# Multivariate statistical methods and data mining in particle physics

Glen Cowan

RHUL Physics

**www.pp.rhul.ac.uk/~cowan**

Academic Training Lectures

CERN

16–19 June, 2008

# Outline

Statement of the problem

Some general considerations

Brief review of statistical formalism

Multivariate classifiers:

- Linear discriminant function
- Neural networks
- Naive Bayes classifier
- Kernel-based methods
- *k*-Nearest-Neighbour
- Decision trees
- Support Vector Machines

Other multivariate problems:

- Multivariate regression
- Unsupervised learning

# Resources

Books:

C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006

T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning*, Springer, 2001

R. Duda, P. Hart, D. Stork, *Pattern Classification*, 2nd ed., Wiley, 2001

A. Webb, *Statistical Pattern Recognition*, 2nd ed., Wiley, 2002

Materials from some recent meetings:

PHYSTAT conference series (2002, 2003, 2005, 2007,...) see
`www.phystat.org`

Caltech workshop on multivariate analysis, 11 February, 2008
`indico.cern.ch/conferenceDisplay.py?confId=27385`

SLAC Lectures on Machine Learning by Ilya Narsky (2006)
`www-group.slac.stanford.edu/sluo/Lectures/Stat2006_Lectures.html`

# Software

**`TMVA`**, Höcker, Stelzer, Tegenfeldt, Voss, Voss, physics/0703039

From **`tmva.sourceforge.net`**, also distributed with ROOT

Variety of classifiers

Good manual

**`StatPatternRecognition`**, I. Narsky, physics/0507143

Further info from **`www.hep.caltech.edu/~narsky/spr.html`**

Also wide variety of methods, many complementary to **`TMVA`**

Currently appears project no longer to be supported

# Motivation

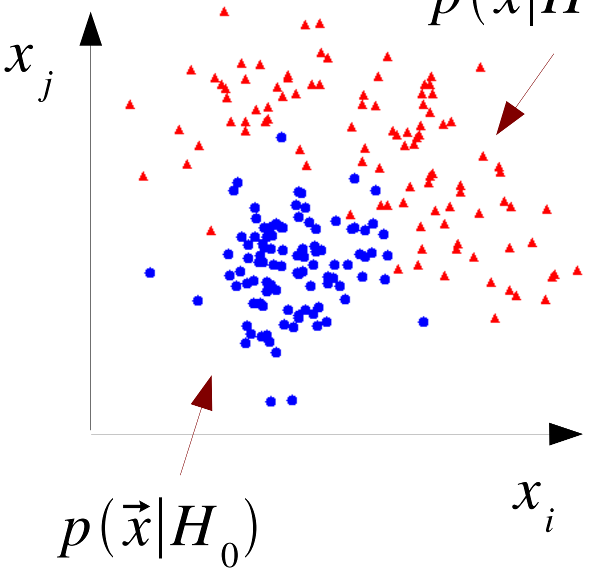
Suppose for each event we measure a set of numbers $\vec{x} = (x_1, \ldots, x_n)$

$x_1$ = jet $p_T$

$x_2$ = missing energy

$x_3$ = particle i.d. measure, ...

$\vec{x}$ follows some $n$-dimensional joint probability density, which depends on the type of event produced, i.e., was it $pp \to t\bar{t}$, $pp \to \tilde{g}\tilde{g}, \ldots$



E.g. hypotheses (class labels) $H_0$, $H_1$, ...

