Introduction to Statistics − Day 2

Lecture 1

- Probability
- Random variables, probability densities, etc.
- Brief catalogue of probability densities

→ Lecture 2

- The Monte Carlo method
- Statistical tests
- Fisher discriminants, neural networks, etc.

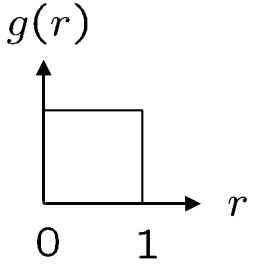
Lecture 3

- Goodness-of-fit tests
- Parameter estimation
- Maximum likelihood and least squares
- Interval estimation (setting limits)

The Monte Carlo method

What it is: a numerical technique for calculating probabilities and related quantities using sequences of random numbers.

The usual steps:



(1) Generate sequence $r_1, r_2, ..., r_m$ uniform in [0, 1].

(2) Use this to produce another sequence $x_1, x_2, ..., x_n$ distributed according to some pdf $f(x)$ in which we're interested (x can be a vector).

(3) Use the x values to estimate some property of $f(x)$, e.g., fraction of x values with $a < x < b$ gives $\int_a^b f(x)\,dx$.

→ MC calculation = integration (at least formally)

MC generated values = 'simulated data'

→ use for testing statistical procedures

Random number generators

Goal: generate uniformly distributed values in [0, 1].

Toss coin for e.g. 32 bit number... (too tiring).

→ 'random number generator'

= computer algorithm to generate $r_1, r_2, \ldots, r_n$.

Example: multiplicative linear congruential generator (MLCG)

$n_{i+1} = (a\, n_i) \bmod m$, where

n_i = integer

a = multiplier

m = modulus

n_0 = seed (initial value)

N.B. mod = modulus (remainder), e.g. 27 mod 5 = 2.

This rule produces a sequence of numbers $n_0, n_1, \ldots$

Random number generators (2)

The sequence is (unfortunately) periodic!

Example (see Brandt Ch 4): $a = 3$, $m = 7$, $n_0 = 1$

$$n_1 = (3 \cdot 1) \bmod 7 = 3$$
$$n_2 = (3 \cdot 3) \bmod 7 = 2$$
$$n_3 = (3 \cdot 2) \bmod 7 = 6$$
$$n_4 = (3 \cdot 6) \bmod 7 = 4$$
$$n_5 = (3 \cdot 4) \bmod 7 = 5$$
$$n_6 = (3 \cdot 5) \bmod 7 = 1 \quad \leftarrow \text{ sequence repeats}$$

Choose a, m to obtain long period (maximum = $m - 1$); m usually close to the largest integer that can represented in the computer.

Only use a subset of a single period of the sequence.

Random number generators (3)

$r_i = n_i/m$ are in [0, 1] but are they 'random'?

Choose a, m so that the r_i pass various tests of randomness:

