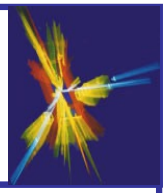


The Monte Carlo method

A numerical technique for calculating probabilities & related things using sequences of random numbers.

The usual steps:

- generate sequence $r_1, r_2, ..., r_m$ uniform in]0,1[.
- use them to produce another sequence $x_1, x_2, ..., x_n$ distributed according to pdf $f(x)$ of interrest
- use obtained x values to estimate some property of $f(x)$, e.g. fraction of x values within $[a,b]$ = $\int_a^b f(x)dx$

$\Rightarrow$ MC calculation sort of integration (at least formally)

Trivial for 1D: $\int_a^b f(x)dx$ obtainable by other methods, but MC more powerful for multi-dimensional problems.

MC x values = ”simulated data”
$\rightarrow$ used for testing statistical procedures

MC methods a wide field, actually own field in itself – here focus on the usage of MC for data analysis e.g. determining the statistical (& systematic) uncertainties. In such cases, MCs are used to generate different data distributions. So, let’s try to answer the question ”how can I generate the type of distribution I need?”.

more complete & deeper discussion found in e.g.

- MATR323 “Basics of Monte Carlo simulation”-course by Prof. Flyura Djurabekova in spring 2026 (period III)

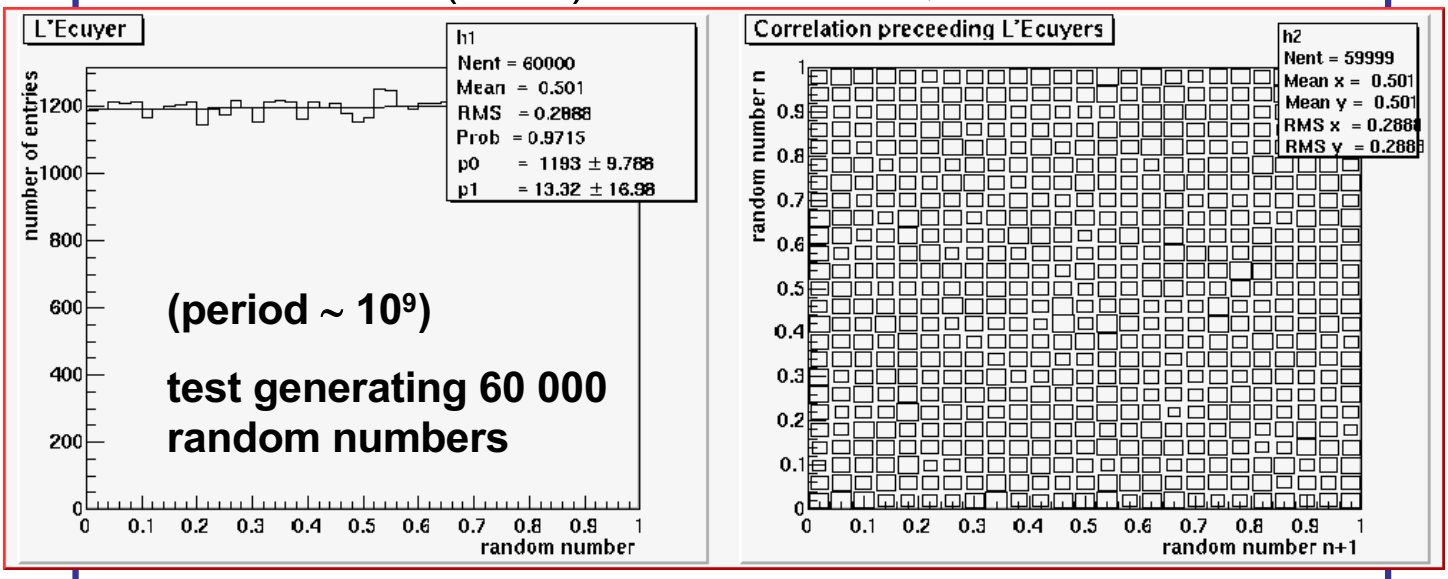
Random number generator

goal: to get uniformly distributed values in]0,1[interval.
$\Rightarrow$ "random number generator"
= computer algorithm to generate $r_1, r_2, \ldots, r_m$.
e.g. multiplicative linear congruential generator (MLCG)

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

n_i = integer, a = **multiplier**, m = **modulus** & n_0 = **seed**.
(NB! mod = modulus (remainder), e.g. 27 mod 5 = 2)
n_i follow periodic sequence in [1, m –1] $\Rightarrow$
$r_i = n_i / m$ distributed in]0,1[.

Choose a & m so that r_i's pass various tests of randomness: r_i's uniform in]0,1[, succeeding r_i's uncorrelated & "period" for r_i's long (maximum = m –1) e.g. L'Ecuyer, *Comm. ACM* 31(1988)742: a = 40692, m = 2147483399

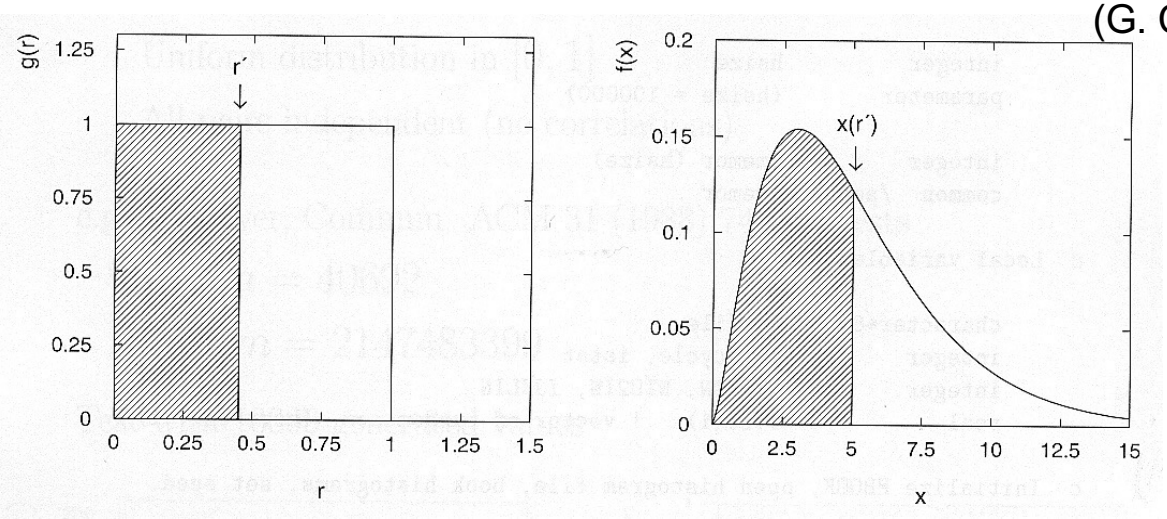


Much better algorithms exist e.g. `Mersenne twister`, with period $\approx 10^{6000}$. Many good algorithms implemented in freely available program libraries
NB! r_i's like above in reality **pseudorandom numbers**
See e.g. F.James, *Comput. Phys. Commun.* 60 (1990) 111

Inverse transform method

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

(G. Cowan)



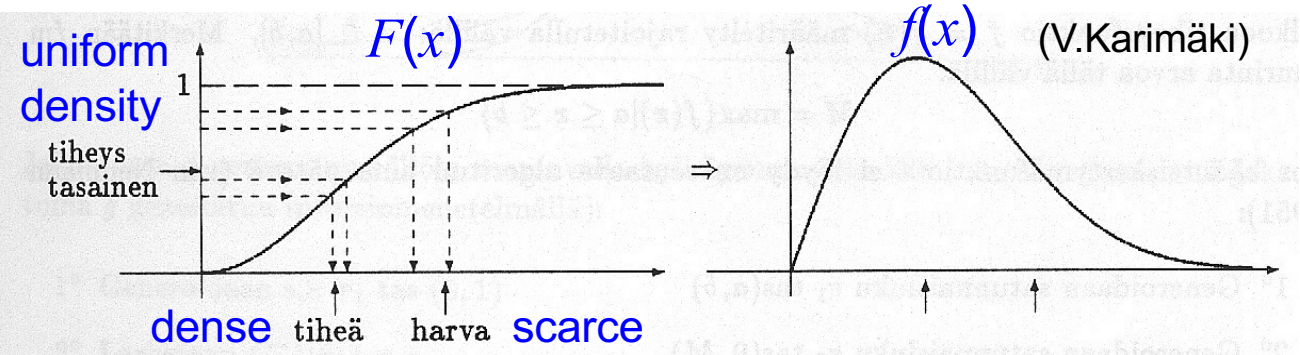
$P(r \le r') = P(x \le x(r'))$ i.e.

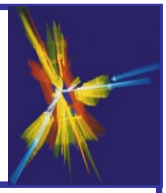
$$\int_{-\infty}^{r'} g(r)dr = r' = \int_{-\infty}^{x(r')} f(x')dx' = F(x(r'))$$

A method that always works when inverse function of cumulative distribution function $F(x)$ can be calculated or put in a table. Then **inverse transform method**:

- sample r from a uniform distribution]0,1[
- calculate $x = F^{-1}(r)$

then generated random numbers x that obey pdf $f(x)$. From following graph it is easy to see that it works





Discrete distributions

Inverse transform method for discrete distributions.
p_i = probability for integer i. First one has to put in table cumulative distribution function $F_j = \Sigma_{i=0}^{j} p_i$, j = 0, ...,N.
If infinite number of possible outcomes i, N to be set so large that $F_N \approx 1$. Generation algorithm for discrete pdf:

(i) sample r from a uniform distribution]0,1[.

(ii) find k so that $F_{k-1} < r < F_k$.

(iii) accept integer k–1

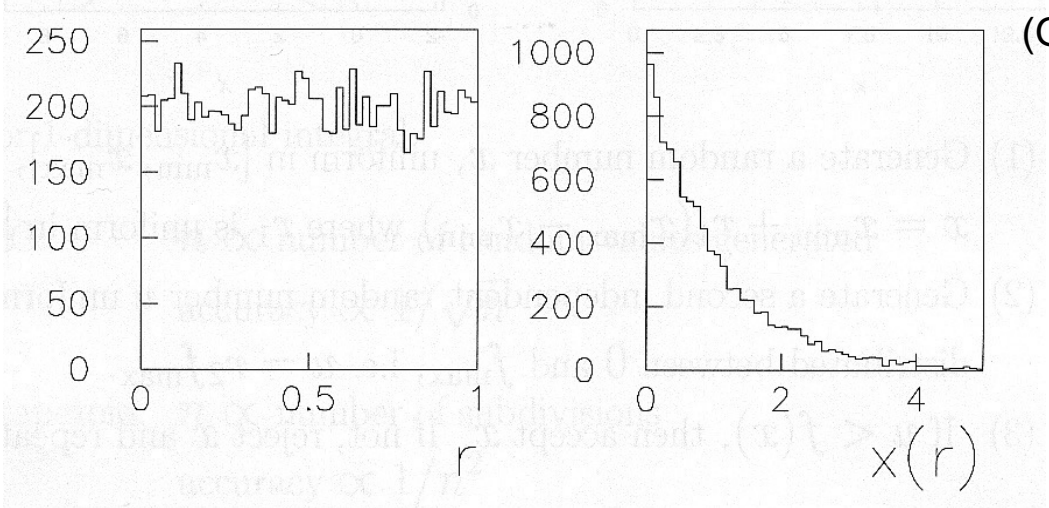
Resulting distribution proportional to probabilities p_k.

Example of inverse transform method:
exponential pdf: $f(x;\xi) = \xi^{-1} e^{-x/\xi} \quad (x \geq 0)$

cumulative distribution function:

$$F(x) = \int_0^x \xi^{-1} e^{-x'/\xi} dx' = 1 - e^{-x/\xi}$$

assume $r \in$]0,1[, now set $r = F(x)$ & solve for $x(r)$ $\Rightarrow$
$x(r) = -\xi \ln(1-r)$ (NB! $x(r) = -\xi \ln r$ works also)

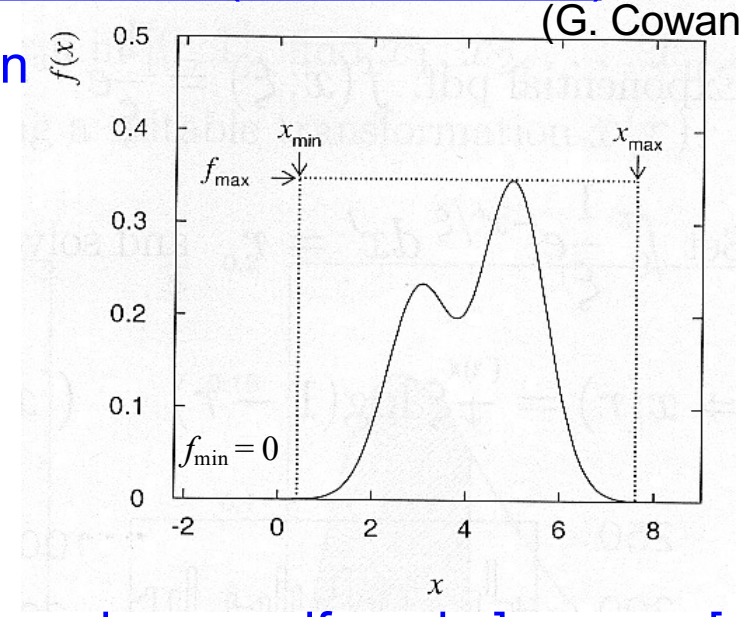


(G. Cowan)

Acceptance-rejection method (von Neumann)

(G. Cowan)

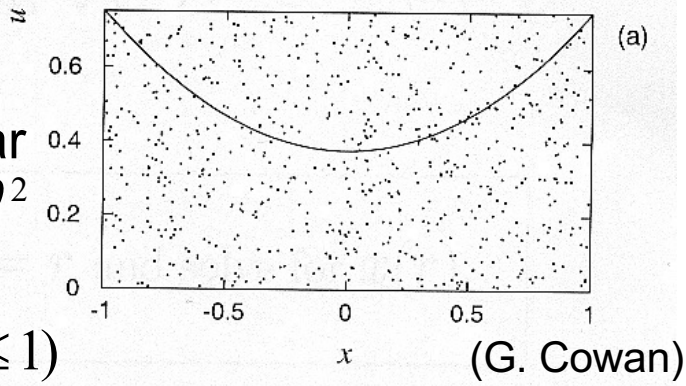
Often analytic solution impossible or very impractical ⇒ acceptance-rejection method (or **hit-or-miss**): enclose pdf in a box]f_{min}= min($f(x)$), f_{max} = max($f(x)$)[



(i) generate a random number x, uniform in]x_{min}, x_{max}[, i.e. $x = x_{min} + r_1(x_{max} - x_{min})$ where r_1 uniform in]0,1[
(ii) generate 2nd random number u uniformly distributed between f_{min} & f_{max}, i.e. $u = f_{min} + r_2(f_{max} - f_{min})$.
(iii) if $u < f(x)$, then accept x. If not, reject x & repeat.

