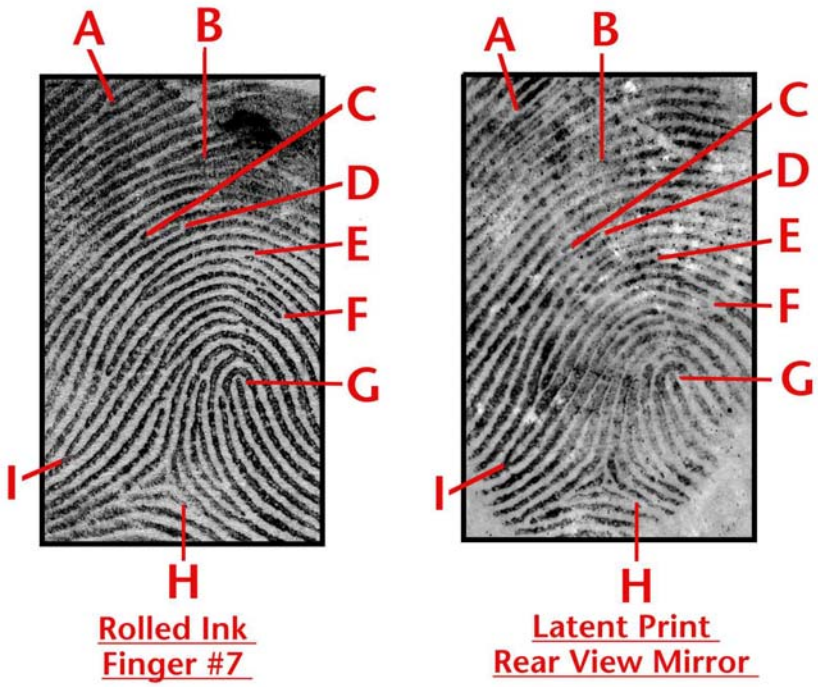
The performance of a classifier **critically** depends on the effectiveness the chosen features.

 © ISVR 2003

EXAMPLE OF (HUMAN) FEATURE EXTRACTION

Consider the problem of recognising a fingerprint.

When presented with a fingerprint, forensic scientists first find key characteristic features in the finger print and compare only them.



Similarly an automatic fingerprint recognition system does not deal with the full image of a fingerprint (which maybe 128 by 128 pixels), but seeks to reduce the dimension of the problem by first performing feature extraction.

 © ISVR 2003

COMPUTATION OF THE PROBABILITIES

The principle of signal classification is straightforward. One merely has to compute all of the probabilities $P(H_k \mid \mathbf{x})$ and choose the hypothesis that yields the largest probability.

The problem now becomes one of computing $P(H_k \mid \mathbf{x})$, to do this we resort to Bayes' rule:

$$P(H_k \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid H_k) P(H_k)}{P(\mathbf{x})}$$

The right-hand side of this expression is more amenable than the left.

We are testing to see if $P(H_k \mid \mathbf{x}) > P(H_j \mid \mathbf{x})$ or equivalently if $\frac{P(\mathbf{x} \mid H_k) P(H_k)}{P(\mathbf{x})} > \frac{P(\mathbf{x} \mid H_j) P(H_j)}{P(\mathbf{x})}$, so that $P(\mathbf{x})$ can be cancelled.

Hence to decide if H_k is more likely than H_j we need only test to see if $P(\mathbf{x} \mid H_k) P(H_k) > P(\mathbf{x} \mid H_j) P(H_j)$

$P(H_k)$ is the prior probability for each hypothesis.

$P(\mathbf{x} \mid H_k)$ is the probability of observing the data given that the hypothesis H_k is true.

 © ISVR 2003

THE ROLE OF THE PRIORS

The prior probabilities $P(H_k)$ reflect the probability of the kth hypothesis occurring, without reference to the measured data.

Clearly these priors should sum to 1, so that:

$$\sum_{k=1}^{M} p(H_k) = 1$$

For example in a speech recogniser one might assign high prior probability to the vowel sounds since these are generally more common in speech that the consonants.

One can regard the priors as weights, which allow the user to emphasise particular hypotheses over others.

In the absence of prior information, a sensible option is to construct a "fair" classifier by selecting:

$$P(H_1) = P(H_2) = ... = P(H_M) = \frac{1}{M}$$

In this the tests we need to construct are simply:

$$P(\mathbf{x} \mid H_k) > P(\mathbf{x} \mid H_j)$$

These are the form of classifier we shall concentrate on here.

 © ISVR 2003

GENERAL STRUCTURE

The general structure of our classifier is then:

1) Take the raw data and extract the features **x**.

2) Compute the probability that **x** would have been generated assume that hypothesis H_k is true.

3) Weight the probabilities computed in 2) by their priors.

4) Find the maximum of the probabilities found in 3).

If equal priors are used then the classifier selects the hypothesis that is most likely to have given rise to the observed data (features).