Often simply "signal", "background"

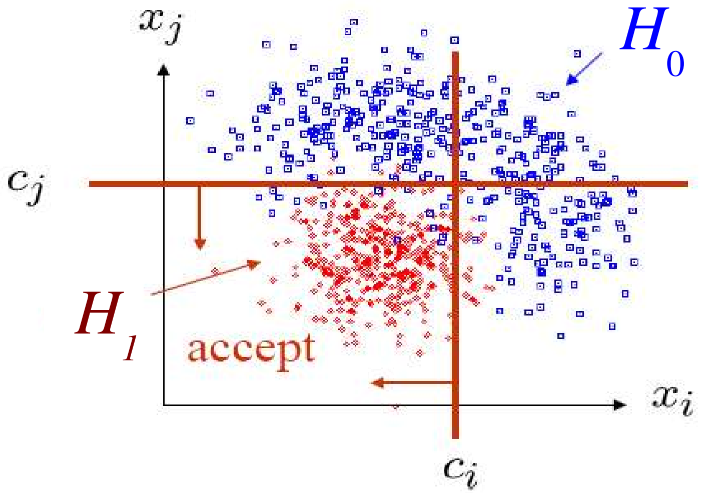
We want to separate (classify) the event types in a way that exploits the information carried in many variables.

# Finding an optimal decision boundary

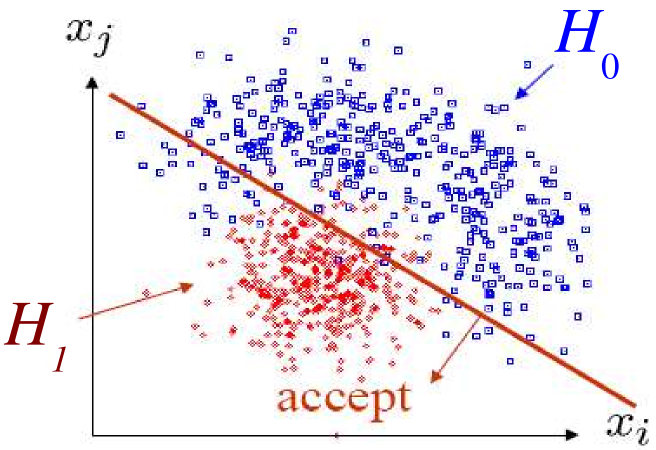
Maybe select events with “cuts”:

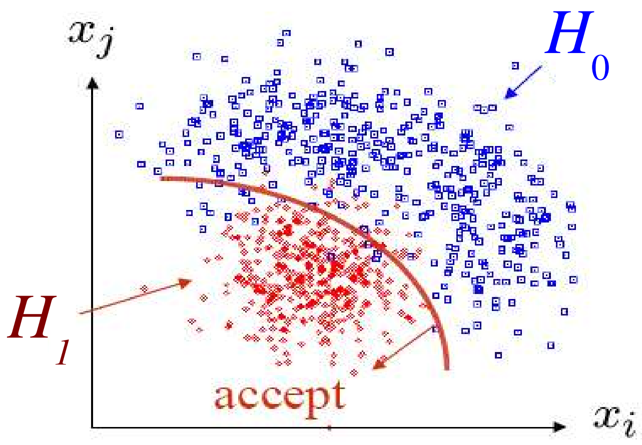
$$x_i < c_i$$

$$x_j < c_j$$



Or maybe use some other type of decision boundary:





Goal of multivariate analysis is to do this in an “optimal” way.

# General considerations

In all multivariate analyses we must consider e.g.

Choice of variables to use

Functional form of decision boundary (type of classifier)

Computational issues

Trade-off between sensitivity and complexity

Trade-off between statistical and systematic uncertainty

Our choices can depend on goals of the analysis, e.g.,

Event selection for further study

Searches for new event types

# Probability – quick review

Frequentist (*A* = outcome of repeatable observation):

$$P(A) = \lim_{n\to\infty} \frac{\text{outcome is } A}{n}$$

Subjective (*A* = hypothesis):

$$P(A) = \text{degree of belief that } A \text{ is true}$$

Conditional probability:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Bayes' theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{P(B|A)P(A)}{\Sigma_i\, P(B|A_i)P(A_i)}$$