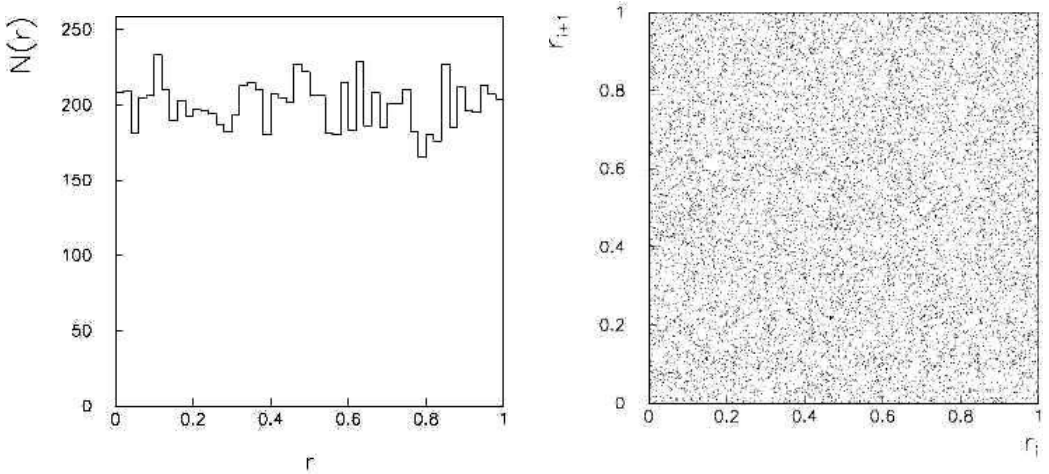
uniform distribution in [0, 1],

all values independent (no correlations between pairs),

e.g. L'Ecuyer, Commun. ACM **31** (1988) 742 suggests

$a = 40692$

$m = 2147483399$

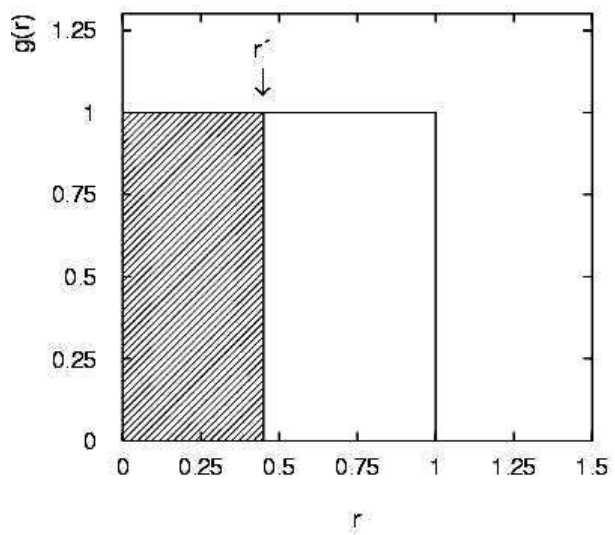


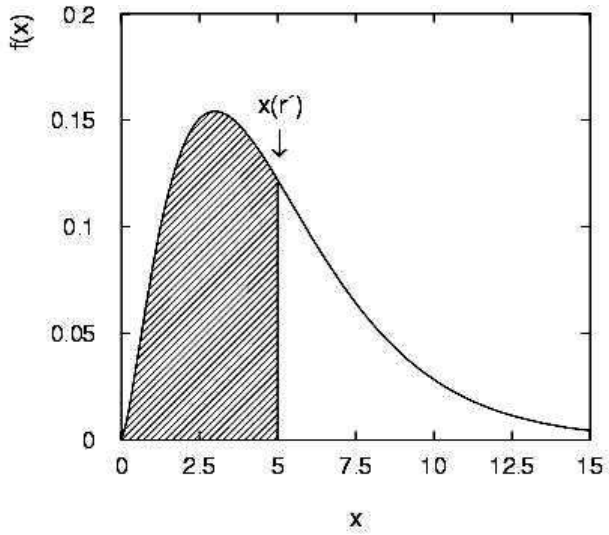
Far better algorithms available, e.g. `RANMAR`, period $\approx 2 \times 10^{43}$.

See F. James, Comp. Phys. Comm. 60 (1990) 111; Brandt Ch. 4

The transformation method

Given $r_1, r_2, \ldots, r_n$ uniform in [0, 1], find $x_1, x_2, \ldots, x_n$
that follow $f(x)$ by finding a suitable transformation $x(r)$.





Require: $P(r \leq r') = P(x \leq x(r'))$

i.e. $$\int_{-\infty}^{r'} g(r)\,dr = r' = \int_{-\infty}^{x(r')} f(x')\,dx' = F(x(r'))$$

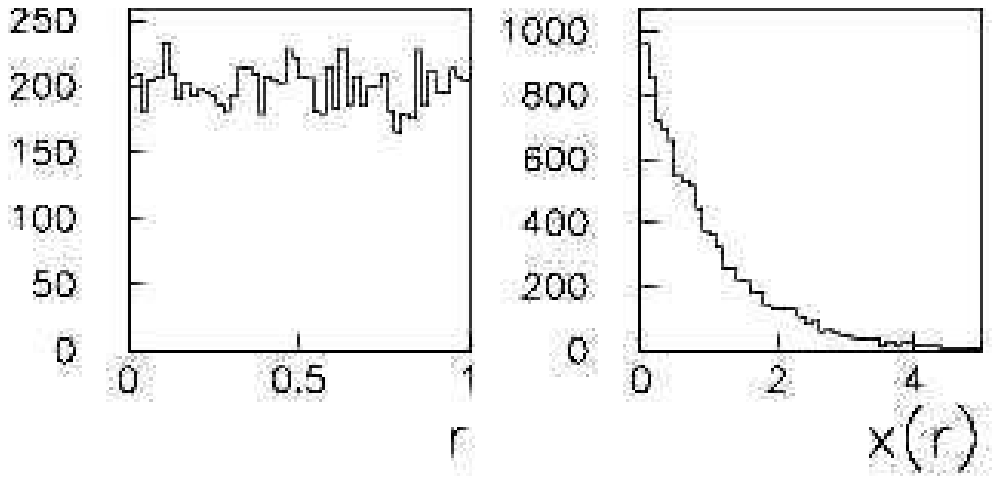
That is, set $F(x) = r$ and solve for $x(r)$.