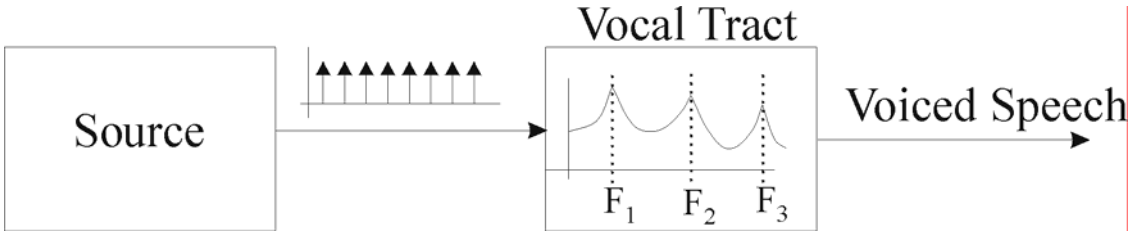
The tricky bit of this problem comes in step 2), how does one compute the probability of a data set under a given hypothesis?

 © ISVR 2003

EXAMPLE (BACKGROUND)

Consider the problem of classifying whether a speech sound is a /a/ (as in sat), a /o/ (as in soot), or a /e/ (as in seat).

The three sounds /a/, /o/ and /e/ are examples of voiced speech which can be modelled as below:



The source represents the vocal folds that open and close to produce a sequence of pulses of air. The intervals between the pulses dictate the pitch of the speech.

This pulse sequence excites the vocal tract, which is a resonant cavity. These resonant frequencies are modified by changing mouth position etc. and are called the “formant” frequencies.

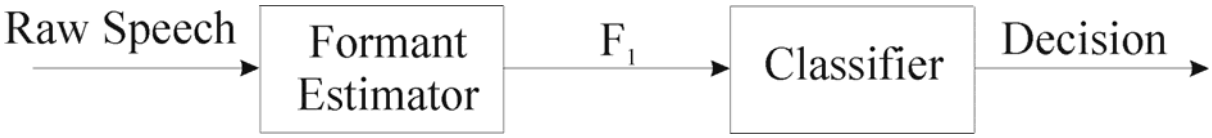
Broadly speaking he formant frequencies control “what is said”, whilst the pitch controls “how it is said”.

Natural features for a speech recognition system to use are estimates of the formant frequencies.

 © ISVR 2003

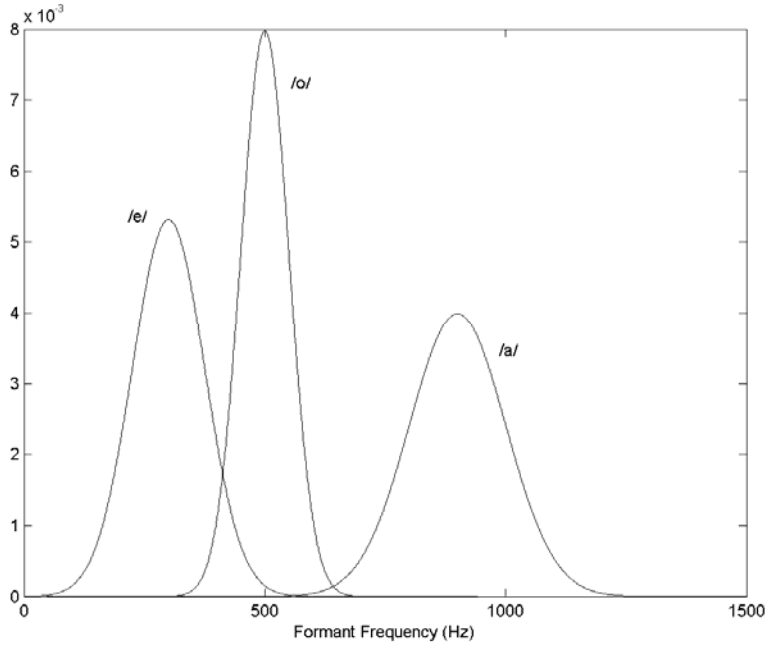
EXAMPLE: A 1-D CLASSIFICATION PROBLEM (I)

To classify speech into one of our 3 sounds (/a/, /o/ or /e/) we might estimate the first formant frequency F_1 and use that as our only feature (such a system is a 1-D classifier).



Let us assume that the pdfs for the first formant for the various speech sounds are Gaussian and that the means and standard deviations are given by

	Mean (μ)	Standard Dev. (σ)
/a/	900	100
/e/	300	75
/o/	500	50



 © ISVR 2003

EXAMPLE: A 1-D CLASSIFICATION PROBLEM (II)

The optimum classifier (assuming no prior information) chooses the speech sound whose pdf is largest for the measured formant frequency. The classifier consists of:

1) Measuring the first formant frequency (F1) from the received speech sound.
2) Computing the likelihoods using each of the speech sound pdfs for (/a/, /o/ or /e/)
3) Deciding which pdf is greatest.