# Test statistics

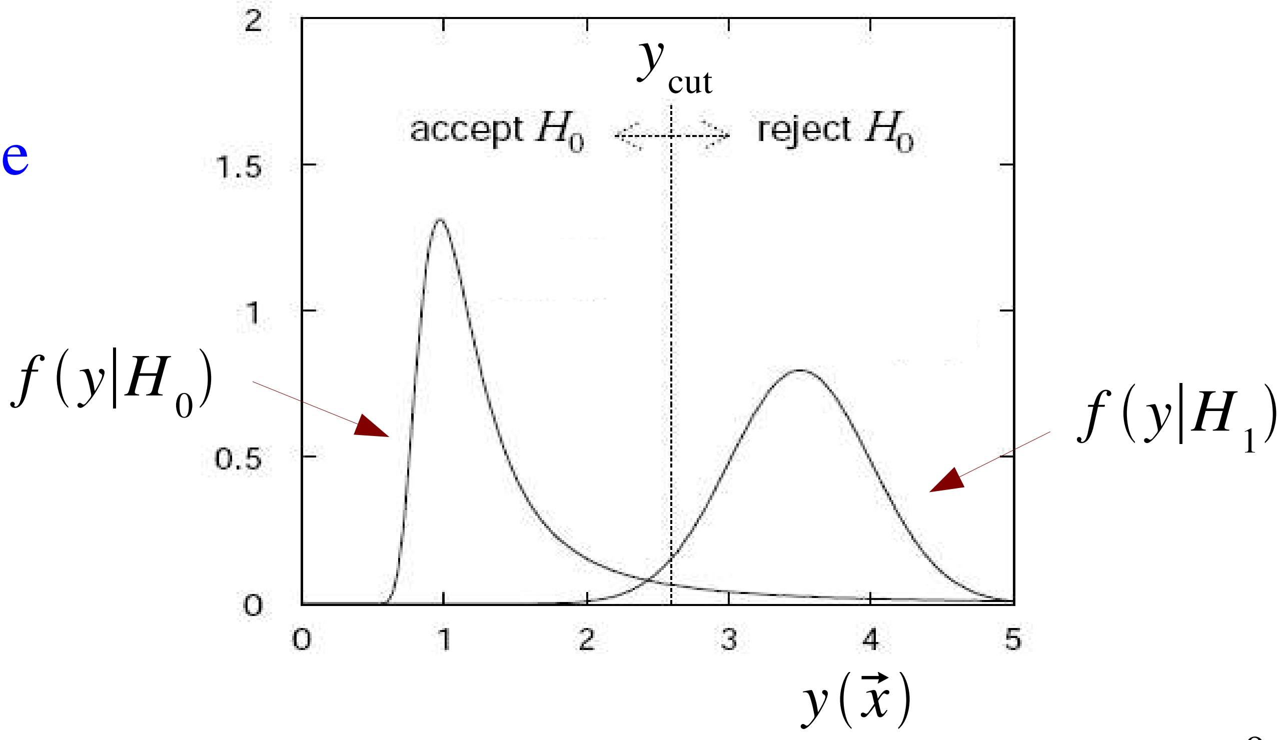
The decision boundary is a surface in the *n*-dimensional space of input variables, e.g., $y(\vec{x}) = \text{const}$.

We can treat the *y*(*x*) as a scalar test statistic or discriminating function, and try to define this function so that its distribution has the maximum possible separation between the event types:

The decision boundary is now effectively a single cut on *y*(***x***), dividing ***x***-space into two regions:

$R_0$ (accept $H_0$)

$R_1$ (reject $H_0$)



# Classification viewed as a statistical test

Probability to reject $H_0$ if it is true (type I error): $\alpha = \int_{R_1} f(y|H_0)\,dy$

$\alpha$ = significance level, size of test, false discovery rate

Probability to accept $H_0$ if $H_1$ is true (type II error): $\beta = \int_{R_0} f(y|H_1)\,dy$

$1 - \beta$ = power of test with respect to $H_1$

Equivalently if e.g. $H_0$ = background, $H_1$ = signal, use efficiencies:

$$\varepsilon_{\mathrm{s}} = \int_{R_1} f(y|H_1)\,dy = 1 - \beta = \text{power} \qquad \varepsilon_{\mathrm{b}} = \int_{R_0} f(y|H_0)\,dy = 1 - \alpha$$

# Purity / misclassification rate

Consider the probability that an event assigned to a certain category is classified correctly (i.e., the purity).