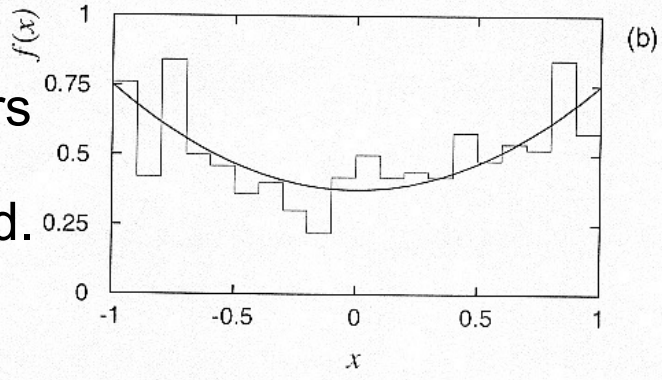
Example: generate a polar angle distribution $1 + \cos\theta^2$ $(-1 \le \cos\theta \le 1)$. $x = \cos\theta$

$$f(x) = \tfrac{3}{8}(1+x^2) \quad (-1 \le x \le 1)$$



(G. Cowan)

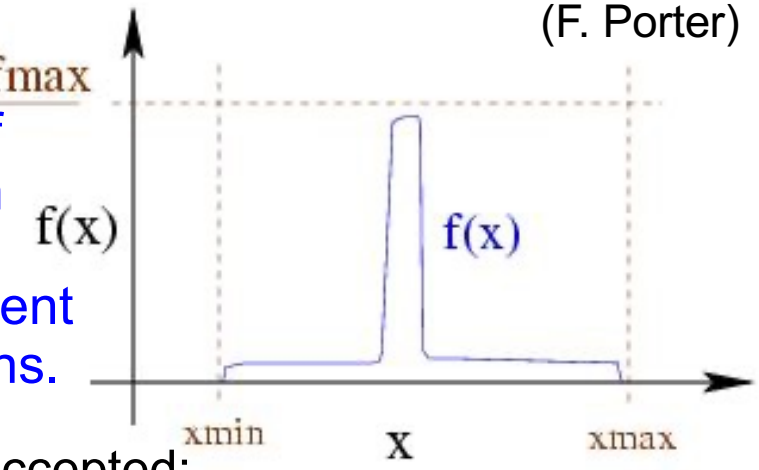
f_{min} = 0, f_{max} = ¾; x of pairs of random numbers lying below the curve accepted. The distribution of the accepted x shown below.



Importance sampling

Acceptance–rejection method simple to apply but efficiency of algorithm depends on area ratio of pdf to enclosing box. Inefficient for "peaky" distributions.

(F. Porter)



Fraction of trial points accepted:

$$\varepsilon = \frac{\text{Area under curve}}{\text{Area of box}} = \frac{\int_{x_{min}}^{x_{max}} f(x)dx}{(f_{max}-f_{min})(x_{max}-x_{min})}$$

Importance sampling:

To improve acceptance-rejection method efficiency, use importance sampling method. First random numbers generated according to $g(x)$ satisfying $g(x) > f(x)$ all over whole x-interval. x's generated according to $g(x)/\int g(x)$ (with e.g. inverse transform method), and x accepted if $u < f(x)$, where u random number $]0, g(x)[$, or weighted with a factor $f(x)/g(x)$

e.g. to estimate the integral of the tail of Gaussian distribution. Instead of using the original Gaussian, use an equivalent Gaussian with larger acceptance rate to reduce uncertainty of the estimate and coverge much faster.

$P(Y > 3)$ if Y Gaussian $N(\mu = 0, \sigma = 1)$: truth ≈ 0.00135. $h(Y_i) = 1$ if $Y_i > 3$; $h(Y_i) = 0$ if $Y_i \leq 3$

(J. Cisewski)

- Draw an iid sample $Y_1, \ldots, Y_{100}$ from a $N(0,1)$, then the estimator is

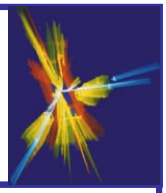
integral $\hat{I} = \frac{1}{100}\sum_{i=1}^{100} h(Y_i)$

- Draw an iid sample $Y_1, \ldots, Y_{100}$ from a $N(4,1)$, then the estimator is

$\hat{I} = \frac{1}{100}\sum_{i=1}^{100} \frac{h(Y_i)f(Y_i)}{g(Y_i)}$ importance sampling

where f is the density of a N(0,1) and g is the density of N(4,1)

$N = 10^5$	Expected Value	Variance
Truth	0.00135	0
Monte Carlo	0.00136	1.3×10^{-5}
Importance Sampling	0.00135	9.5×10^{-8}

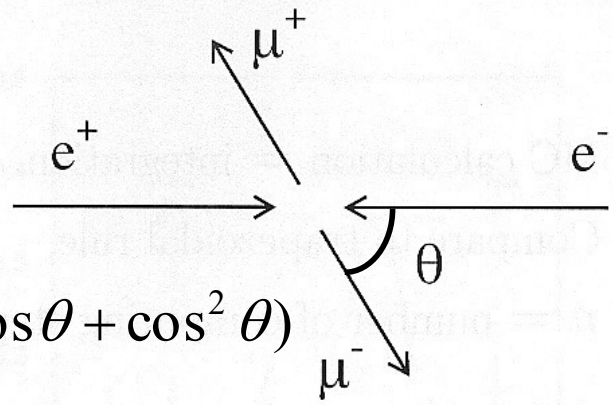


MC methods often used to ”mimic” data in e.g. particle physics, a two-step process: simulation of physics, ”**event generator**” & simulation of response of experimental apparatus,”**detector simulation**”.

MC event generator:

simple example:

$e^+e^- \rightarrow \mu^+\mu^-$



generate θ and ϕ:

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

$$g(\phi) = 1/2\pi$$

In reality implemented into program packages that accounts for all (known) effects (hadronisation, initial and final state radiation, longlived particles etc…)

MC detector simulation (built on e.g. GEANT4):

Input: particle list & momenta from event generator

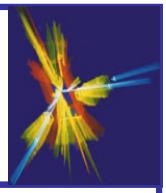
simulate particle interactions with detector material & detector response: multiple Coulomb scattering (generate particle scattering angle), ionization energy loss (generate energy loss dE/dx), electromagnetic & hadronic showers, produce detector signals, electronics response ...

Output: simulated ”raw” data → input to reconstruction

Usage:

Predict what should be seen at ”detector level” given a hypothesis at ”generator level”. Compare with real data.

- optimize measurement & experiment sensitivity.
- estimate efficiency & purity (expected signal & background)
- simulate measurement many times (”gedanken” experiment)



Standard Gaussian distributed random numbers

cumulative distribution function $F(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-x'^2/2} dx'$

Square evaluation method: Generate numbers in polar coordinates & afterwards transform them to cartesian

$$f(x)f(y)dxdy = re^{-r^2/2}dr\frac{d\phi}{2\pi} = g(r)h(\phi)drd\phi$$

r & ϕ generated according to $g(r)$ & $h(\phi) \Rightarrow$

x & y independent Gaussian distributed random numbers.

Marsaglia polar method: very efficient algorithm for generating Gaussian random numbers

u_1 & u_2 uniform in]0,1[

construct $v_1 = 2u_1 - 1$ & $v_2 = 2u_2 - 1$ (uniform in]-1,1[),

if $r^2 = v_1^2 + v_2^2 > 1$ start over again, otherwise

$$z_1 = v_1\sqrt{-2\ln r^2/r^2} \quad \text{and} \quad z_2 = v_2\sqrt{-2\ln r^2/r^2}$$

z_1 & z_2 independent Gaussian numbers with $\mu = 0$ & $\sigma = 1$

$z_i' = \mu + \sigma z_i$ Gaussian numbers with mean μ & variance σ^2

Poisson random numbers: efficient algorithm for small ν:

Set $k = 1$ & $A = 1$ at start, then iterate until successful choice

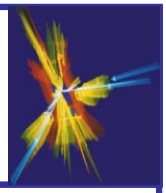
(i) generate u (uniform in]0,1[), replace A with uA

(ii) if $A < \exp(-\nu)$, where ν mean of Poisson distribution, accept $n_k = k - 1$ & stop.

(iii) replace k by $k+1$ and repeat (i).

for large ν (> ~10), faster & easier to generate Gaussian random numbers since Poisson distribution ≈ Gaussian.

Accuracy of MC methods



χ^2 random numbers: for even n, generate $n/2$ uniform random numbers u_i; then $y = -2\ln\left(\prod_{i=1}^{n/2} u_i\right)$ follows $\chi^2(n)$.

for odd n, generate $(n-1)/2$ uniform numbers u_i & one Gaussian z; then $y = -2\ln\left(\prod_{i=1}^{(n-1)/2} u_i\right) + z^2$ follows $\chi^2(n)$.

Binomial random numbers: principle same as for any discrete distribution. Use inverse transform method and tabulate cumulative distribution $F(x)$
Most computer libraries include generators for most common distributions like Gaussian, Poisson, χ^2 etc…

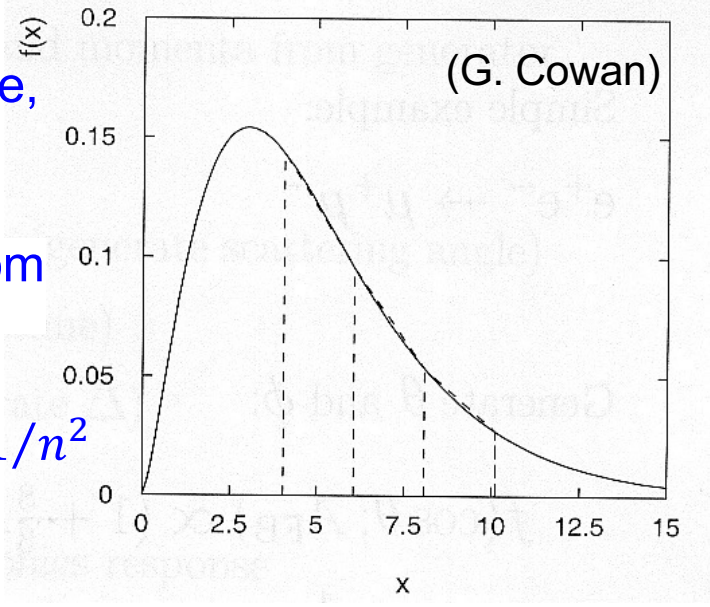
Accuracy of Monte Carlo methods:

MC = ”integration”.
compare to trapezoidal rule,
n = # of computing steps

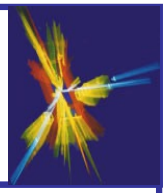
for 1D integral: MC: $n \propto$ number of accepted random values, accuracy $\propto 1/\sqrt{n}$

trapezoid: $n \propto$ number of subdivisions, accuracy $\propto 1/n^2$

in 1D trapezoid wins!!!
but in d dimensions:
MC: accuracy $\propto 1/\sqrt{n}$ ← independent of d !
trapezoid: accuracy $\propto 1/n^{2/d}$



MC wins for d > 4. Gaussian quadrature better than trapezoid but for high enough d, MC always wins!!
(see e.g. F. James, *Rep. Prog. Phys.* 43 (1980) 1145).



A statistical test:

applied to physics mainly in two different ways
– how to distinguish events of interest, ”**signal**”, from (a large number of) uninteresting events, ”**background**”.
– how well a given hypothesis compatible (without any alternative) with observed data, ”**goodness-of-fit**” test.

Hypothesis testing:

Statistics related to questions where answers not numerical but logical – a ”yes” or ”no”.

Example: ”Is this particle an electron / a signal event?”

Statistics (& science in general) can never in absolute terms say that something is true.
Only falsify (”empirical falsification” Karl Popper 1930).

Answer generally a probability.
This probability refers to ensemble of statements.