For a Gaussian process the pdf is defined as:

$$p(F_1) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(F_1-\mu)/2\sigma^2}$$

At step 2) one computes the above for each of the hypotheses (speech sounds) and compares their values.

 © ISVR 2003

PARTITIONING OF THE FEATURE SPACE

In practice the one does not need to compute the pdfs for all of the hypotheses, one can in fact partition the feature space.

In the previous one can simplify the classifier by noting that if:

$F_1 < 412$ then one will classify the signal as an /e/.

$412 < F_1 < 642$ then one will classify the signal as an /o/.

$F_1 > 642$ then one will classify the signal as an /a/.

This obviates the need compute values of pdfs, as in step 2) (on previous O/H.

The boundaries in the above are defined by the points in feature space where two adjacent pdfs have the same value (i.e. where they cross).

In the case of Gaussian distributions are then computed by solving:

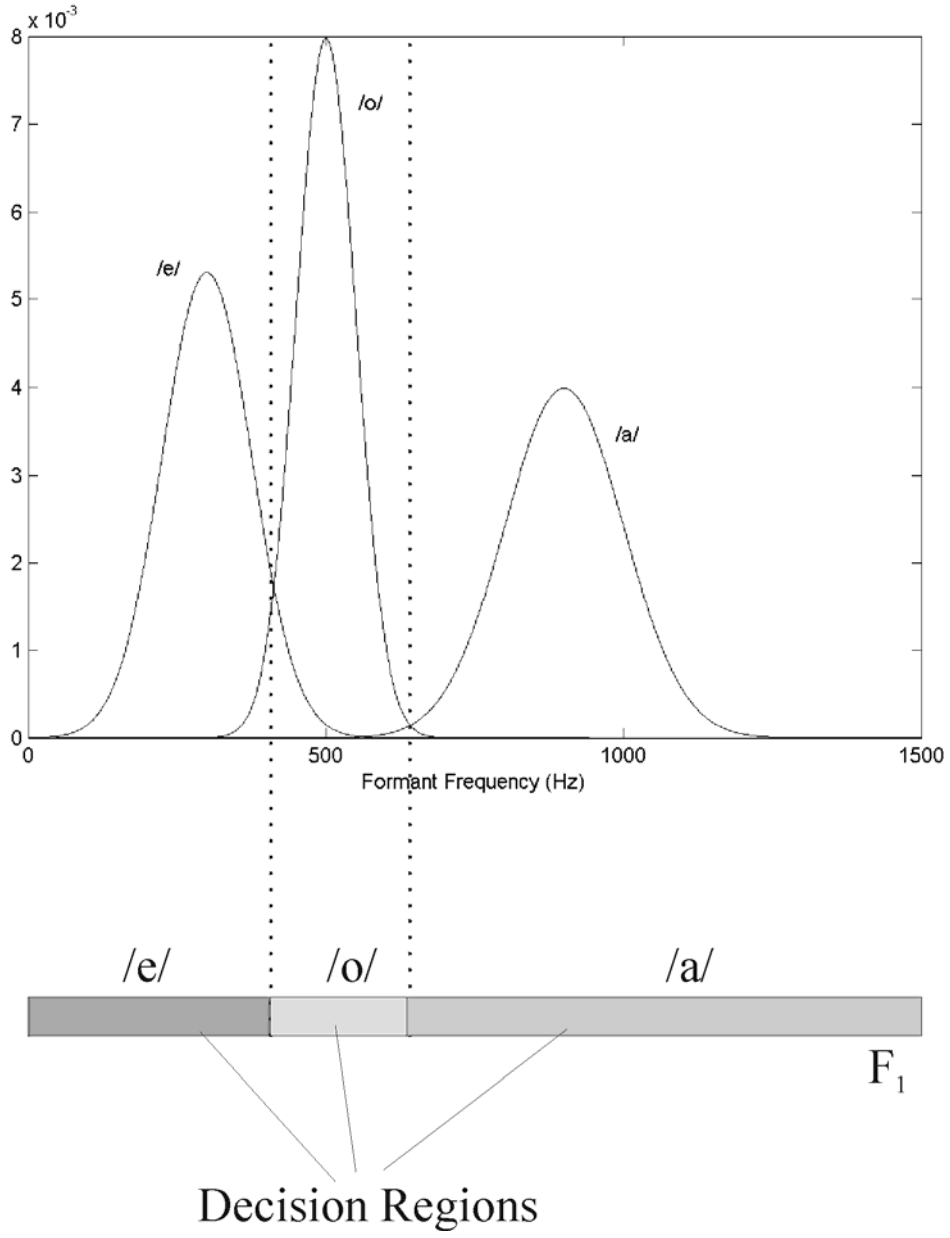
$$\frac{1}{\sigma_1\sqrt{2\pi}} e^{-(x-\mu_1)/2\sigma_1^2} = \frac{1}{\sigma_2\sqrt{2\pi}} e^{-(x-\mu_2)/2\sigma_2^2}$$

$$\Rightarrow \ln\left\{\frac{\sigma_1^2}{\sigma_2^2}\right\} + \left(\frac{x-\mu_1}{\sigma_1}\right)^2 - \left(\frac{x-\mu_2}{\sigma_2}\right)^2 = 0$$

where μ_1 and σ_1 are the mean and standard deviation of the class below a boundary and μ_2 and σ_2 correspond to the class above the boundary.