Use Bayes' theorem:

Here $R_1$ is signal region

prior probability

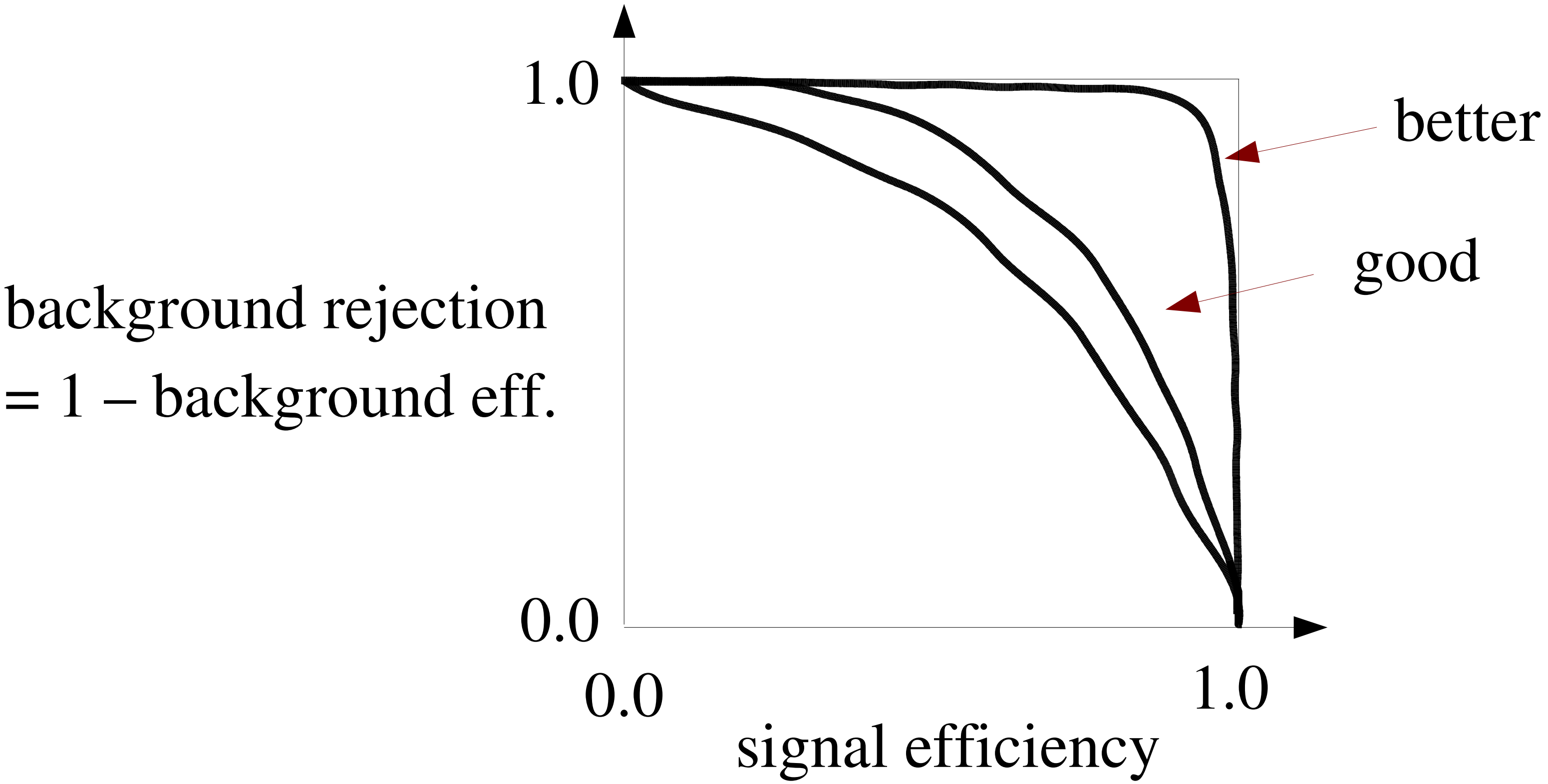
$$P(\mathrm{s}|\mathbf{x} \in R_1) = \frac{P(\mathbf{x} \in R_1|\mathrm{s})P(\mathrm{s})}{P(\mathbf{x} \in R_1|\mathrm{s})P(\mathrm{s}) + P(\mathbf{x} \in R_1|\mathrm{b})P(\mathrm{b})}$$