Result of a measurement $\vec{x} = (x_1, \ldots, x_n)$ follows a pdf in n-dimensional space that depends on variables $x_1, \ldots, x_n$ e.g. pdf $f(\vec{x})$ specified by some **hypotheses** H_0, H_1… each having a probability density

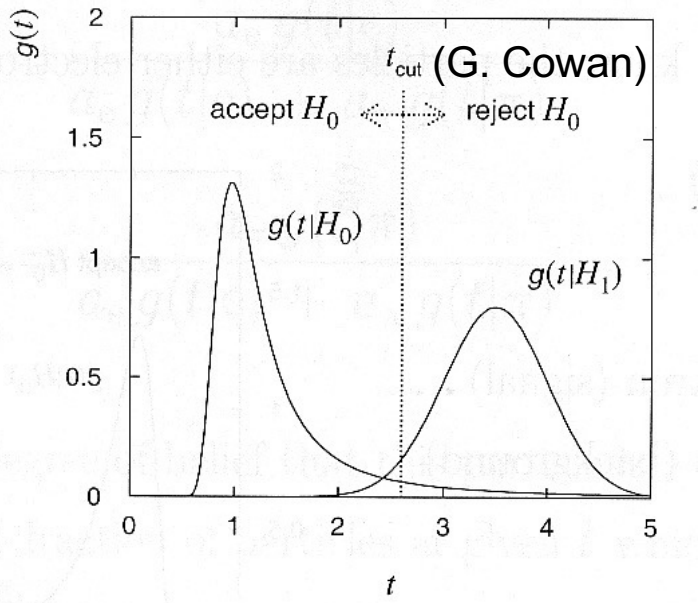
$$f(\vec{x} \mid H_0), f(\vec{x} \mid H_1) \text{ etc ..}$$

Equivalent to provide a statement & ask if true or not.

null hypothesis: H_0, hypothesis whose validity tested.
alternative hypothesis: H_1(, H_2 ...) to compare with H_0.
simple hypothesis: $f(\vec{x})$ completely specified.
composite hypothesis: $f(\vec{x}, \theta)$, parameter(s) θ unknown.

Test statistic

Many dimensions akward ⇒ use **test statistic** $t(\bar{x})$ of lower dimension (e.g. 1D) to compactify data, keeping as much hypothesis discrimination power as possible.
A test statistic t following pdf's $g(t(\bar{x})\,|\,H_0), g(t(\bar{x})\,|\,H_1)$



Often formulate compatibility between data & various hypotheses in terms of an acceptence or a rejection of null hypothesis $\boldsymbol{H_0}$

Critical region i.e. where t not likely to occur if $\boldsymbol{H_0}$ true $t \geq \boldsymbol{t_{cut}}$ in figure (alternative define **acceptance region**)

If observed value $\boldsymbol{t_{obs}}$ in critical region, reject $\boldsymbol{H_0}$, otherwise ”accept” (or strictly speaking, not reject !!).

Critical region chosen such that probability for t to be there assuming hypothesis $\boldsymbol{H_0}$ some small value α = **significance level**.

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

$\boldsymbol{H_0}$ rejection even if true error of 1st kind (= probability α)
$\boldsymbol{H_0}$ acceptence if some other hypothesis true error of 2nd kind (probability β)

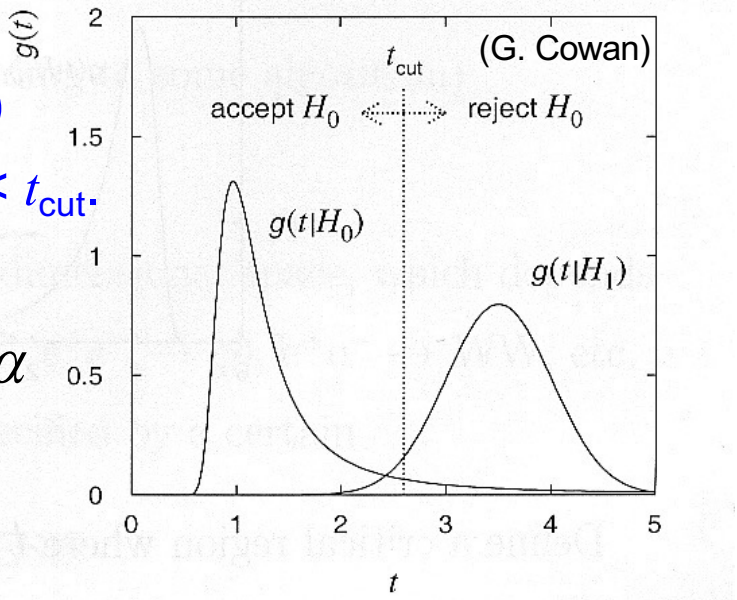
$$\beta = \int_{-\infty}^{t_{cut}} g(t\,|\,H_1)\,dt$$

Can define $1-\beta$ as power to discriminate against alternative hypothesis.

Example with particle identification

n energy loss measurements for a particle in a particle detector, t = truncated mean of the measurements
& suppose all particles either electrons (e) or pions (π).

H_0 = electron (signal)
H_1 = pion (background)



select electrons with $t < t_{cut}$.
selection efficiencies:

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

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

Higher t_{cut}: higher e efficiency but more π background.
Lower t_{cut}: lower e efficiency but better sample purity.

For observed value t, probability P, to have an e (or π)?

$$P(e\,|\,t) = \frac{a_e g(t\,|\,e)}{a_e g(t\,|\,e) + a_\pi g(t\,|\,\pi)} \qquad P(\pi\,|\,t) = \frac{a_\pi g(t\,|\,\pi)}{a_e g(t\,|\,e) + a_\pi g(t\,|\,\pi)}$$

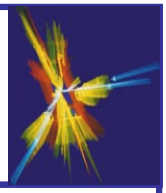
$a_e(a_\pi)$ = fraction of electrons (pions) in sample; $a_e + a_\pi = 1$

Bayesian: degree of belief that particle is an e or a π.
Frequentist: particle fraction at given t which is e or π.

sample purity, p_e, fraction of actual electrons in sample:

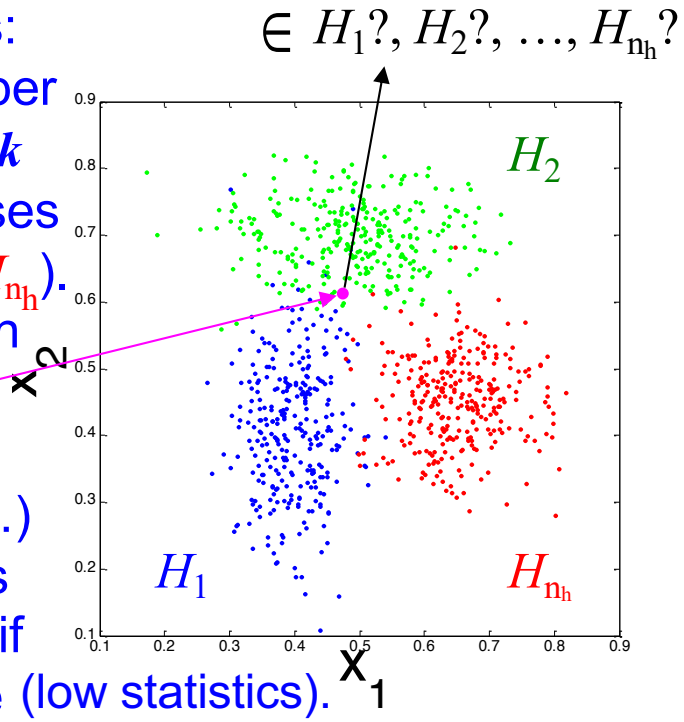
$$p_e = \int_{-\infty}^{t_{cut}} a_e g(t\,|\,e)dt \Big/ \int_{-\infty}^{t_{cut}} [a_e g(t\,|\,e) + a_\pi g(t\,|\,\pi)]dt$$

= electron probability averaged over interval $(-\infty, t_{cut}]$.
NB! purity depend on (un)known e & π fractions a_e & a_π.



When is discriminant analysis needed?

aim of discriminant analysis: distinguish between a number of distinct hypotheses, H_k , k = 1,…, n_h. various hypotheses known *a priori* (H_1, H_2, …, H_{n_h}). problem: determine to which hypothesis new element x (determined *a posteriori* by simulation, measurement,…) belongs. Careful hypothesis testing required, especially if only few elements available (low statistics).



Discriminant analysis techniques applied in many fields e.g. astrophysics, particle physics, biophysics, imaging,...

A multidimensional test statistic $\bar{t} = (t_1,\ldots,t_m)$ and hypotheses H_0 (”signal”) & H_1 (”background”). Optimal choice of critical region, i.e. what selection to use ?

Neyman-Pearson lemma: acceptance region giving highest power (& also highest signal purity) for a given significance level α (or selection efficiency $\varepsilon = 1-\alpha$) :

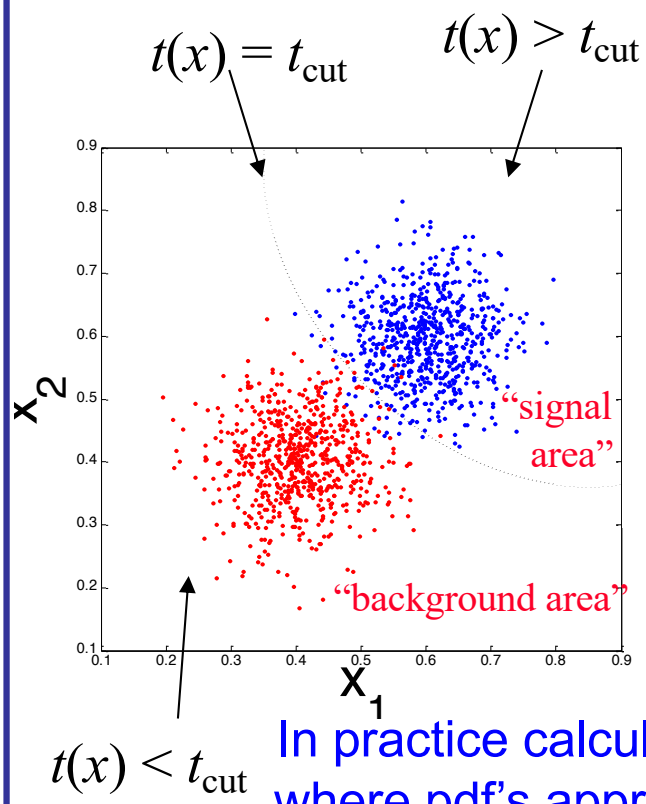
$$\frac{g(\bar{t}\,|\,H_0)}{g(\bar{t}\,|\,H_1)} > c,$$

c = constant, fixed by significance level (or selection purity)

equivalently, optimal test statistic i.e. the **likelihood ratio** for simple hypotheses H_0 & H_1. Requiring $r > c$ gives maximum significance level (efficiency) for a given power (purity).

$$r = \frac{g(\bar{t}\,|\,H_0)}{g(\bar{t}\,|\,H_1)},$$

Constructing a test statistic



A vector $\bar{x} = (x_1, \ldots, x_n)$ measured for each event. construct 1D test statistic $t(\bar{x})$ to distinguish between two hypotheses H_0 & H_1.

most optimal choice: the likelihood ratio $t(\bar{x}) = \dfrac{f(\bar{x}\,|\,H_0)}{f(\bar{x}\,|\,H_1)}$,

but need pdfs analytically

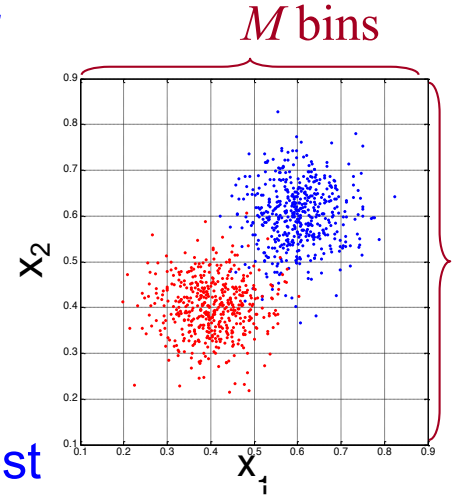
$f(\bar{x}\,|\,H_0)$ & $f(\bar{x}\,|\,H_1)$

(in most cases not possible).

In practice calculated, from MC simulations, where pdf's approximated by multidimensional histograms filled with $\bar{x}$ of each generate event.

method impractical if number of input variables too large since available MC statistics finite.

<u>compromise:</u> make Ansatz for functional form of $t(\bar{x})$ with a fewer number of input variables; choose variables (e.g. based on MC) that give best discrimination between H_0 & H_1.



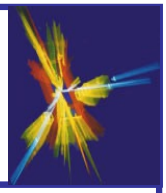
Number of parameters to be determined by MC: M^2

<u>linear test statistic</u>: e.g. Fisher discriminant – straight "line" cut in $\bar{x}$-space:

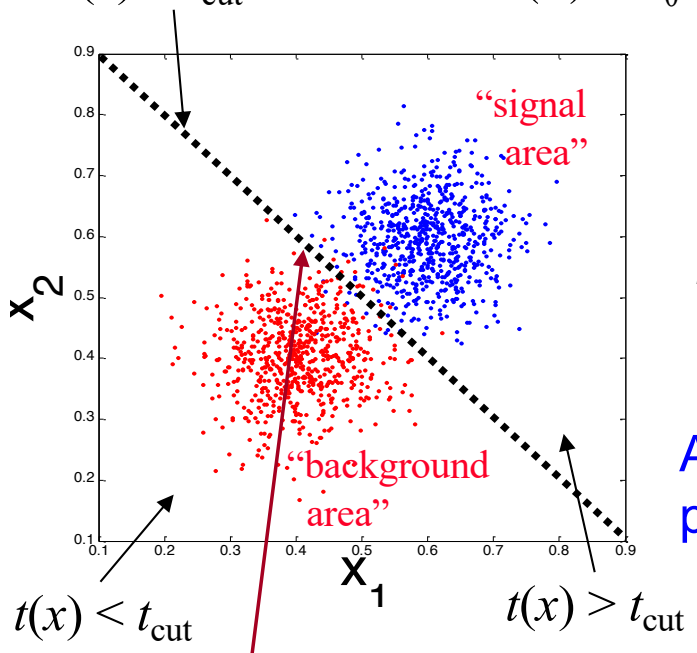
$$t(\bar{x}) = \sum_{i=1}^{n} a_i x_i = \bar{a}^T \bar{x}$$

<u>nonlinear test statistic:</u> e.g. neural networks – more complicated cut function in $\bar{x}$-space.