Example of the transformation method

Exponential pdf: $f(x;\xi) = \frac{1}{\xi} e^{-x/\xi} \quad (x \geq 0)$

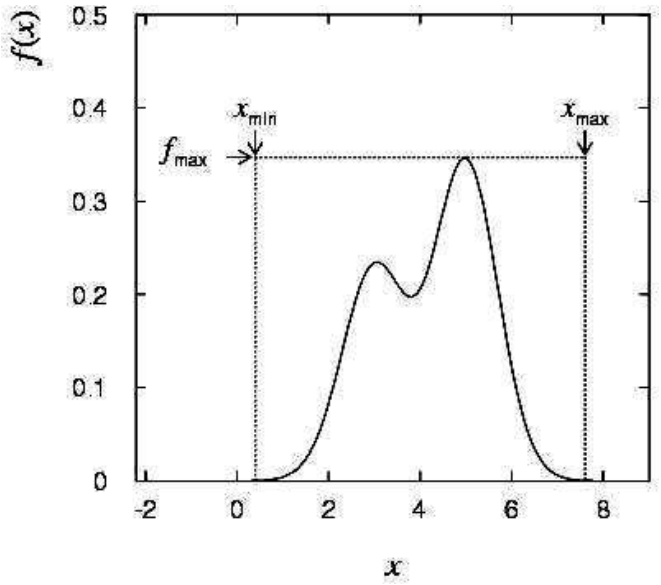
Set $\int_0^x \frac{1}{\xi} e^{-x'/\xi}\, dx' = r$ and solve for $x\,(r)$.

$\rightarrow \quad x(r) = -\xi \ln(1-r) \qquad (\, x(r) = -\xi \ln r$ works too.)



The acceptance-rejection method

Enclose the pdf in a box:



(1) Generate a random number x, uniform in $[x_{\min}, x_{\max}]$, i.e.
$x = x_{\min} + r_1(x_{\max} - x_{\min})$, r_1 is uniform in [0,1].

(2) Generate a 2nd independent random number u uniformly distributed between 0 and $f_{\max}$, i.e. $u = r_2 f_{\max}$.

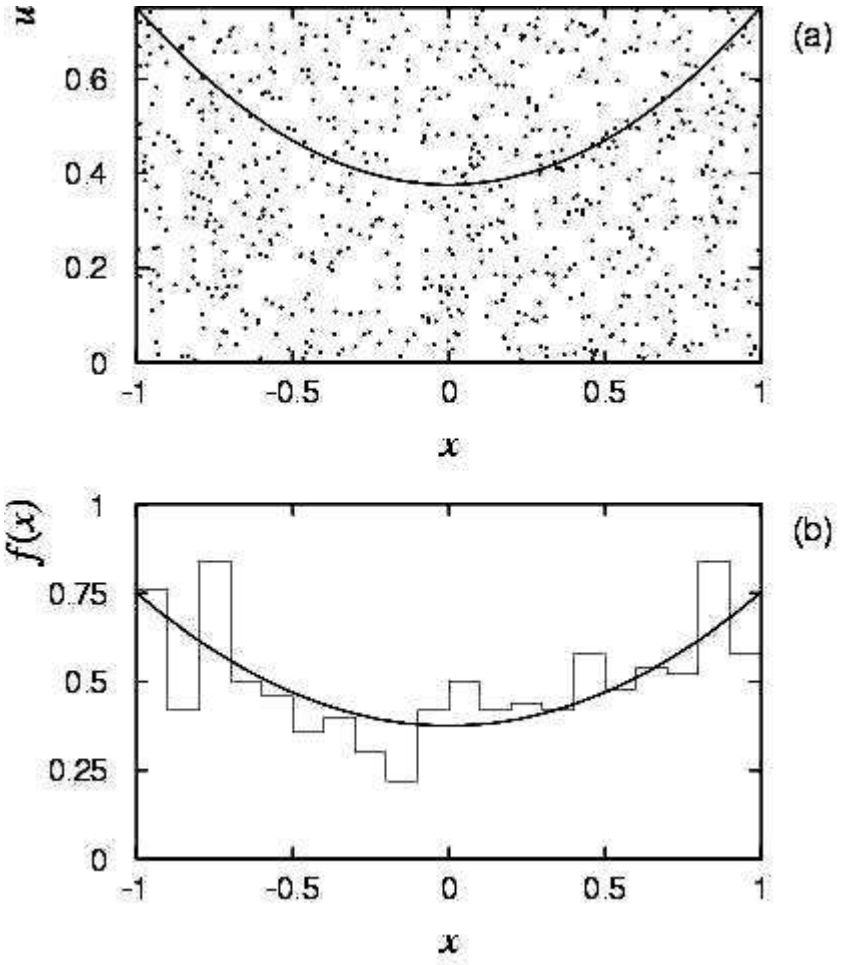
(3) If $u < f(x)$, then accept x. If not, reject x and repeat.