posterior probability

N.B. purity depends on the prior probabilities for an event to be signal or background (~s, b cross sections).

# ROC curve

We can characterize the quality of a classification procedure with the receiver operating characteristic (ROC curve)



Independent of prior probabilities.

Area under ROC curve can be used as a measure of quality.

# Constructing a test statistic

The Neyman-Pearson lemma states: to obtain the highest background rejection for a given signal efficiency (highest power for a given significance level), choose the acceptance region for signal such that

$$\frac{p(\vec{x}|\mathrm{s})}{p(\vec{x}|\mathrm{b})} > c$$

where $c$ is a constant that determines the signal efficiency.

Equivalently, the optimal discriminating function is given by the likelihood ratio:

$$y(\vec{x}) = \frac{p(\vec{x}|\mathrm{s})}{p(\vec{x}|\mathrm{b})}$$

N.B. any monotonic function of this is just as good.

# Bayes optimal analysis

From Bayes' theorem we can compute the posterior odds:

$$\underbrace{\frac{p(\mathrm{s}|\vec{x})}{p(\mathrm{b}|\vec{x})}}_{\text{posterior odds}} = \underbrace{\frac{p(\vec{x}|\mathrm{s})}{p(\vec{x}|\mathrm{b})}}_{\text{likelihood ratio}} \underbrace{\frac{p(\mathrm{s})}{p(\mathrm{b})}}_{\text{prior odds}}$$

which is proportional to the likelihood ratio.

So placing a cut on the likelihood ratio is equivalent to ensuring a minimum posterior odds ratio for the selected sample.

# Purity vs. efficiency trade-off

The actual choice of signal efficiency (and thus purity) will depend on goal of analysis, e.g.,

Trigger selection (high efficiency)

Event sample used for precision measurement (high purity)

Measurement of signal cross section: maximize $s/\sqrt{s+b}$

Discovery of signal: maximize expected significance ~ $s/\sqrt{b}$

# Neyman-Pearson doesn't always help

The problem is that we usually don't have explicit formulae for the pdfs $p(\boldsymbol{x}|\mathrm{s})$, $p(\boldsymbol{x}|\mathrm{b})$, so for a given $\boldsymbol{x}$ we can't evaluate the likelihood ratio.

Instead we may have Monte Carlo models for signal and background processes, so we can produce simulated data:

generate $\vec{x} \sim p(\vec{x}|\mathrm{s}) \longrightarrow \vec{x}_{1,} \ldots, \vec{x}_{N_s}$

generate $\vec{x} \sim p(\vec{x}|\mathrm{b}) \longrightarrow \vec{x}_{1,} \ldots, \vec{x}_{N_b}$

"training data"
events of known type

Naive try: enter each (s,b) event into an $n$-dimensional histogram, use e.g. $M$ bins for each of the $n$ dimensions, total of $M^n$ cells.

$n$ is potentially large → prohibitively large number of cells to populate, can't generate enough training data.