Linear test statistics



Ansatz: $t(\bar{x}) = a_0 + \sum_{i=1}^{n} a_i x_i = a_0 + \bar{a}^T \bar{x}$



$t(\bar{x})$ a **Fisher linear discriminant** see e.g. R.A. Fisher, *Annals of Eugenics* **7** (1936) 179; R.A. Fisher: Contributions to mathematical statistics, 1950.

A choice of $\bar{a}$ gives certain pdf's $g(t\,|\,H_0)$ & $g(t\,|\,H_1)$

choose a_i's to maximize "separation" between $g(t\,|\,H_0)$ & $g(t\,|\,H_1)$ $\Rightarrow$ must define "separation" more exactly !!

n = 2, a t_{cut} can be interpreted a straight line in the x_1–x_2 plane.

Data $\bar{x} = (x_1, \ldots, x_n)$ have mean values & covariance matrix

$$(\mu_k)_i = \int x_i f(\bar{x}\,|\,H_k)\,dx_1 \ldots dx_n \qquad i, j = 1, \ldots, n$$

$$(V_k)_{ij} = \int (x-\mu_k)_i (x-\mu_k)_j f(\bar{x}\,|\,H_k)\,dx_1 \ldots dx_n \qquad k = 0,1$$

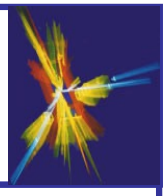
Each hypothesis has certain mean & variance of $t(\bar{x})$

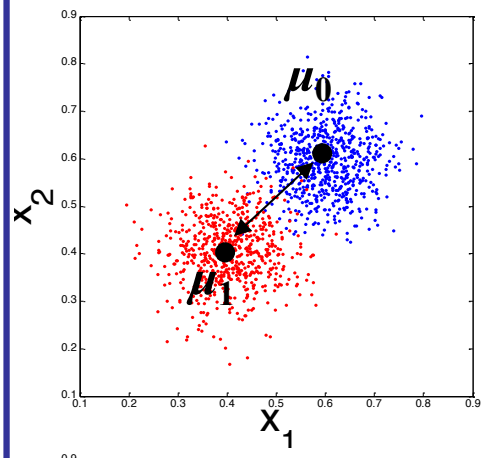
$$\tau_k = \int t(\bar{x}) g(t(\bar{x})\,|\,H_k)\,dt = \bar{a}^T \bar{\mu}_k$$

$$\Sigma_k^2 = \int (t(\bar{x}) - \tau_k)^2 g(t(\bar{x})\,|\,H_k)\,dt = \bar{a}^T \mathbf{V}_k \bar{a}$$

To optimize separation: large $|\tau_0 - \tau_1|$, small Σ_0^2 & Σ_1^2 (pdf's tightly concentrated about well separated means).

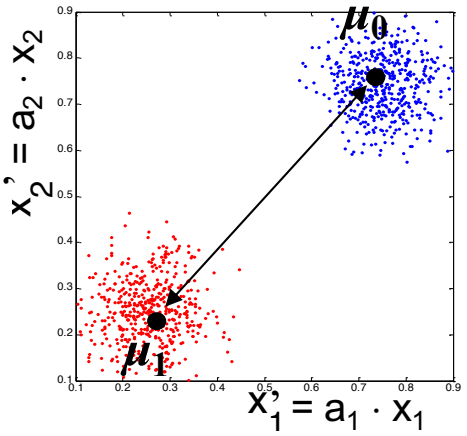
Fisher discriminant

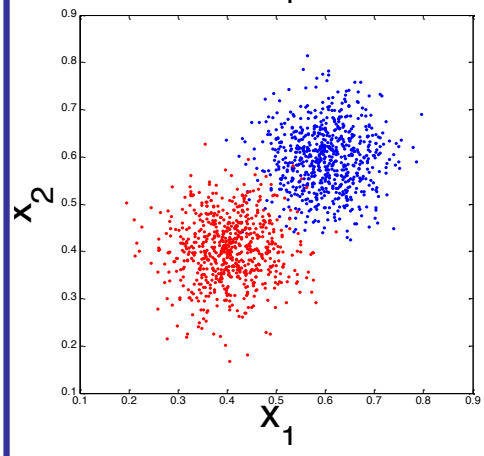




MAX

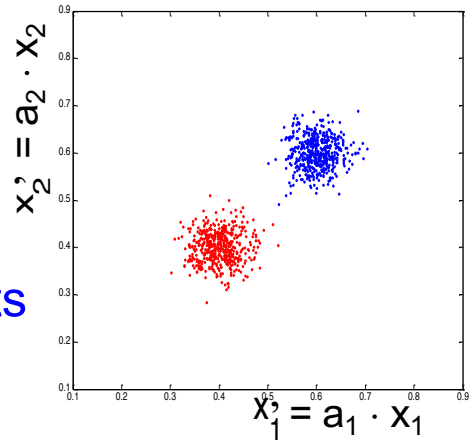
distance between μ_0 & μ_1





MIN

spread i.e. concentrate the elements of H_0 & H_1 .



Fisher defined as a measure of separation (taking both into account): $J(\bar{a}) = (\tau_0 - \tau_1)^2 \big/ (\Sigma_0^2 + \Sigma_1^2)$

Numerator & denominator can be written as

$$(\tau_0 - \tau_1)^2 = \sum_{i,j=1}^{n} a_i a_j (\mu_0 - \mu_1)_i (\mu_0 - \mu_1)_j = \sum_{i,j=1}^{n} a_i a_j B_{ij} = \bar{a}^T \mathbf{B}\, \bar{a}$$

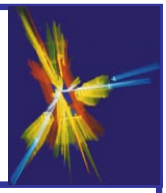
$$\Sigma_0^2 + \Sigma_1^2 = \sum_{i,j=1}^{n} a_i a_j (\mathbf{V}_0 + \mathbf{V}_1)_{ij} = \bar{a}^T \mathbf{W}\, \bar{a}$$

This gives $J(\bar{a}) = \dfrac{\bar{a}^T \mathbf{B}\, \bar{a}}{\bar{a}^T \mathbf{W}\, \bar{a}} = \dfrac{\text{separation between classes}}{\text{sum of variances within classes}}$

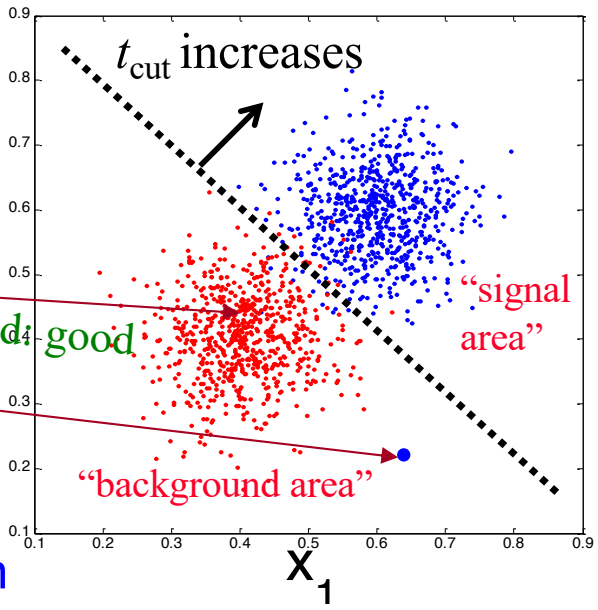
set $\partial J(\bar{a}) / \partial a_i = 0, \quad \text{for all } i \quad \Rightarrow \quad \bar{a} \propto \mathbf{W}^{-1} (\bar{\mu}_0 - \bar{\mu}_1)$

Above **Fisher's linear discriminant function** & corresponding test statistic a **Fisher discriminant.**

Fisher discriminant



Note coefficients a_i only determined up to arbitrary scale factor. Use scale (t_{cut}) & offset a_0 to fix purity and efficiency. Larger t_{cut} can improve signal purity with efficiency loss & vice-versa.



To determine coefficients a_i, need matrix **W** & expectation values $\mu_{(0,1)i}$. Usually estimated from a set of training data e.g. from a MC simulation. Important point being that one doesn't need to determine joint pdf's $f(\bar{x}\,|\,H_0)$ & $f(\bar{x}\,|\,H_1)$ as n-dimensional histograms, only means & covariances. Instead of determining M^n parameters now only $n(n+1)/2$.

Maximizing $J(\bar{a}) = \dfrac{(\tau_0 - \tau_1)^2}{(\Sigma_0^2 + \Sigma_1^2)}$ with fixed τ_0 & τ_1 same as minimizing $\Sigma_0^2 + \Sigma_1^2 = E_0[(t(\bar{x}) - \tau_0)^2] + E_1[(t(\bar{x}) - \tau_1)^2]$

(E_k denotes expectation value under hypothesis H_k)

→ maximizing Fisher's $J(\bar{a})$ kind of **least square** problem.
(more about parameter determination using least squares later)

Special Fisher discriminants: $f(\bar{x}|H_k)$ multidimensional Gaussians with same covariance matrix $\mathbf{V}_0 = \mathbf{V}_1 \equiv \mathbf{V}$

$$f(\bar{x}\,|\,H_k) = \frac{1}{(2\pi)^{n/2}\,|\,\mathbf{V}\,|^{1/2}} \exp[-\tfrac{1}{2}(\bar{x} - \bar{\mu}_k)^T \mathbf{V}^{-1}(\bar{x} - \bar{\mu}_k)], \quad k = 0, 1$$

and the Fisher discriminant: $t(\bar{x}) = a_0 + (\bar{\mu}_0 - \bar{\mu}_1)^T \mathbf{V}^{-1}\bar{x}$

Fisher discriminant

Recall likelihood ratio (maximum efficiency at given purity):

$$r = \frac{f(\bar{x}\,|\,H_0)}{f(\bar{x}\,|\,H_1)} = \exp[\tfrac{1}{2}(\bar{x}-\bar{\mu}_1)^T \mathbf{V}^{-1}(\bar{x}-\bar{\mu}_1) - \tfrac{1}{2}(\bar{x}-\bar{\mu}_0)^T \mathbf{V}^{-1}(\bar{x}-\bar{\mu}_0)]$$

$$r = \exp[(\bar{\mu}_0-\bar{\mu}_1)^T \mathbf{V}^{-1}\bar{x} - \tfrac{1}{2}\bar{\mu}_0^T \mathbf{V}^{-1}\bar{\mu}_0 + \tfrac{1}{2}\bar{\mu}_1^T \mathbf{V}^{-1}\bar{\mu}_1] \propto e^t$$

i.e. $t(\bar{x}) \propto \ln r + \text{const.}$ t given by monotonic function of r

$\Rightarrow$ Fisher discriminant = likelihood ratio i.e. most optimal.

NB! for non-equal $\mathbf{V}_i$ or non-Gaussian pdfs, no longer true.

Multidimensional Gaussian with equal covariance matrices also gives simple expressions for posterior hypotheses probabilities

e.g. (Bayes' theorem)
$$P(H_0\,|\,\bar{x}) = \frac{f(\bar{x}\,|\,H_0)\pi_0}{f(\bar{x}\,|\,H_0)\pi_0 + f(\bar{x}\,|\,H_1)\pi_1} = \frac{1}{1+(\pi_1/\pi_0 r)},$$

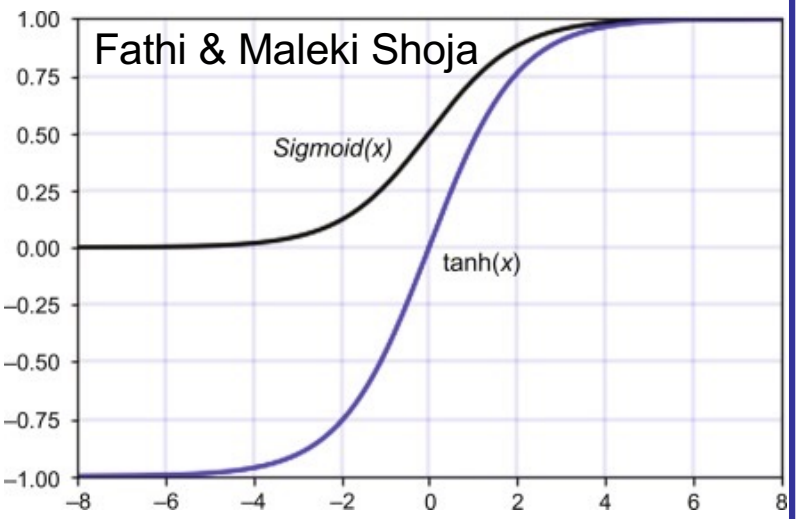
where π_0 & π_1 prior probabilities for H_0 & H_1.