Example with acceptance-rejection method

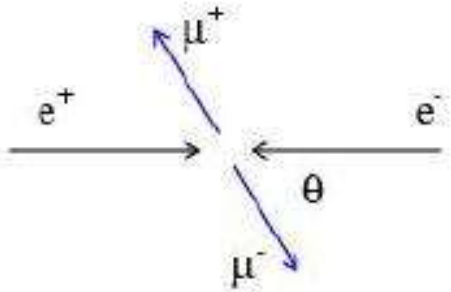
$$f(x) = \frac{3}{8}(1 + x^2)$$

$$(-1 \leq x \leq 1)$$

If dot below curve, use x value in histogram.



Monte Carlo event generators



Simple example: $e^+e^- \to \mu^+\mu^-$

Generate $\cos\theta$ and ϕ:

$$f(\cos\theta; A_{FB}) \propto (1 + \frac{8}{3} A_{FB} \cos\theta + \cos^2\theta) ,$$

$$g(\phi) = \frac{1}{2\pi} \quad (0 \leq \phi \leq 2\pi)$$

Less simple: 'event generators' for a variety of reactions:

$e^+e^- \to \mu^+\mu^-$, hadrons, ...

$pp \to$ hadrons, D-Y, SUSY,...

e.g. PYTHIA, HERWIG, ISAJET...

Output = 'events', i.e., for each event we get a list of generated particles and their momentum vectors, types, etc.

Monte Carlo detector simulation

Takes as input the particle list and momenta from generator.

Simulates detector response:

multiple Coulomb scattering (generate scattering angle),
particle decays (generate lifetime),
ionization energy loss (generate Δ),
electromagnetic, hadronic showers,
production of signals, electronics response, ...

Output = simulated raw data → input to reconstruction software:

track finding, fitting, etc.

Predict what you should see at 'detector level' given a certain hypothesis for 'generator level'. Compare with the real data.

Estimate 'efficiencies' = #events found / # events generated.

Programming package: `GEANT`

Statistical tests (in a particle physics context)

Suppose the result of a measurement for an individual event is a collection of numbers $\vec{x} = (x_1, \ldots, x_n)$

x_1 = number of muons,

x_2 = mean p_t of jets,

x_3 = missing energy, ...

$\vec{x}$ follows some n-dimensional joint pdf, which depends on the type of event produced, i.e., was it

$$pp \to t\bar{t}\ , \qquad pp \to \tilde{g}\tilde{g}\ , \ldots$$

For each reaction we consider we will have a hypothesis for the pdf of $\vec{x}$, e.g., $f(\vec{x}|H_0)$, $f(\vec{x}|H_1)$, etc.

Often call H_0 the signal hypothesis (the event type we want); H_1, H_2, ... are background hypotheses.

Selecting events

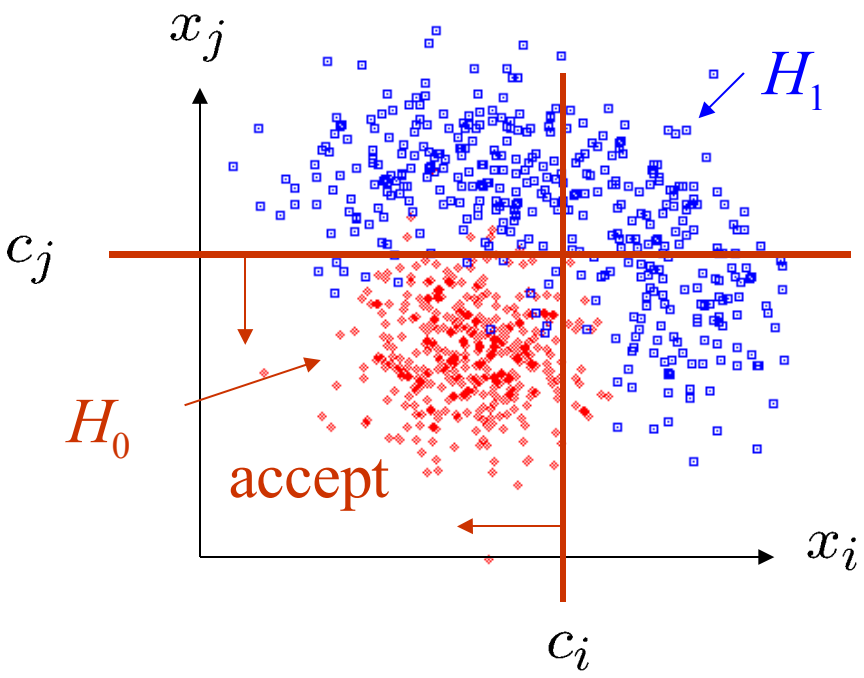
Suppose we have a data sample with two kinds of events, corresponding to hypotheses H_0 and H_1 and we want to select those of type H_0.

Each event is a point in $\vec{x}$ space. What 'decision boundary' should we use to accept/reject events as belonging to event type H_0?