# Strategies for event classification

A compromise solution is to make an Ansatz for the form of the test statistic $y(\boldsymbol{x})$ with fewer parameters; determine them (using MC) to give best discrimination between signal and background.

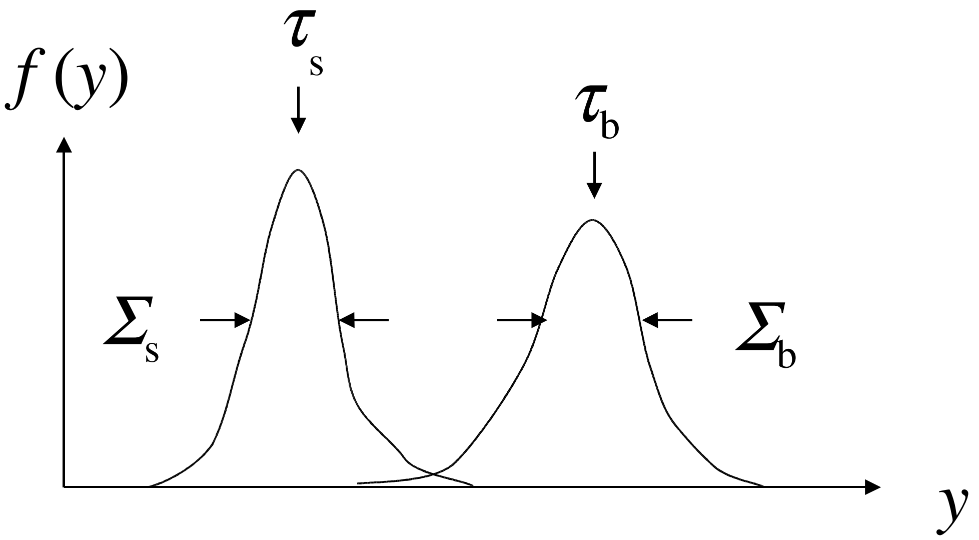
Alternatively, try to estimate the probability densities $p(\boldsymbol{x}|\mathrm{s})$, $p(\boldsymbol{x}|\mathrm{b})$ and use the estimated pdfs to compute the likelihood ratio at the desired $\boldsymbol{x}$ values (for real data).

# Linear test statistic

Ansatz: $$y(\vec{x}) = \sum_{i=1}^{n} w_i x_i = \vec{w}^T \vec{x}$$

Choose the parameters $w_1$, ..., w$_n$ so that the pdfs $f(y|\mathrm{s}), f(y|\mathrm{b})$ have maximum 'separation'. We want:

large distance between mean values, small widths



→ Fisher: maximize $$J(\vec{w}) = \frac{(\tau_\mathrm{s} - \tau_\mathrm{b})^2}{\Sigma_\mathrm{s}^2 + \Sigma_\mathrm{b}^2}$$

# Coefficients for maximum separation

We have $$(\mu_k)_i = \int x_i \, p(\vec{x}|H_k) \, d\vec{x}$$

mean, covariance of $\boldsymbol{x}$

$$(V_k)_{ij} = \int (x-\mu_k)_i (x-\mu_k)_j \, p(\vec{x}|H_k) \, d\vec{x}$$

where $k = 0, 1$ (hypothesis)

and $i, j = 1, \ldots, n$ (component of $\boldsymbol{x}$)

For the mean and variance of $y(\vec{x})$ we find

$$\tau_k = \int y(\vec{x}) \, p(\vec{x}|H_k) \, d\vec{x} = \vec{w}^T \vec{\mu}_k$$

$$\Sigma_k^2 = \int (y(\vec{x}) - \tau_k)^s \, p(\vec{x}|H_k) \, d\vec{x} = \vec{w}^T V_k \vec{w}$$