Combining this with above expression for r, one gets

$$P(H_0\,|\,\bar{x}) = \frac{1}{1+e^{-t}} \equiv s(t)$$

the **logistic sigmoid** function (see figure below)

Logistic sigmoid a very common neural network **activation function** (= giving the weight of the node in the network). Another common one is the hyperbolic tangent:



$$P(H_0|\bar{x}) = \tanh(t) = (e^t - e^{-t})/(e^t + e^{-t})$$

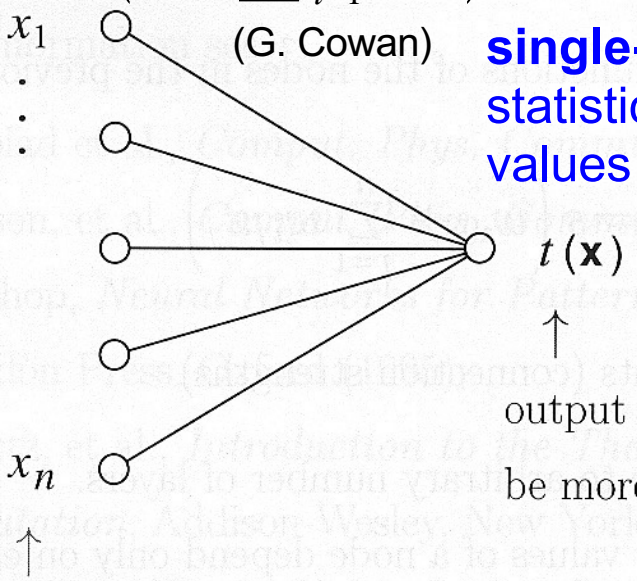
Optimal decision boundary mostly not "straight line" (due to non-equal $\mathbf{V}_i$ or non-Gaussian pdfs), use instead nonlinear test statistics: **neural networks, boosted decision trees…**

Artificial Neural Networks (ANN) today used in many fields: medical imaging, pattern recognition, financial forecasting etc... but also in physics. Assume $t(\bar{x})$ to have the form

$$t(\bar{x}) = s\left(a_0 + \sum_{i=1}^{n} a_i x_i\right)$$

where $s(\cdot)$ activation function (often logistic sigmoid or $\tanh(t)$)

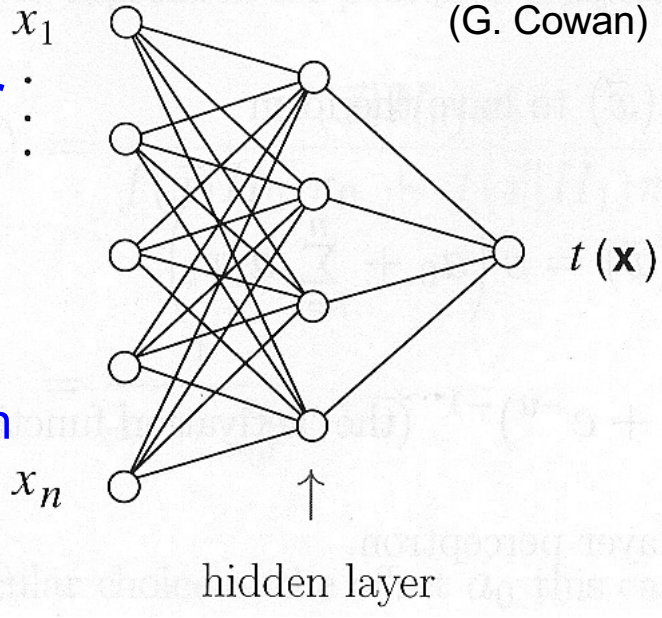
(G. Cowan)



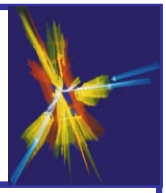
single-layer perceptron a test statistic of this form. $\bar{x}$ input values represented as a set of nodes. $s(\cdot)$ monotonic $\Rightarrow$ $t(\bar{x})$ corresponds to a **linear test statistic.**

Generalized to **multilayer perceptrons (MLP)** with one or several hidden layers. Each hidden layer has m nodes (h_1, ... h_m). Usually connection are restricted: value of a given node only depends on nodes in previous layer, a **feed-forward network.**

(G. Cowan)



Neural networks



In a two-layer perceptron output defined by

$$t(\bar{x}) = s\left(a_0 + \sum_{i=1}^{m} a_i h_i(\bar{x})\right), \text{ where } h_i(\bar{x}) = s\left(w_{i0} + \sum_{j=1}^{n} w_{ij} x_j\right)$$

$t(\bar{x})$ **non-linear test statistic**, since non-linear function of inputs x_i. a_i and w_i **connection strengths** or **weights** (n+1)m free parameters for n input parameters. With more nodes, ANN closer to optimal but more parameters to fix.

Parameters determined by minimizing an error function

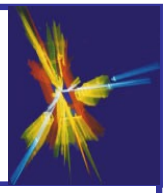
$$\varepsilon = E_0[(t - t^{(0)})^2] + E_1[(t - t^{(1)})^2]$$

where $t^{(0)}$ & $t^{(1)}$ target values e.g. 0 & 1 for logistic sigmoid, equivalent to least square method for Fisher discriminants.

To determine parameters minimizing ε, iterative numerical methods, network training or learning, used. In practice expectation values replaced by means computed from training samples, i.e. MC simulations. Learning starts with random initial values for weights and proceed evaluating function using training data. Weights adjusted to minimize ε by different methods, i.e. error-back-propagation.

Theorem: MLP with a single hidden layer having sufficient number of nodes can approximate arbitrarily well optimal decision boundary;*Leshno et al., Neural networks 6 (1993) 861*

Advantages/disadvantages of using a single hidden layer with many nodes over many hidden layers not known exactly but seems reasonable to assume that many layers have better performance & more stable than a single layer $\Longrightarrow$**deep learning** (networks with 10's or 100's of hidden layers)



Nonlinear test statistics

Focus here on supervised learning i.e. answer for each training event known

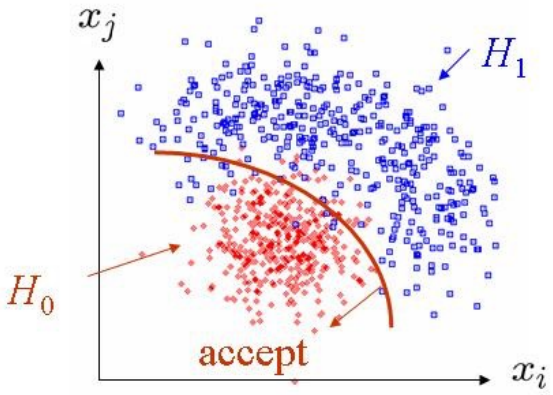
The optimal decision boundary may not be a hyperplane,

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

Multivariate statistical methods are a Big Industry:

- artificial neural networks
- boosted decision trees
- generative AI

…

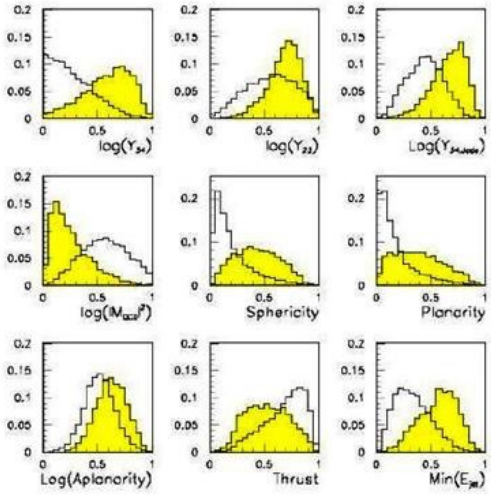


Physics can benefit from progress in Machine learning

Neural network example from LEP II

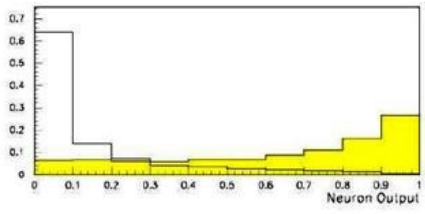
Signal: $e^+e^- \to W^+W^-$ (often 4 well separated hadron jets) yellow

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



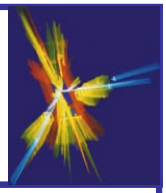
← input variables based on jet structure, event shape, ...
none by itself gives much separation.

Neural network output does better...



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

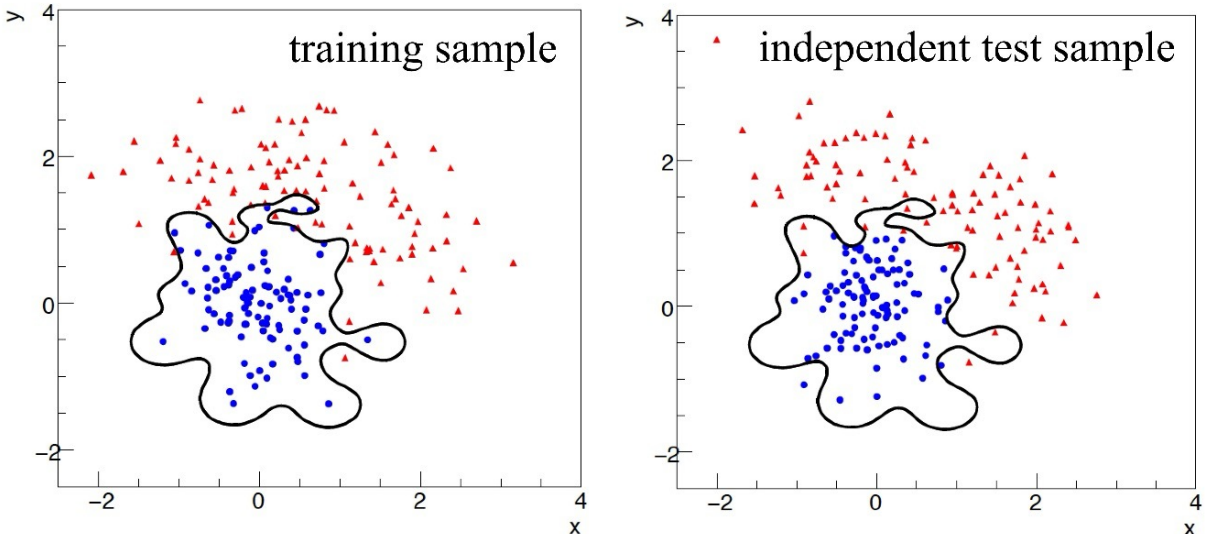
Machine learning in python: sklearn, https://scikit-learn.org/



Overtraining

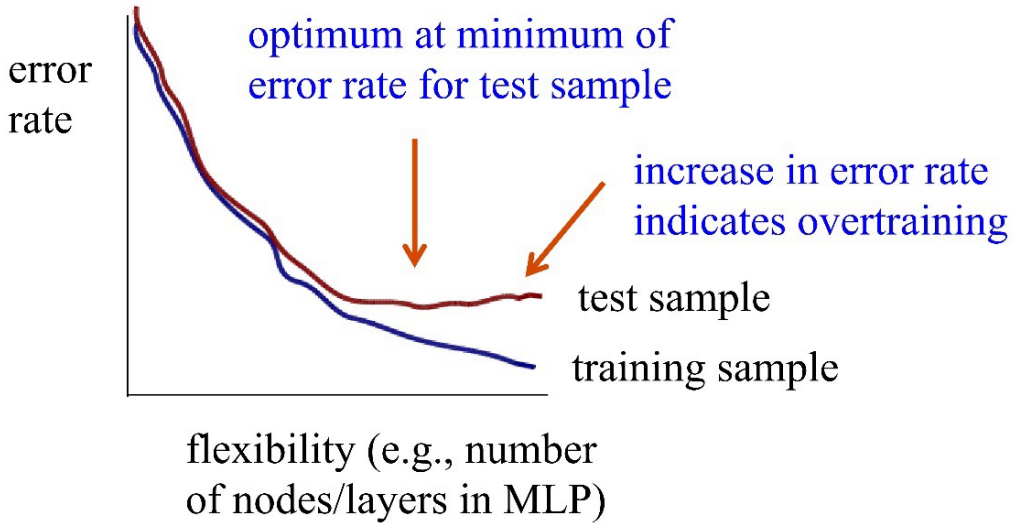
Including more parameters in a classifier makes its decision boundary increasingly flexible, e.g., more nodes/layers for a neural network.

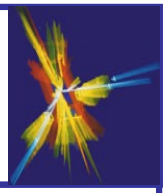
A "flexible" classifier may conform too closely to the training points; the same boundary will not perform well on an independent test data sample (→ "overtraining").



Monitoring overtraining

If we monitor the fraction of misclassified events (or similar, e.g., error function $E(\boldsymbol{w})$) for test and training samples, it will usually decrease for both as the boundary is made more flexible:





For multiclassification problems can work better in practice than e.g. neural networks

Decision trees

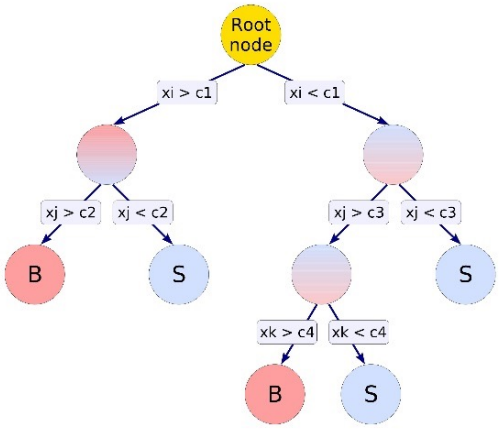
A training sample of signal and background data is repeatedly split by successive cuts on its input variables.

Order in which variables used based on best separation between signal and background.

Iterate until stop criterion reached, based e.g. on purity, minimum number of events in a node.

Resulting set of cuts is a 'decision tree'.