Perhaps select events with 'cuts':

$$x_i < c_i$$

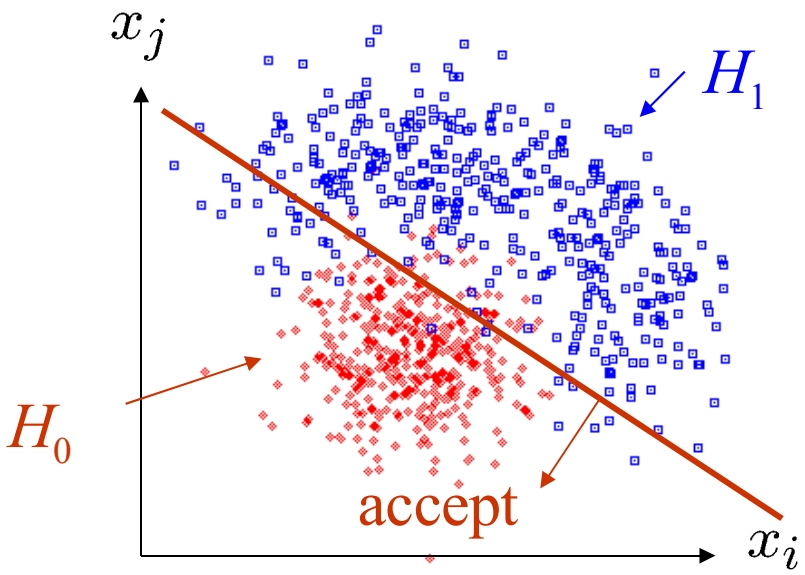
$$x_j < c_j$$



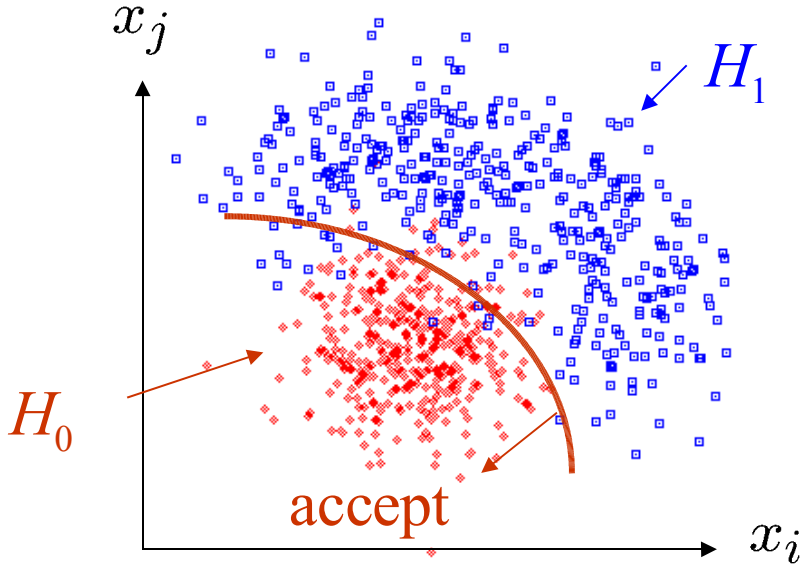
Other ways to select events

Or maybe use some other sort of decision boundary:

linear



or nonlinear



How can we do this in an 'optimal' way?

What are the difficulties in a high-dimensional space?

Test statistics

Construct a 'test statistic' of lower dimension (e.g. scalar)

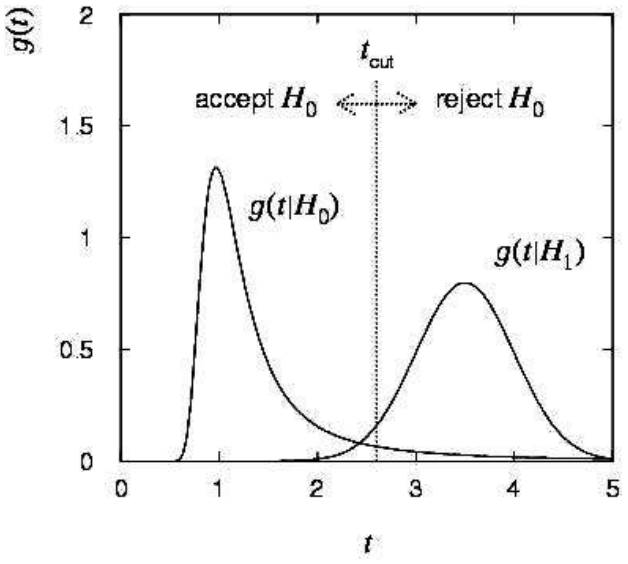
$$t(x_1, \ldots, x_n)$$

Try to compactify data without losing ability to discriminate between hypotheses.

We can work out the pdfs $g(t|H_0),\ g(t|H_1),\ \ldots$

Decision boundary is now a single 'cut' on t.