# Determining the coefficients *w*

The numerator of $J(\boldsymbol{w})$ is

$$(\tau_0-\tau_1)^2 = \sum_{i,j=1}^{n} w_i w_j (\mu_0-\mu_1)_i (\mu_0-\mu_1)_j$$

'between' classes

$$= \sum_{i,j=1}^{n} w_i w_j B_{ij} = \vec{w}^T B \vec{w}$$

and the denominator is

'within' classes

$$\Sigma_0^2 + \Sigma_1^2 = \sum_{i,j=1}^{n} w_i w_j (V_0+V_1)_{ij} = \vec{w}^T W \vec{w}$$

→ maximize $$J(\vec{w}) = \frac{\vec{w}^T B \vec{w}}{\vec{w}^T W \vec{w}} = \frac{\text{separation between classes}}{\text{separation within classes}}$$

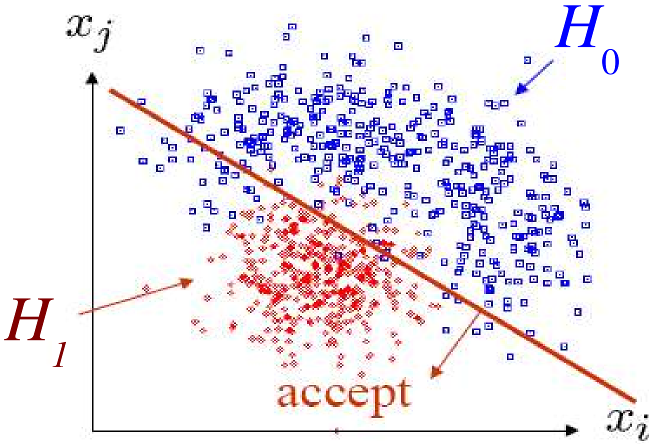
# Fisher discriminant function

Setting $\frac{\partial J}{\partial w_i} = 0$ gives Fisher's linear discriminant function:

$$\boxed{y(\vec{x}) = \vec{w}^T \vec{x} \qquad \text{with}\ \vec{w} \propto W^{-1}(\vec{\mu}_0 - \vec{\mu}_1)}$$

Gives linear decision boundary.

Projection of points in direction of decision boundary gives maximum separation.



# Comment on least squares

We obtain equivalent separation between the classes if we multiply the $w_i$ by a common scale factor and add an offset $w_0$:

$$y(\vec{x}) = w_0 + \sum_{i=1}^{n} w_i x_i$$

Thus we can fix the mean values $\tau_0$ and $\tau_1$ for the two classes to arbitrary values, e.g., 0 and 1.

Then maximizing $J(\vec{w}) = (\tau_0 - \tau_1)^2 / (\Sigma_0^2 + \Sigma_1^2)$ means minimizing

$$\Sigma_0^2 + \Sigma_1^2 = E_0[(y - \tau_0)^2] + E_1[(y - \tau_1)^2]$$

Maximizing Fisher's $J(\mathbf{w})$ → 'least squares'

Estimate expectation values with averages of training (e.g. MC) data:

$$E_k[(y - \tau_k)^2] \rightarrow \frac{1}{N_k} \sum_{i=1}^{N_k} (y_i - \tau_k)^2$$

# Fisher discriminant for Gaussian data

Suppose $f(\boldsymbol{x}|H_k)$ is a multivariate Gaussian with mean values

$$E_0[\vec{x}]=\vec{\mu}_0 \text{ for } H_0 \qquad\qquad E_1[\vec{x}]=\vec{\mu}_1 \text{ for } H_1$$

and covariance matrices $V_0 = V_1 = V$ for both. We can write the Fisher's discriminant function (with an offset) is

$$y(\vec{x})=w_0+(\vec{\mu}_0-\vec{\mu}_1)V^{-1}\vec{x}$$

The likelihood ratio is thus

$$\frac{p(\vec{x}|H_0)}{p(\vec{x}|H_1)}=\exp[-\frac{1}{2}(\vec{x}-\vec{\mu}_0)^T V^{-1}+\frac{1}{2}(\vec{x}-\vec{\mu})_1^T V^{-1}(\vec{x}-\vec{\mu}_1)]$$

$$=e^y$$

# Fisher for Gaussian data (2)

That is, $y(\boldsymbol{x})$ is a monotonic function of the likelihood ratio, so for this case the Fisher discriminant is equivalent to using the likelihood ratio, and is therefore optimal.