 © ISVR 2003

PARTITIONING PICTORIALLY



Decision regions are the areas of in the feature space that correspond to making a particular classification.

 © ISVR 2003

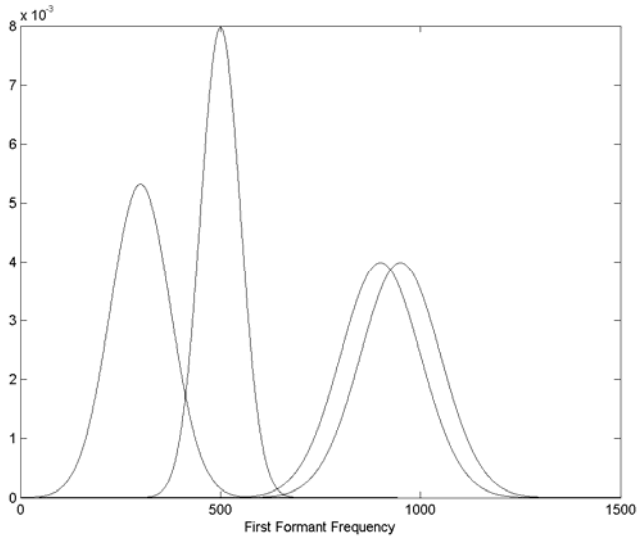
EXTENDING THE EXAMPLE

The example (as presented so far) is based on classifying speech according to only the first formant frequency.

There are typically around 50 different speech sounds and to find a unique region for each sound on the single axis reflecting the first formant is unreasonable.

Consider for example the sound /õ/ (as in sot). The distribution of the first formant frequency has a mean of 950 and a standard deviation of 100.

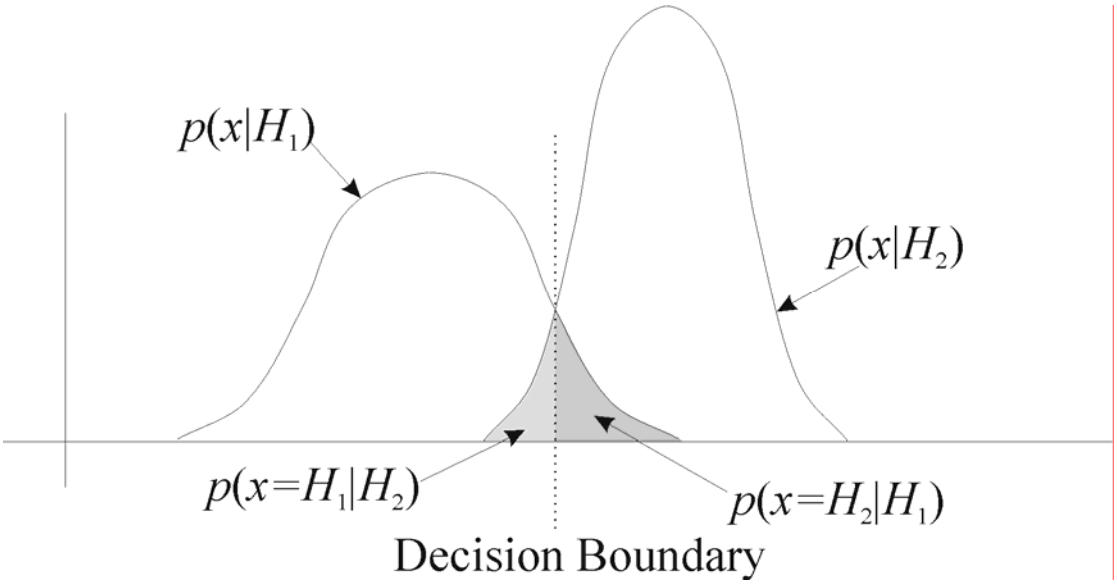
Basing a classifier on the first formant frequency only will result in system that frequently confuses /õ/ with /a/.



 © ISVR 2003

MISCLASSIFICATIONS

One can compute the probability that a classifier will make an erroneous classification by computing $p(x = H_1 \mid H_2)$, i.e. probability of deciding that *x* came from hypothesis 1 given that it actually came from hypothesis 2.



If two pdfs overlap significantly then the probability of making a misclassification increases.