This effectively divides the sample space into two regions, where we accept or reject H_0.

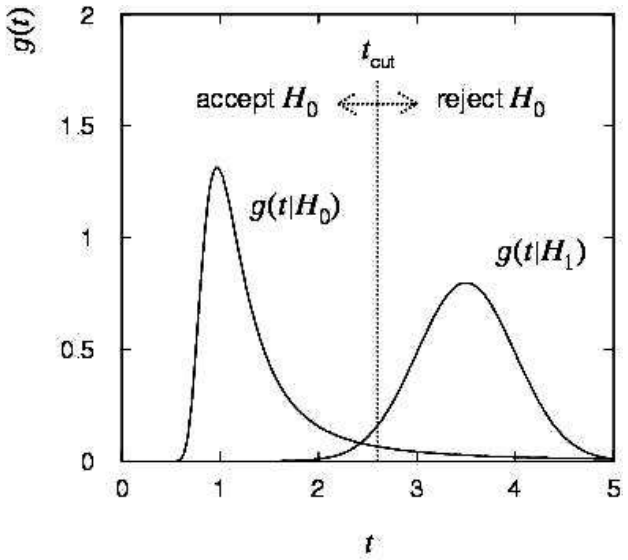


Significance level and power of a test

Probability to reject H_0 if it is true (error of the 1st kind):

$$\alpha = \int_{t_{\text{cut}}}^{\infty} g(t|H_0)\, dt$$

(significance level)



Probability to accept H_0 if H_1 is true (error of the 2nd kind):

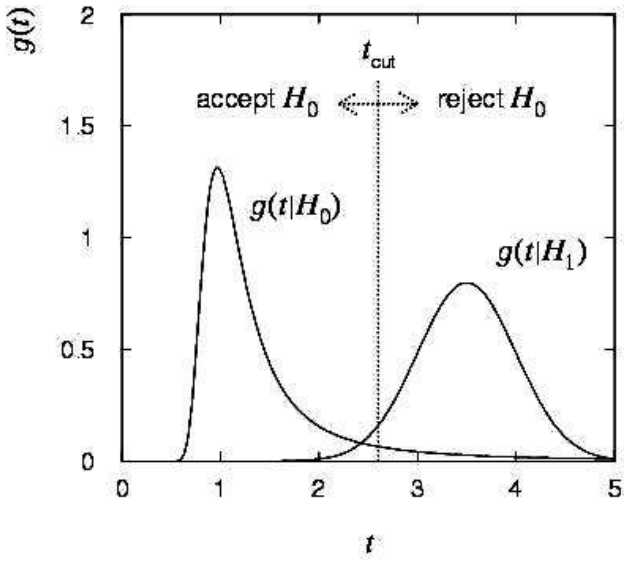
$$\beta = \int_{-\infty}^{t_{\text{cut}}} g(t|H_1)\, dt$$

$(1 - \beta = \text{power})$

Efficiency of event selection

Probability to accept an event which is signal (signal efficiency):

$$\varepsilon_{\mathsf{s}} = \int_{-\infty}^{t_{\mathsf{cut}}} g(t|\mathsf{s})\, dt = 1 - \alpha$$



Probability to accept an event which is background (background efficiency):

$$\varepsilon_{\mathsf{b}} = \int_{-\infty}^{t_{\mathsf{cut}}} g(t|\mathsf{b})\, dt = \beta$$

Purity of event selection

Suppose only one background type b; overall fractions of signal and background events are π_s and π_b (prior probabilities).

Suppose we select events with $t < t_{cut}$. What is the 'purity' of our selected sample?

Here purity means the probability to be signal given that the event was accepted. Using Bayes' theorem we find:

$$P(\mathrm{s}|t < t_{\mathrm{cut}}) = \frac{P(t < t_{\mathrm{cut}}|\mathrm{s})\pi_{\mathrm{s}}}{P(t < t_{\mathrm{cut}}|\mathrm{s})\pi_{\mathrm{s}} + P(t < t_{\mathrm{cut}}|\mathrm{b})\pi_{\mathrm{b}}}$$

$$= \frac{\varepsilon_{\mathrm{s}}\pi_{\mathrm{s}}}{\varepsilon_{\mathrm{s}}\pi_{\mathrm{s}} + \varepsilon_{\mathrm{b}}\pi_{\mathrm{b}}}$$

So the purity depends on the prior probabilities as well as on the signal and background efficiencies.

Constructing a test statistic

How can we select events in an 'optimal way'?

Neyman-Pearson lemma (proof in Brandt Ch. 8) states:

To get the lowest ε_b for a given ε_s (highest power for a given significance level), choose acceptance region such that

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

where c is a constant which determines ε_s.