Tends to be sensitive to fluctuations in training sample.



sklearn implementation: Randomforestregressor

Boosted decision trees

Boosting combines a number classifiers into a stronger one; improves stability with respect to fluctuations in input data.

To use with decision trees, increase the weights of misclassified events and reconstruct the tree.

Iterate → forest of trees (perhaps > 1000). For the mth tree,

$$T_m(\vec{x}) = \begin{cases} 1 & \vec{x} \text{ in signal acceptance region} \\ -1 & \text{otherwise} \end{cases}$$

application: Jan Welti, PhD thesis in physics, HU-P-D235 (2017)

Define a score α_m based on error rate of mth tree.

Boosted tree = weighted sum of the trees: $T(\vec{x}) = \sum_m \alpha_m T_m(\vec{x})$

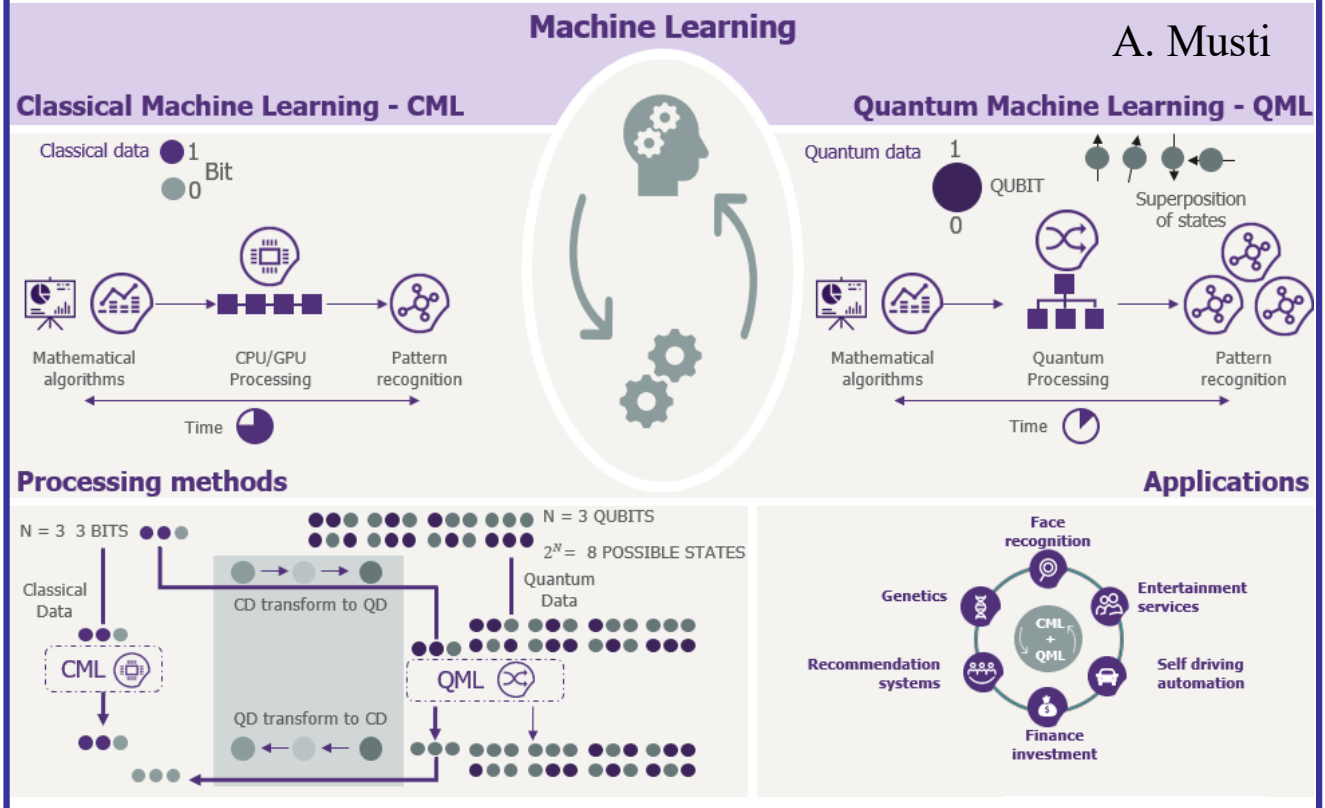
Algorithms: AdaBoost (Freund & Schapire), ε-boost (Friedman).

bagging = draw bootstraped samples from training sample, create tree for each & finally combine them

Quantum machine learning?

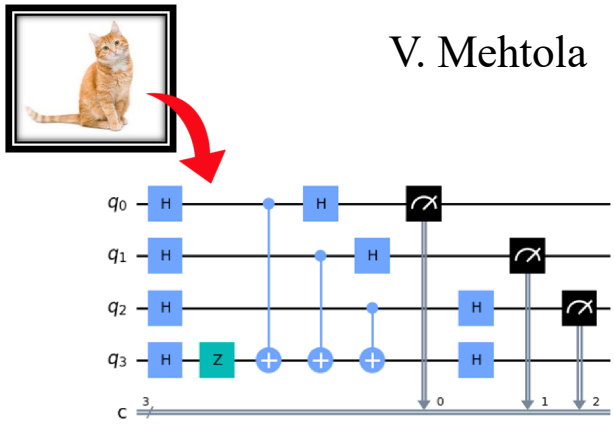
- When data points projected in higher and higher dimensions, becomes harder for a classical computer to deal with it. Even if it does, it takes too much time.
- **Sometimes, classical machine learning algorithms too taxing for classical computers.**
- **Solution: quantum computers?** They use super-position & entanglement to solve problems (potentially) much faster than their classical counterparts.

A. Musti



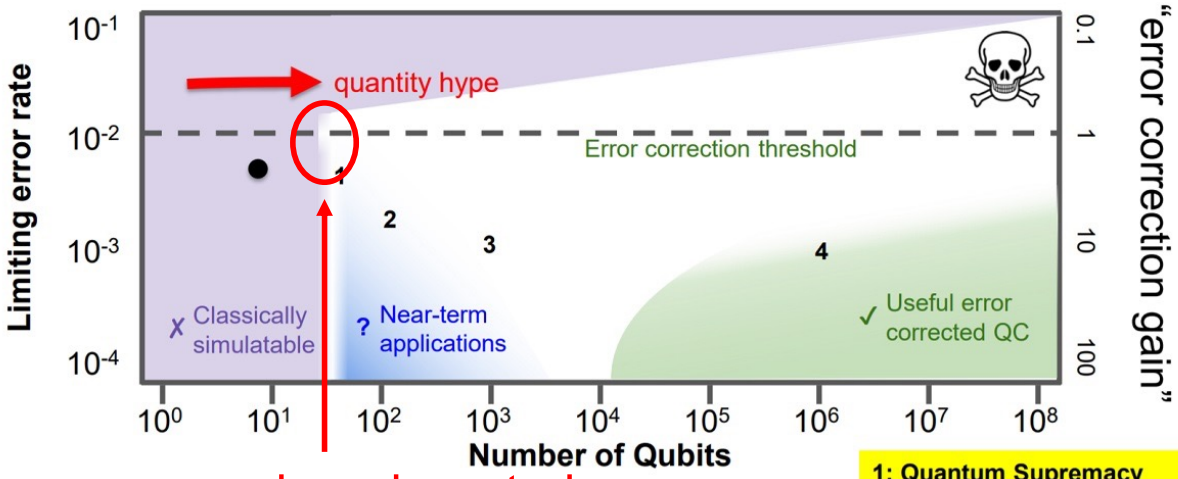
V. Mehtola

- Operations on qubits are unitary matrices
- One can encode classical data into the circuit by parametrizing these unitary operations
  - one can encode up to 2^n floats into the qubits
- Measurement leads to only an n-bit string (or n floats if running the circuit many times as one usually does)



Quantum machine learning?

Need Both Quality and Quantity



we are somewhere here today

Google strategy

1: Quantum Supremacy
2: Look for near-term apps
3: Error correction
4: Full QC

First trial in particle physics in Finland:
V. Mehtola MSc thesis 2025

quantum kernel for selecting Vector Boson Scattering events from background

$\Longrightarrow$

Need sufficient number of qubits to remove "noise" (= limit error rate) to be competitive with classical machine learning.

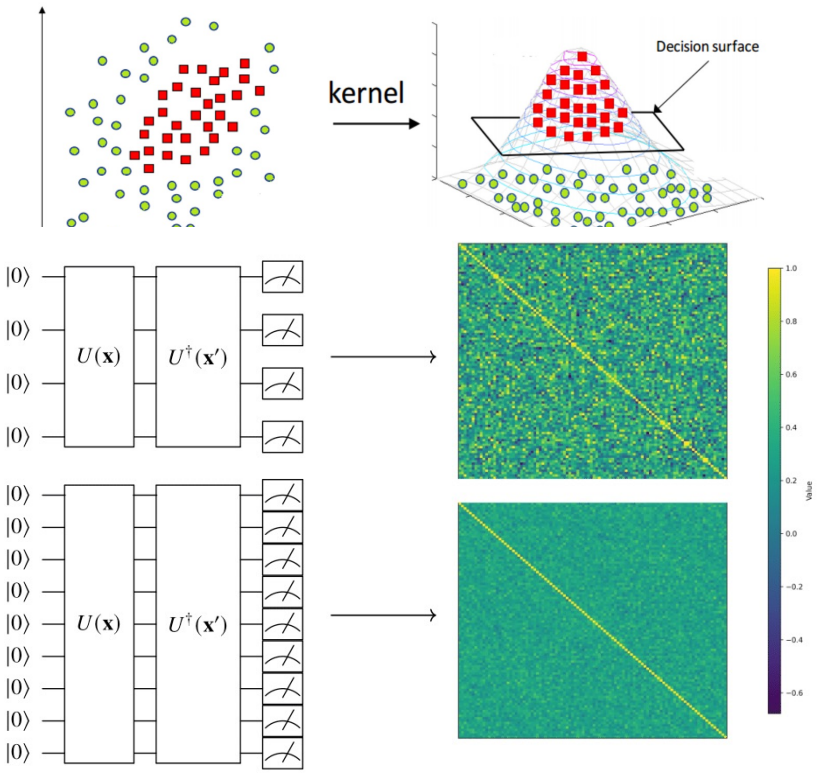
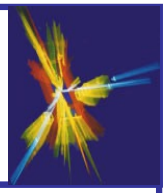


Figure 3: A figure for visualizing the general phenomenon of concentration (exponential or not). On the left, there are two quantum kernel circuits with same feature map scaled to four and nine qubits, respectively. To their right is their respective Gram matrix. For the nine-qubit version, off-diagonal variance is visibly lower. In the case that this scaling is exponential in the number of qubits, the feature map $U(\mathbf{x})$ does not scale well at least for the given data and pre-processing.



A simple example (2D)

Consider two variables, x_1 and x_2, and suppose we have formulas for the joint pdfs for both signal (s) and background (b) events (in real problems the formulas are usually not available).

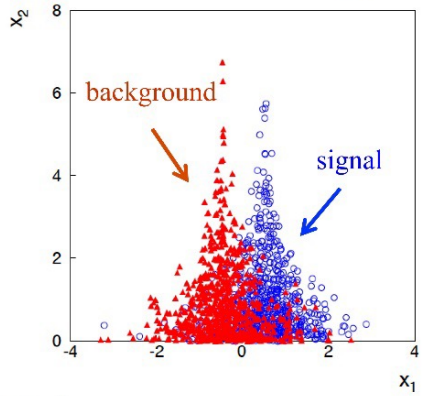
$f(x_1|x_2)$ ~ Gaussian, different means for s/b,
Gaussians have same σ, which depends on x_2,
$f(x_2)$ ~ exponential, same for both s and b,
$f(x_1, x_2) = f(x_1|x_2)\, f(x_2)$:

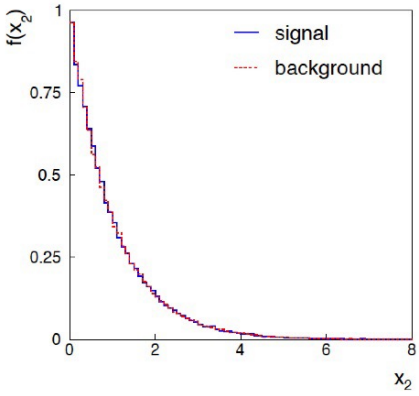
$$f(x_1, x_2|\mathrm{s}) = \frac{1}{\sqrt{2\pi}\sigma(x_2)} e^{-(x_1-\mu_\mathrm{s})^2/2\sigma^2(x_2)} \frac{1}{\lambda} e^{-x_2/\lambda}$$

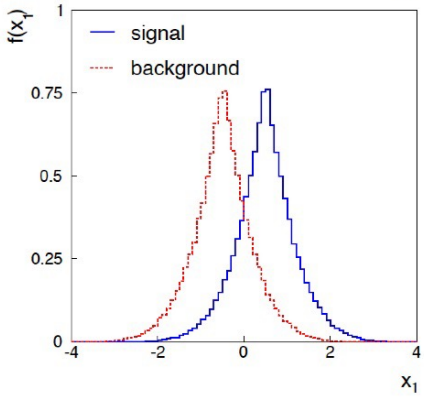
$$f(x_1, x_2|\mathrm{b}) = \frac{1}{\sqrt{2\pi}\sigma(x_2)} e^{-(x_1-\mu_\mathrm{b})^2/2\sigma^2(x_2)} \frac{1}{\lambda} e^{-x_2/\lambda}$$

$$\sigma(x_2) = \sigma_0 e^{-x_2/\xi}$$

Joint and marginal distributions of x_1, x_2

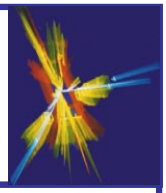






Distribution $f(x_2)$ same for s, b.