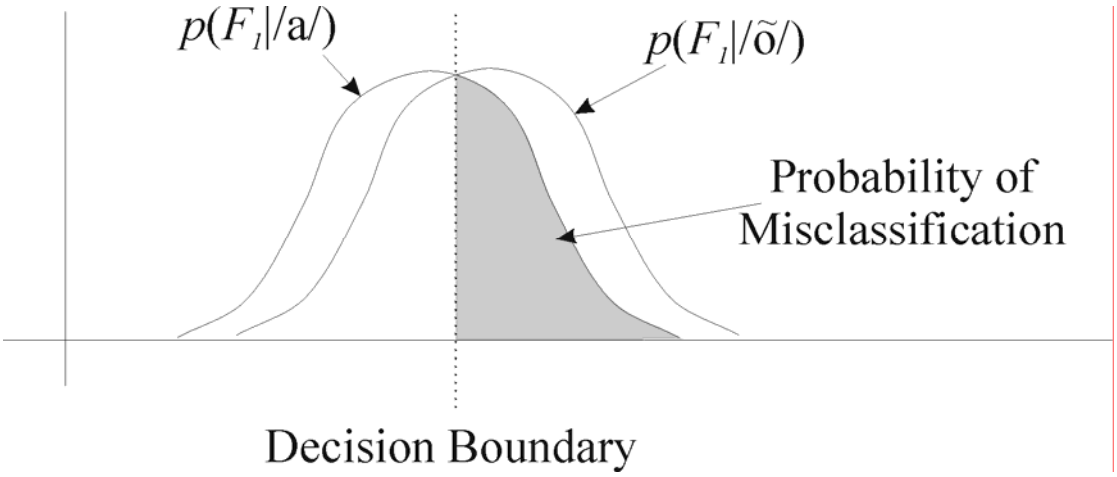
More formally if *x=B* defines the decision boundary then the probability of misclassifying as H_2 given that H_1 is:

$$\int_B^{\infty} p(x \mid H_1)\,dx$$

 © ISVR 2003

RETURNING TO THE EXAMPLE

If one uses just the first formant frequency to distinguish between /õ/ and /a/ then the two pdfs almost completely overlap.



The probability of misclassification is almost 0.5 (i.e. not much better than tossing a coin!)

To improve the ability of the classifier to discriminate between sounds a natural choice is to use more features, i.e. to look at something in addition to the first formant.

The “natural” extension is to consider a classifier that is based on two features – the first and the second formant frequencies.

This is a 2-D classifier.

 © ISVR 2003

A 2-D GAUSSIAN CLASSIFIER

Consider the problem where **x** is a vector of 2 features.

We assume that the features are distributed according to a Gaussian distribution (under each of the hypotheses), so that:

$$p(\mathbf{x} \mid H_k) = \frac{1}{2\pi\sqrt{|R|}} e^{-(\mathbf{x}-\boldsymbol{\mu})^t R^{-1}(\mathbf{x}-\boldsymbol{\mu})/2}$$

$\boldsymbol{\mu}$ is the mean vector: $\boldsymbol{\mu} = E[\mathbf{x}]$

R is the (2×2) covariance matrix: $R = E\left[(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^t\right] = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}$

σ_k^2 is the variance of the kth feature.

$\sigma_{mn} = E\left[(x_m - \mu_m)(x_n - \mu_n)\right]$ is the covariance between the mth and the nth feature.

Thus to define the distribution of a 2-D feature vector one needs to specify 5 values (2 means, 2 variances and 1 covariance).

 © ISVR 2003

A *N*-D GAUSSIAN CLASSIFIER

Consider now a classifier based on a feature vector **x** containing *N* values.

The pdf for such a vector (under a Gaussian assumption) is:

$$p(\mathbf{x} \mid H_k) = \frac{1}{(2\pi)^{N/2}\sqrt{|R|}} e^{-(\mathbf{x}-\boldsymbol{\mu})^t R^{-1}(\mathbf{x}-\boldsymbol{\mu})/2}$$

To define such a distribution one needs to specify $N(N+3)/2N$ parameters, consisting of *N* means, *N* variances, and $N(N-1)/2$ covariances σ_{mn}

One may seek to simplify the model somewhat (with a potential loss of performance), to reduce the number of parameters.

For example if the features are all assumed to be independent then one need only define 2*N* parameters (just the means and variances). The associated pdf is:

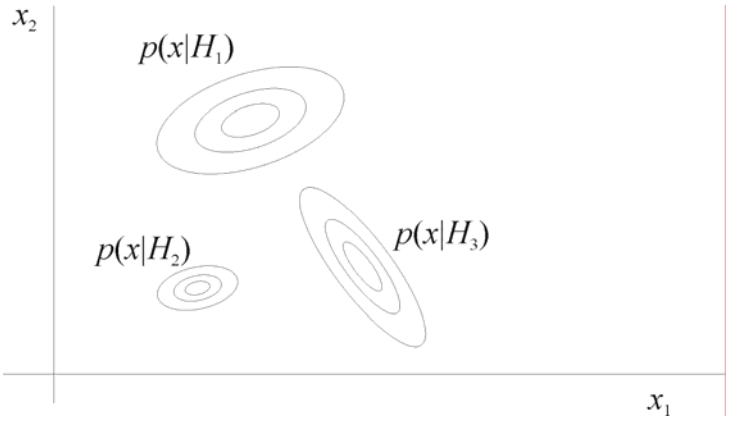
$$p(\mathbf{x} \mid H_k) = \frac{1}{(2\pi)^{N/2}\prod_{k=1}^{N}\sigma_k} e^{-\frac{1}{2}\sum_{k=1}^{N}\left(\frac{x_k-\mu_k}{\sigma_k}\right)^2}$$