Equivalently, optimal scalar test statistic is

$$t(\vec{x}) = \frac{f(\vec{x}|\mathrm{s})}{f(\vec{x}|\mathrm{b})}$$

Why Neyman-Pearson doesn't always help

The problem is that we usually don't have explicit formulae for the pdfs $f(\vec{x}|\mathrm{s})$, $f(\vec{x}|\mathrm{b})$.

Instead we may have Monte Carlo models for signal and background processes, so we can produce simulated data, and enter each event into an n-dimensional histogram.

Use e.g. M bins for each of the n dimensions, total of M^n cells.

But n is potentially large, $\rightarrow$ prohibitively large number of cells to populate with Monte Carlo data.

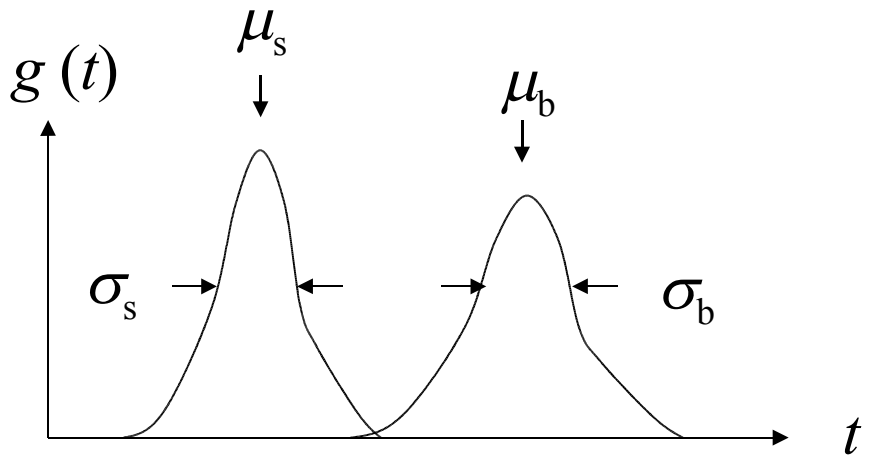
Compromise: make Ansatz for form of test statistic $t(\vec{x})$ with fewer parameters; determine them (e.g. using MC) to give best discrimination between signal and background.

Linear test statistic

Ansatz: $$t(\vec{x}) = \sum_{i=1}^{n} a_i x_i$$

Choose the parameters a_1, ..., a_n so that the pdfs $g(t|\mathrm{s})$, $g(t|\mathrm{b})$ have maximum 'separation'. We want:

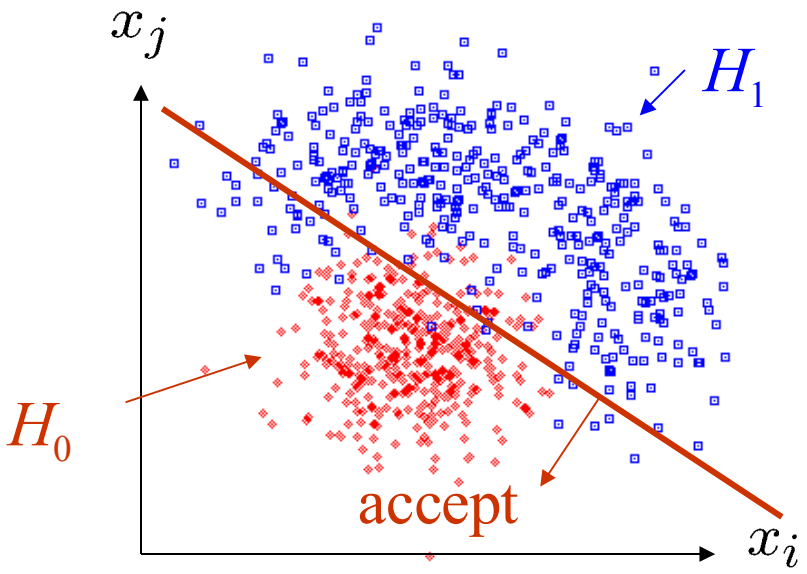
large distance between mean values, small widths



→ Fisher: maximize $J(\vec{a}) = \dfrac{(\mu_\mathrm{s} - \mu_\mathrm{b})^2}{\sigma_\mathrm{s}^2 + \sigma_\mathrm{b}^2}$

Fisher discriminant

Using this definition of separation gives a Fisher discriminant.



Corresponds to a linear decision boundary.

Equivalent to Neyman-Pearson if the signal and background pdfs are multivariate Gaussian with equal covariances; otherwise not optimal, but still often a simple, practical solution.