So does x_2 help discriminate between the two event types?



Likelihood ratio for 2D example

Neyman-Pearson lemma says best critical region is determined by the likelihood ratio:

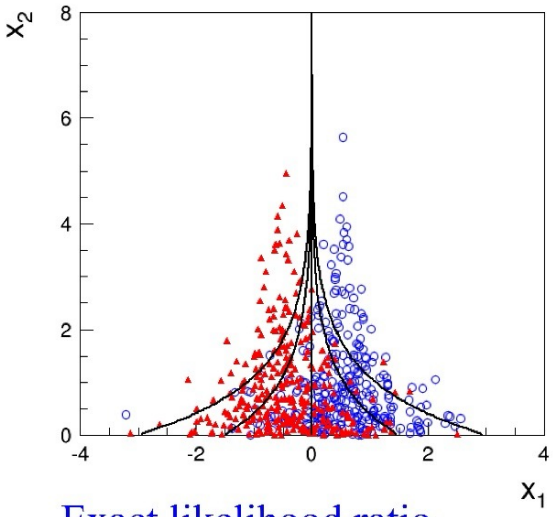
$$t(x_1, x_2) = \frac{f(x_1, x_2|\mathrm{s})}{f(x_1, x_2|\mathrm{b})}$$

Equivalently we can use any monotonic function of this as a test statistic, e.g.,

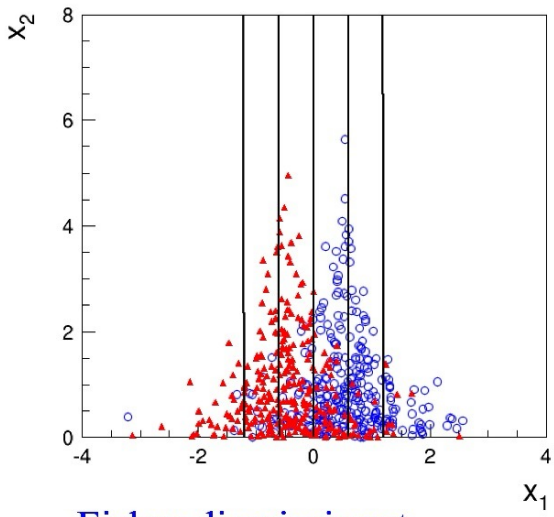
$$\ln t = \frac{\frac{1}{2}(\mu_\mathrm{b}^2 - \mu_\mathrm{s}^2) + (\mu_\mathrm{s} - \mu_\mathrm{b})x_1}{\sigma_0^2 e^{-2x_2/\xi}}$$

Boundary of optimal critical region will be curve of constant ln t, and this depends on x_2!

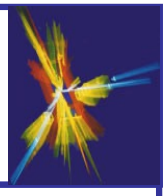
Contours of constant MVA output



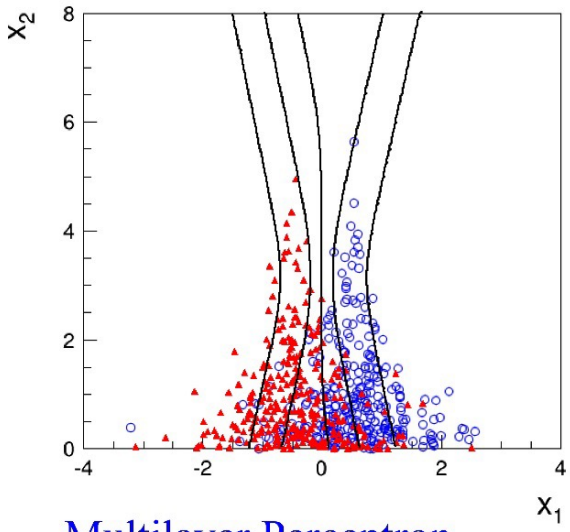
Exact likelihood ratio



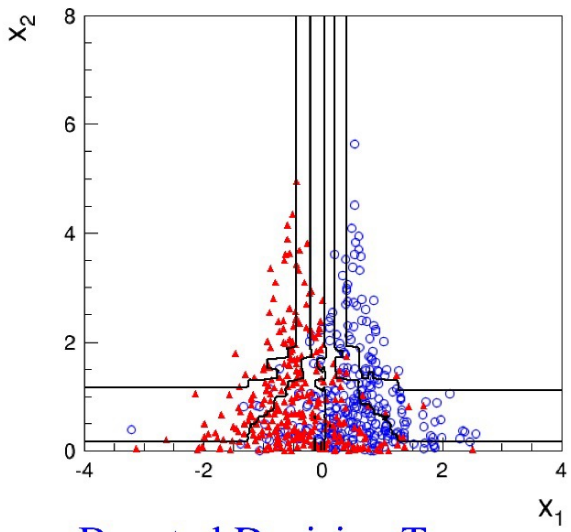
Fisher discriminant



Contours of constant MVA output



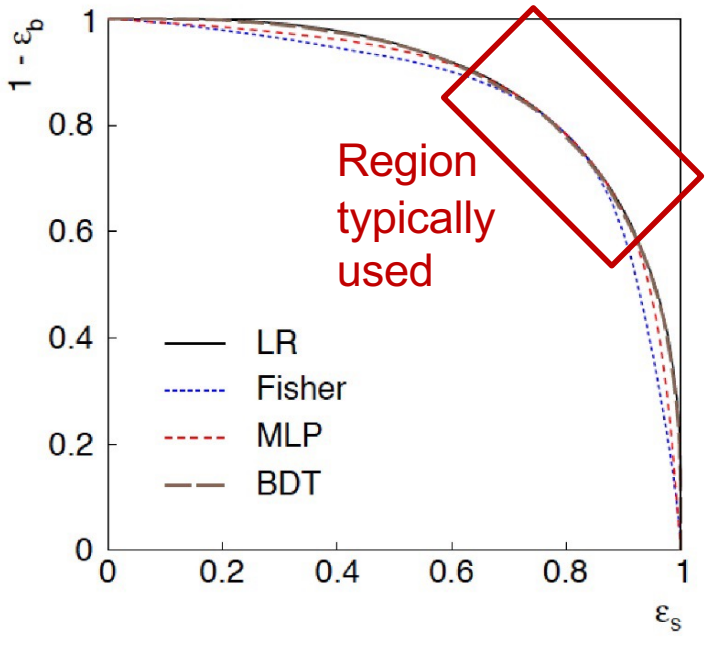
Multilayer Perceptron
1 hidden layer with 2 nodes



Boosted Decision Tree
200 iterations (AdaBoost)

Training samples: 10^5 signal and 10^5 background events

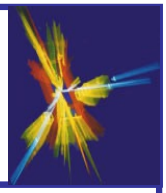
ROC curve



ROC = "receiver operating characteristic" (term from signal processing).

Shows (usually) background rejection ($1-\varepsilon_b$) versus signal efficiency ε_s.

Higher curve is better; usually analysis focused on a small part of the curve.



Multivariate analysis discussion

First clean the input of faulty, problematic or inadequate data
For all methods, need to check:

Sensitivity to statistically unimportant variables
(best to drop those that don't provide discrimination);

Level of smoothness in decision boundary (sensitivity to over-training)

Given the test variable, next step is e.g., select n events and estimate a cross section of signal: $\hat{\sigma}_s = (n - b)/\varepsilon_s L$

Now need to estimate systematic error...

If e.g. training (MC) data ≠ Nature, test variable is not optimal, but not necessarily biased.

But our estimates of background b and efficiencies would then be biased if based on MC. (True also for 'simple cuts'.)

But in a cut-based analysis it may be easier to avoid regions where untested features of MC are strongly influencing the decision boundary.

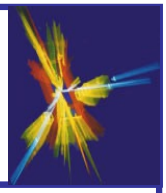
Look at control samples to test joint distributions of inputs.

Try to estimate backgrounds directly from the data (sidebands).

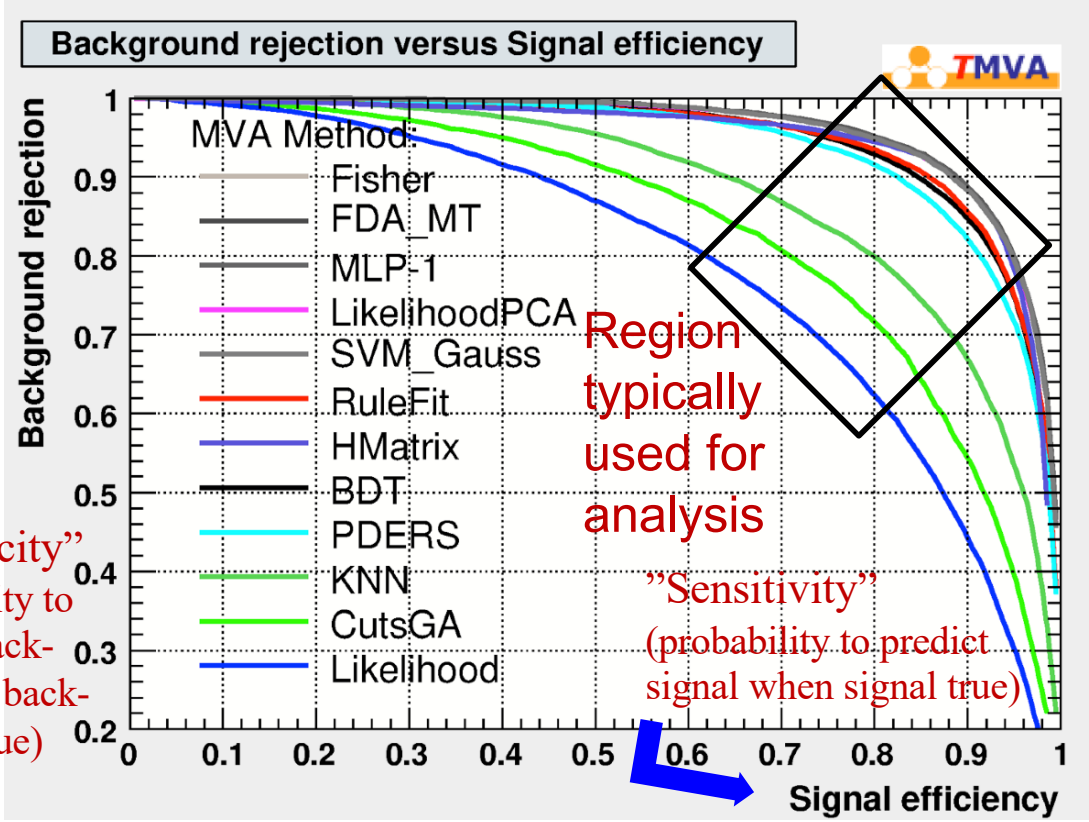
The purpose of the statistical test is often to select objects for further study and then measure their properties.

Need to avoid input variables that are correlated with the properties of the selected objects that you want to study. (Not always easy; correlations may be poorly known.)

Also decide whether "hard" classification (defining explicitly signal & background regions) is necessary or whether "soft" classification (that gives simply signal & background probabilities) is sufficient.

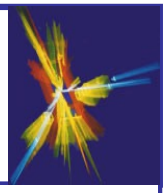


Toolkit for Multivariate Analysis (TMVA) included in ROOT-package enables usage of many techniques simultaneously:



Multivariate/machine learning techniques included:

- Neural networks
- Deep networks
- Multilayer perceptron (MLP)
- Boosted/Bagged decision trees (BDT)
- Support Vector Machine (SVM)
- Fisher and Function discriminant analysis (FDA)
- Multidimensional probability density estimation (PDE) range-search approach (PDERS)
- Multidimensional k-nearest neighbour classifier (KNN)
- Predictive learning via rule ensembles (RuleFit)
- Projective likelihood estimation (PDE approach)
- Rectangular cut optimisation



Resources on multivariate methods

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

PDG review on machine learning:
https://pdg.lbl.gov/2025/reviews/rpp2024-rev-machine-learning.pdf

Software for multivariate analysis

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

From **tmva.sourceforge.net**, also distributed with ROOT
Variety of classifiers
Good manual

https://root.cern.ch/

SKLEARN - Machine Learning in Python

- Simple and efficient tools for predictive data analysis
- Built on NumPy, SciPy, and matplotlib

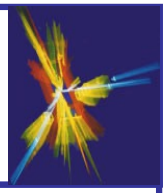
https://scikit-learn.org/stable/

KERAS – Python interface to many different neural network packages https://keras.io/

PyTorch – Python interface for neural networks in Torch programme package https://pytorch.org/

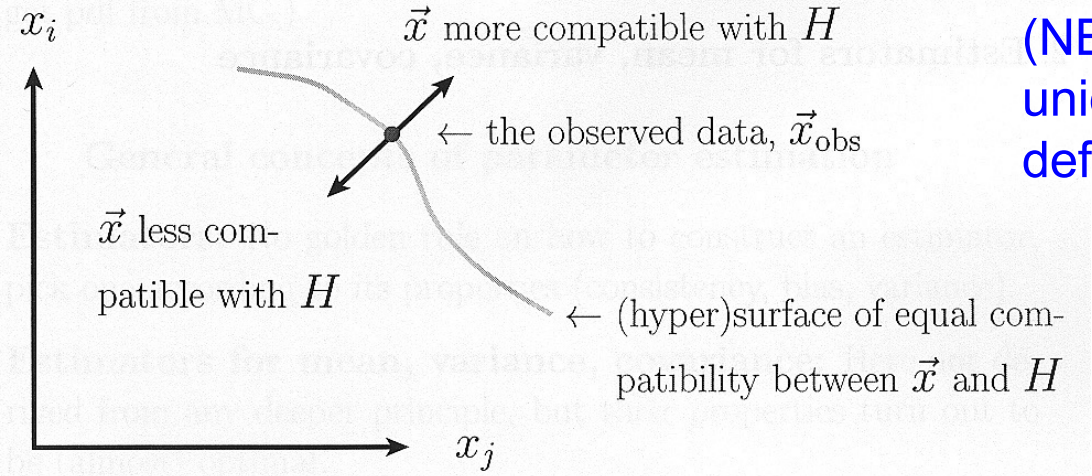
Dedicated course at University of Helsinki: DATA11002 Introduction to Machine Learning 5 ECTS (period 2/2025)

Goodness-of-fit tests



Want to substanciate how well hypothesis H compatible with observed data (without reference to an alternative).