 © ISVR 2003

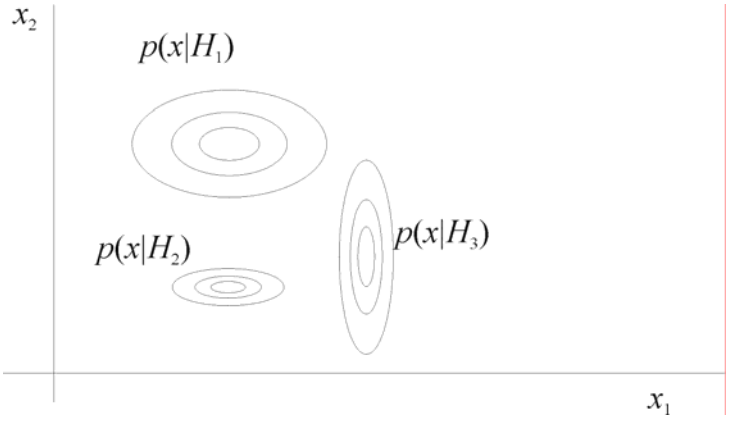
THE EFFECT OF THIS SIMPLIFICATION

By assuming that the features are uncorrelated one reduces the effectiveness of the Gaussian model.

A general (2-D) Gaussian model uses pdfs which have elliptical contours and are arbitrary aligned with the axes.



Whereas simplifying the model only allows one to define pdfs whose principle axes are aligned with the axes of the feature space.