Nonlinear test statistics

The optimal decision boundary may not be a hyperplane,

→ nonlinear test statistic $t(\vec{x})$

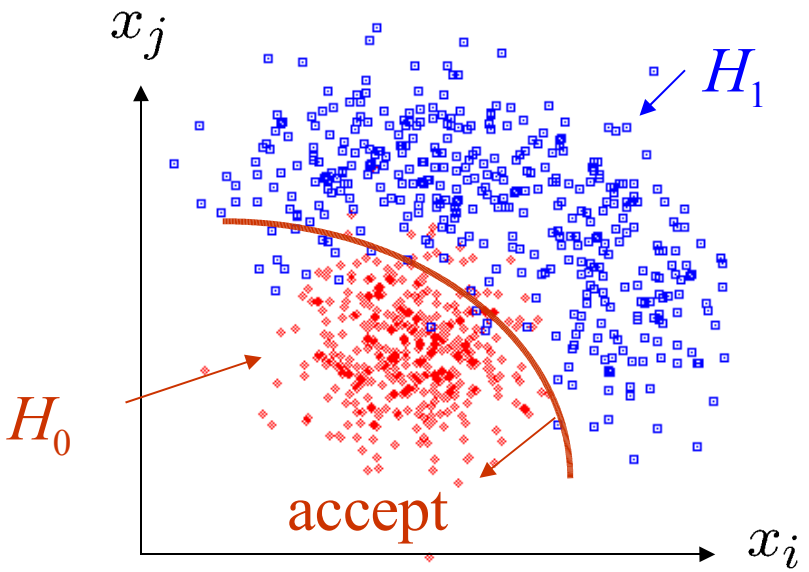
Multivariate statistical methods are a Big Industry:

Neural Networks,

Support Vector Machines,

Kernel density methods,

...

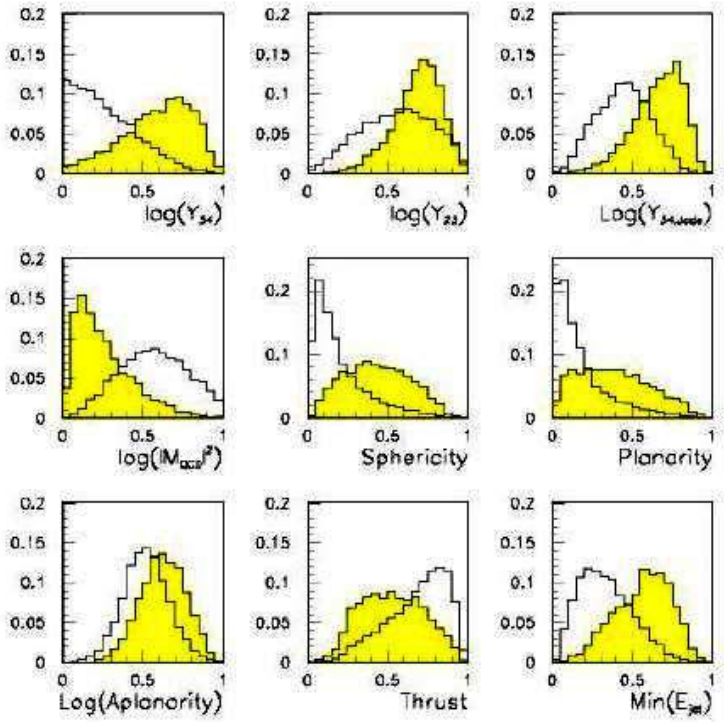


Particle Physics can benefit from progress in Machine Learning.

Neural network example from LEP II

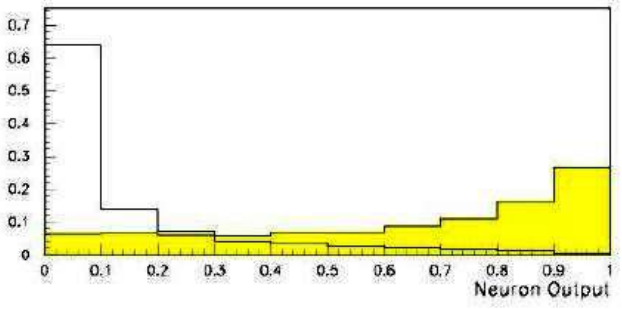
Signal: $e^+e^- \to W^+W^-$ (often 4 well separated hadron jets)

Background: $e^+e^- \to qqgg$ (4 less well separated hadron jets)



← input variables based on jet structure, event shape, ...
none by itself gives much separation.

Neural network output does better...



(Garrido, Juste and Martinez, ALEPH 96-144)

Wrapping up lecture 2

We've seen the Monte Carlo method:

calculations based on sequences of random numbers,
used to simulate particle collisions, detector response.

And we looked at statistical tests and related issues:

discriminate between event types (hypotheses),
determine selection efficiency, sample purity, etc.

Some modern (and less modern) methods were mentioned:

Fisher discriminants, neural networks,
support vector machines,...

In the next lecture we will talk about goodness-of-fit tests and then move on to another main subfield of statistical inference: parameter estimation.