For non-Gaussian data this no longer holds, but linear discriminant function may be simplest practical solution.

Often try to transform data so as to better approximate Gaussian before constructing Fisher discrimimant.

# Fisher and Gaussian data (3)

Multivariate Gaussian data with equal covariance matrices also gives a simple expression for posterior probabilities, e.g.,

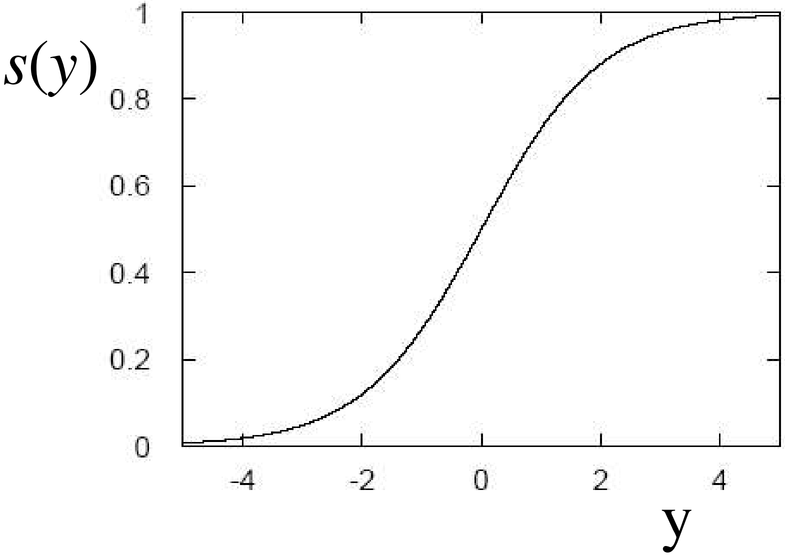
$$P(H_0|\vec{x}) = \frac{p(\vec{x}|H_0)\,P(H_0)}{p(\vec{x}|H_0)\,P(H_0) + p(\vec{x}|H_1)\,P(H_1)}$$

For Gaussian ***x*** and a particular choice of the offset $w_0$ this becomes:

$$P(H_0|\vec{x}) = \frac{1}{1+e^{-y(\vec{x})}} \equiv s(y(\vec{x}))$$

which is the logistic sigmoid function:

(We will use this later in connection with Neural Networks.)



# Transformation of inputs

If the data are not Gaussian with equal covariance, a linear decision boundary is not optimal. But we can try to subject the data to a transformation

$$\varphi_1(\vec{x}), \ldots, \varphi_m(\vec{x})$$

and then treat the $\phi_i$ as the new input variables. This is often called "feature space" and the $\phi_i$ are "basis functions". The basis functions can be fixed or can contain adjustable parameters which we optimize with training data (cf. neural networks).

In other cases we will see that the basis functions only enter as dot products

$$\vec{\varphi}^T(\vec{x}_i)\vec{\varphi}(\vec{x}_j) = K(\vec{x}_i, \vec{x}_j)$$

and thus we will only need the "kernel function" $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$

# Lecture 1 Summary

The optimal solution to our classification problem should be to use the likelihood ratio as the test variable – unfortunately not possible.

So far we have seen a simple linear ansatz for the test statistic, which is optimal only for Gaussian data (with equal covariance).

If the data are not Gaussian, we can transform to “feature space”

$$\varphi_1(\vec{x}), \ldots, \varphi_m(\vec{x})$$

Next time we will consider this to obtain nonlinear test statistics such as neural networks.