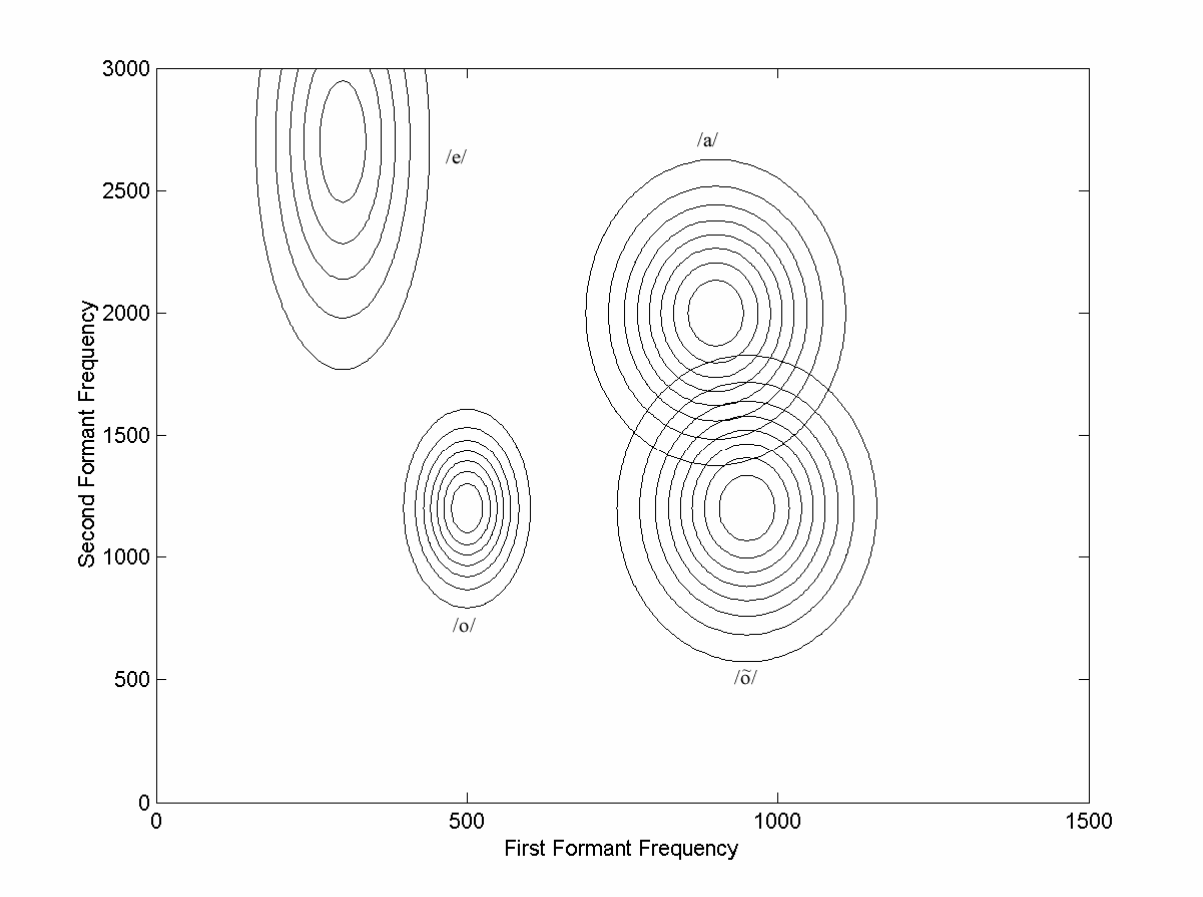


 © ISVR 2003

EXAMPLE USING A 2-D GAUSSIAN CLASSIFIER

Let us consider now the design of our speech classifier for 4 sounds, but let it now use a pair of features: the first and second formant frequencies.

	F_1 Mean (μ_1)	F_1 St. Dev. (σ_1)	F_2 Mean (μ_2)	F_2 St. Dev. (σ_2)
/a/	900	100	2000	300
/e/	300	75	2700	500
/o/	500	50	1200	200
/õ/	950	100	1200	300



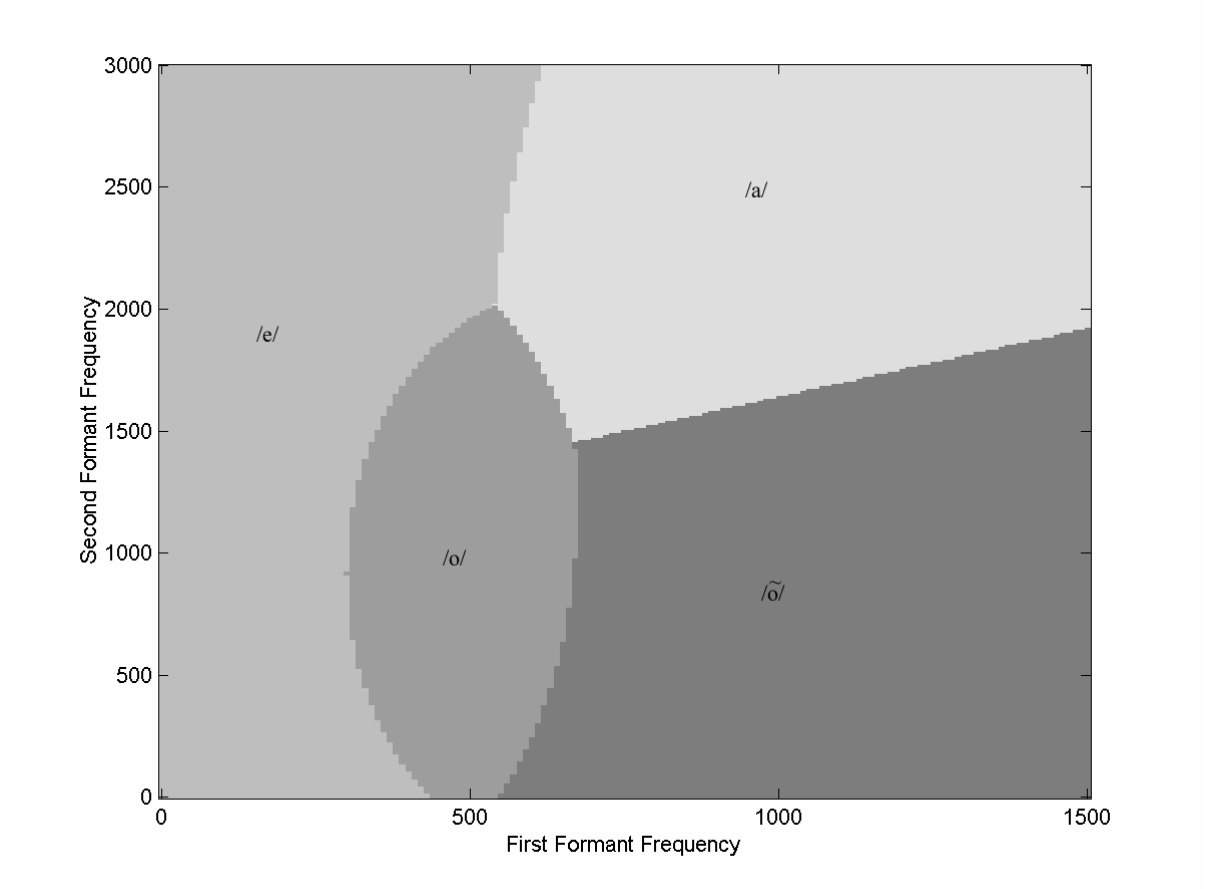
 © ISVR 2003

2-D DECISION SURFACES

Like the 1-D classifier a 2-D classifier can be regarded as constructing a portioning of the (2-D) feature space.

For our example 2-D classifier then the portioning is depicted below.

Each shaded region represents the area in feature space that causes a certain classification to be made.



 © ISVR 2003

TRAINING

In practice one rarely knows the pdfs of the features *a priori*. The most common situation faced by the designer of a classifier is to have a set of example data from each class, called the *training data*.

In principle one can estimate the pdfs for each class of data by using the training data.

Typically there will be at least a modest number of features and estimating pdfs requires large training data sets.

By assuming Gaussianity of the features one simplifies the problem: only the mean and covariances/covariance matrix need be estimated.

So that for each class one computes:

$$\boldsymbol{\mu}_k = \frac{1}{T_k}\sum_{n=1}^{T_k} \mathbf{x}_{n,k} \qquad \Phi_n = \frac{1}{T_k}\sum_{n=1}^{T_k} \left(\mathbf{x}_{n,k} - \boldsymbol{\mu}_k\right)\left(\mathbf{x}_{n,k} - \boldsymbol{\mu}_k\right)^t$$

where T is number of examples of the kth class in the training data and $\mathbf{x}_{n,k}$ are the training data in the kth class.

For each new sample, **x**, one then finds the maximum, over k, of:

$$p\left(\mathbf{x} \mid H_k\right) = \frac{1}{\left(2\pi\right)^{N/2}\sqrt{\left|\Phi_k\right|}} e^{-\left(\mathbf{x}-\boldsymbol{\mu}_k\right)^t \Phi_k^{-1}\left(\mathbf{x}-\boldsymbol{\mu}_k\right)/2}$$

 © ISVR 2003

SIMPLIFICATIONS (COMMON COVARIANCE)

We have already discussed the possibility of assuming independence for all of the features, so that the covariance matrices are diagonal.

An alternative assumption is to assume a common covariance matrix for each class, this means that only a single covariance matrix need be estimated rather than one per class, again reducing the dimension of the problem.

This approximation also simplifies the classifier, since for each class one computes:

$$p(\mathbf{x} \mid H_k) = \frac{1}{(2\pi)^{N/2}\sqrt{|\Phi|}} e^{-(\mathbf{x}-\boldsymbol{\mu}_k)^t \Phi^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)/2}$$

Where Φ is independent of the class *k*.

Thus the leading term in $p(\mathbf{x} \mid H_k)$ is the same for all classes.

The most likely class is thus the one for which $(\mathbf{x}-\boldsymbol{\mu}_k)^t \Phi^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)$ is smallest, this can be regarded as a distance measure.

The covariance matrix, Φ, is assumed to be common to all the classes then it can be estimated from all the training data.

 © ISVR 2003

INTERPRETATION OF COMMON COVARIANCE MODEL

The covariance matrix, Φ, can be factorised so that $\Phi^{-1} = C^t C$, which means that the distance measure, $d(\mathbf{x}, \boldsymbol{\mu}_k)$, becomes:

$$d(\mathbf{x}, \boldsymbol{\mu}_k) = (\mathbf{x} - \boldsymbol{\mu}_k)^t \Phi^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) = (\mathbf{x} - \boldsymbol{\mu}_k)^t C^t C (\mathbf{x} - \boldsymbol{\mu}_k)$$

We can define a transformed feature space and transformed mean vectors:

$$\mathbf{v} = C\mathbf{x} \qquad \hat{\boldsymbol{\mu}}_k = C\boldsymbol{\mu}_k = E[\mathbf{v}]$$

The distance measure can then be written as:

$$d(\mathbf{x}, \boldsymbol{\mu}_k) = (\mathbf{v} - \hat{\boldsymbol{\mu}}_k)^t (\mathbf{v} - \hat{\boldsymbol{\mu}}_k)$$

The distance measure $d(\mathbf{x}, \boldsymbol{\mu}_k)$ is thus the Euclidean distance between the two vectors $\mathbf{v}$ and $\hat{\boldsymbol{\mu}}_k$. Note this classification only depends on the distances to the class means.

The transformation $\mathbf{v} = C\mathbf{x}$ creates a new feature vector, **v**, in which the features are uncorrelated and have unit variance, *i.e.*

$$E[\mathbf{v}\mathbf{v}^t] = I$$

In this new feature space one classifies a vector according to which mean vector is closest (in the Euclidian sense).

 © ISVR 2003

A NON-PARAMETRIC CLASSIFIER

One of the most common forms of non-parametric classifier is the *k*-Nearest Neighbour (*k*NN) classifier.

For illustrative purposes consider the simple Nearest Neighbour classifier (k=1).

Nearest Neighbour classifiers employ a distance metric, $d(.,.)$, *e.g.* the Euclidian distance.

The distance between the incoming feature vector **x** and each sample in the training set is computed.

The algorithm classifies **x** as being in the class associated with the nearest training sample.

In the general *k*NN scheme one identifies the k nearest training sample and a "voting" strategy is used to pick the class.

*k*NN is simple to implement and can give quite good performance in cases where the features have a non-Gaussian distribution.

 © ISVR 2003