Hypothesis H predict $f(\bar{x}|H)$ for data vector $\bar{x} = (x_1, \ldots x_n)$
Observe a single point in $\bar{x}$-space: $\bar{x}_{\text{obs}}$. What can be said about validity of H in light of data? → decide what parts of $\bar{x}$-space less compatible with H than observed point $\bar{x}_{\text{obs}}$.



(NB! no unique definition)

Construct test statistic $t(\bar{x})$ whose value reflects level of compatibility between $\bar{x}$ & H. e.g.
high $t \rightarrow$ data less compatible with H
low $t \rightarrow$ data more compatible with H.

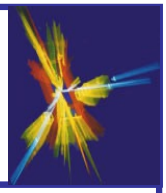
Express **goodness-of-fit** by giving ***P*-value:**
P = probability to observe data $\bar{x}$ (or $t(\bar{x})$) having equal or lesser compatibility with H than $\bar{x}_{\text{obs}}$ (or $t(\bar{x}_{\text{obs}})$).

NB! P-value not be interpreted as probability that H true !!
P-value also observed significance level/confidence level

If H true, then (for continuous $\bar{x}$) P uniform in [0,1]
If H not true, then pdf of P (usually) peaked close to 0.

Does small P-value really mean H is false? No, P-value = probability to obtain such a result "by chance" if H is true.

Goodness-of-fit example



Example: fit of charged particle trajectory in magnetic field. Primary aim, obtain parameters of trajectory ("helix"). Goodness-of-fit variables to determine compatibility of trajectory segments/hits to charged particle trajectory but also for checks of covariance matrix for helix parameters. Much simplified, if trajectory consists of N hits measured in $R\phi$ & M hits measured in z, then track fit ≈ to minimize:

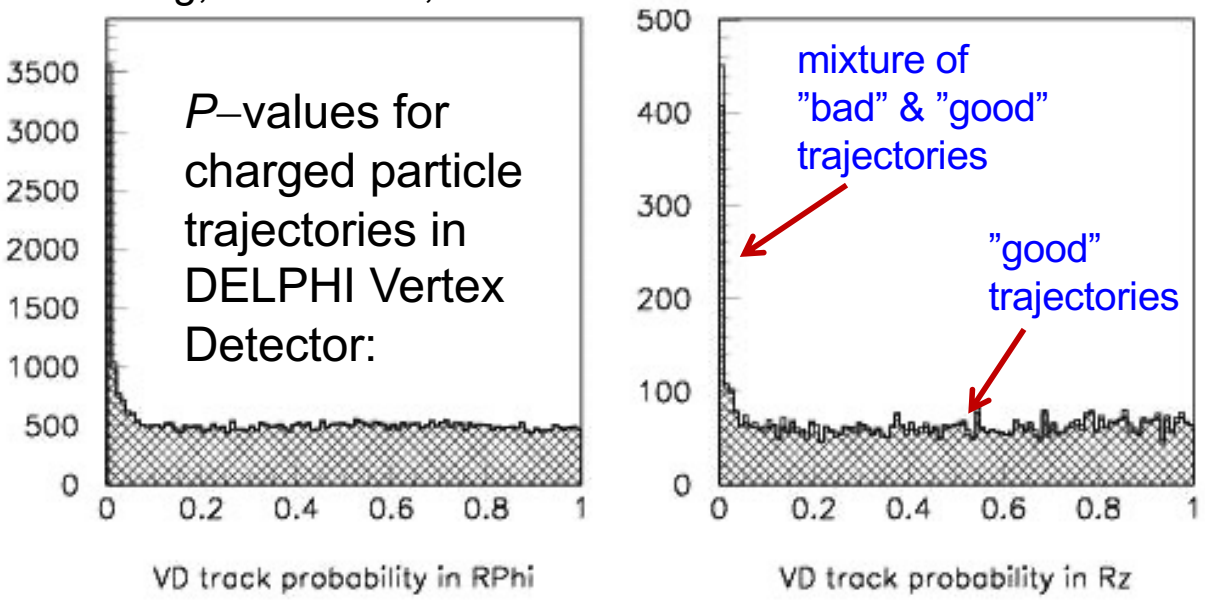
$$\chi^2 = \sum_{i=1}^{N} (R\phi_i - f_{R\phi}(R_i))^2 / \sigma(R\phi_i)^2 \;\&\; \chi^2 = \sum_{i=1}^{M} (z_i - f_z(R_i))^2 / \sigma(z_i)^2$$

where $f_{R\phi/z}(R)$ describes dependence w.r.t. to radius, R. Typical for particle physics. N_{dof} = N–3 (solenoidal field) & M–2. Here multiple scattering & energy loss ignored. Calculate corresponding P-value from observed χ^2 value:

$$P = \int_{\chi^2}^{\infty} f(z; N_{dof})\,dz$$

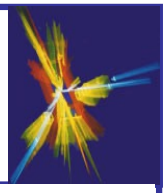
where $f(z;N_{dof})$ chi-square pdf for N_{dof} degrees of freedom.

K.Österberg, PhD thesis, HIP-1998-01



Distribution should be flat: raise towards 0 indicate that there is a class of hits with a slightly worse resolution & peak at 0 misassociated hits or badly reconstructed tracks.

Kolmogorov-Smirnov test



To make goodness-of-fit tests of distributions, the **Kolmogorov–Smirnov (K-S) test** is often used.

• By construction, it was developed for the comparison of functions, but is widely also used for binned data.

• The K-S test is predominantly sensitive to the shape of the distribution, much more than the χ^2–test. It's also normalization independent, which is not the χ^2–test.

Used mainly in three ways:

– test whether data sample follows certain distribution

– test whether two data sample could be consistent to come from the same distribution

– test whether data in two histograms are consistent with each other

- Kolmogorov distance is defined as:
  - $dist_K = Max|F_{parent}(x) - F_{test}(x)|$, where F is cumulative distribution for parent and test normalized to one.
  - distance is also know as Kolmogorov test statistic.
- Probability that $dist_K \geq X$ is given by Kolmogorov distribution function:

$$P(dist_k \geq x) = 2\sum_{i=1}^{\infty}(-1)^{i-1} e^{-2i^2x^2}$$

1

NB! true if n large (> ~10), where n is size of data sample.

Kolmogorov-Smirnov test

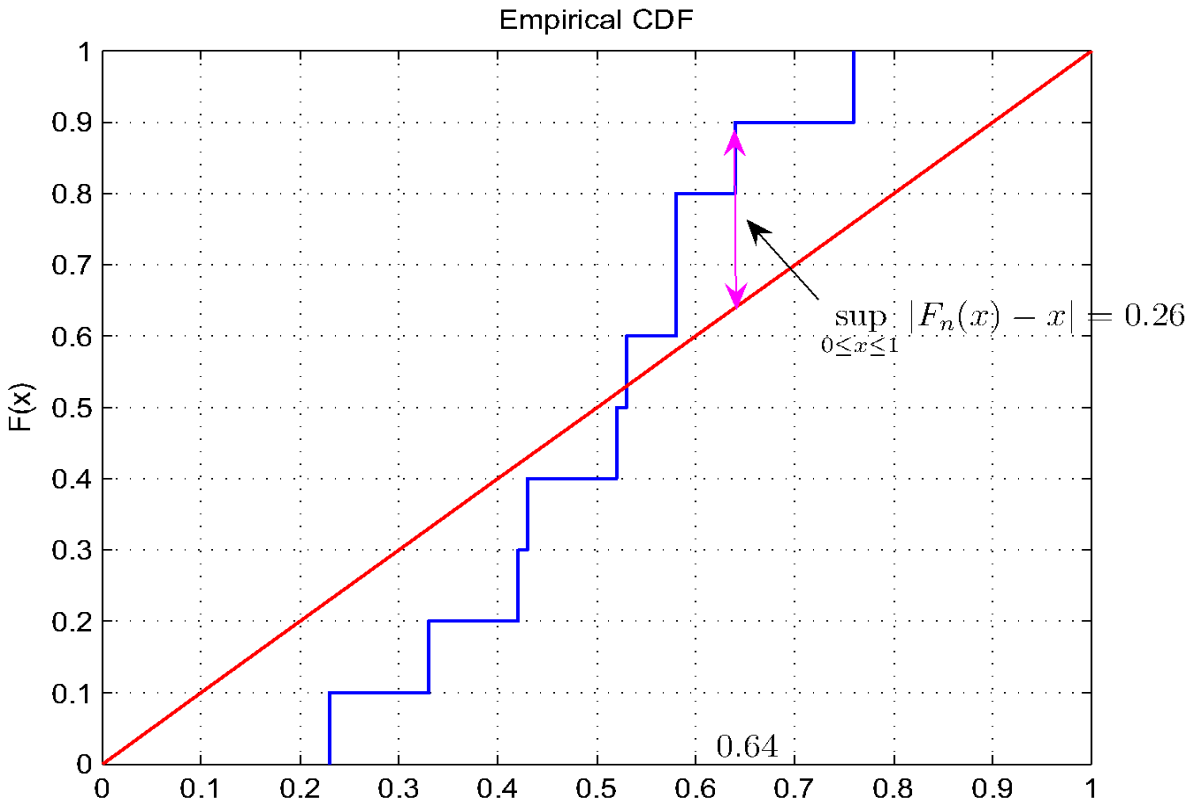
Example: 10 random numbers between 0 and 1:
0.58, 0.42, 0.52, 0.33, 0.43, 0.23, 0.58, 0.76, 0.53, 0.64

Are they uniformly distributed in [0,1]?

i.e. H_0: $F_{\text{test}}(x) = F_0(x) = x$

$$\max_{0\le x\le 1}\left|F_{data}(x) - F_{test}(x)\right| = \max_{1\le i\le n}\left(\left|F_n(X_i^-) - x\right|, \left|F_n(X_i^+) - x\right|\right)$$

cumulative distribution function value before/after i^{th} jump

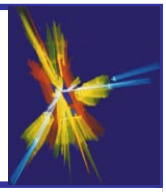


KS probability: $dist_k \to dist_k \cdot \sqrt{n}$ so here $0.26 \cdot \sqrt{10} \approx 0.82$

hypothesis testing at significance level $\alpha = 0.05$:

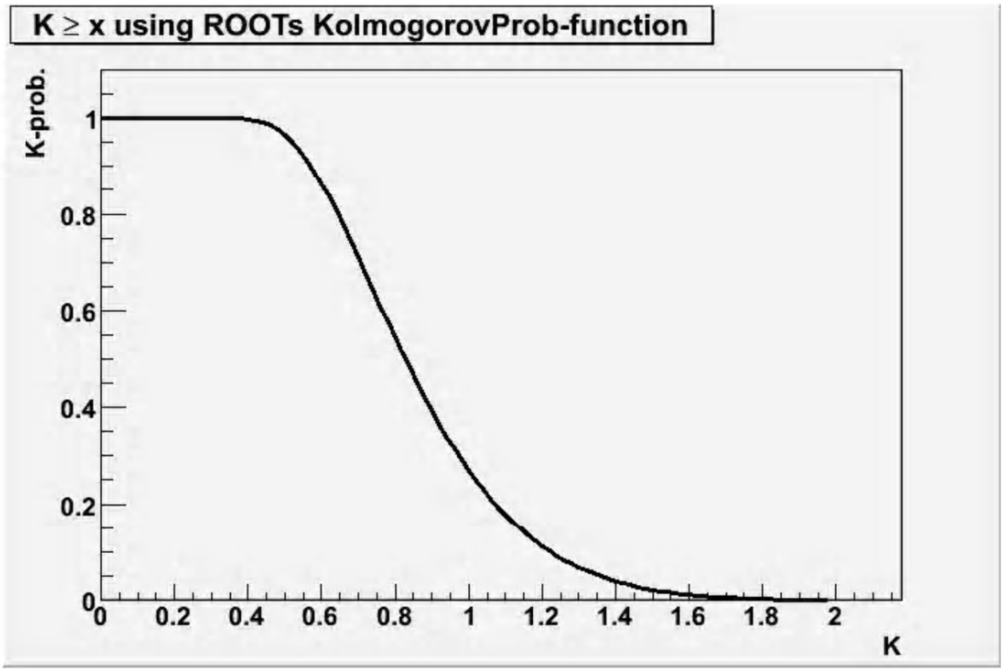
$1 - P(c \ge x) = 0.05 \Rightarrow c = 1.35$ ($0.82 \Rightarrow$ P-value = 17 %)

Here $dist_k \cdot \sqrt{n} = 0.82 < c = 1.35 \Rightarrow$ hence hypothesis random numbers uniform in [0,1] can not be rejected.



KS-test

- Kolmogorov distribution is slow to integrate and there are plenty of algorithms to calculate Kolmogorovs probability: eg. ROOTs TMath library, matlab, octave, mathematica, java...



- When comparing how two one-dimensional distibutions differ, Kolmogorov probability is calculated by replacing:

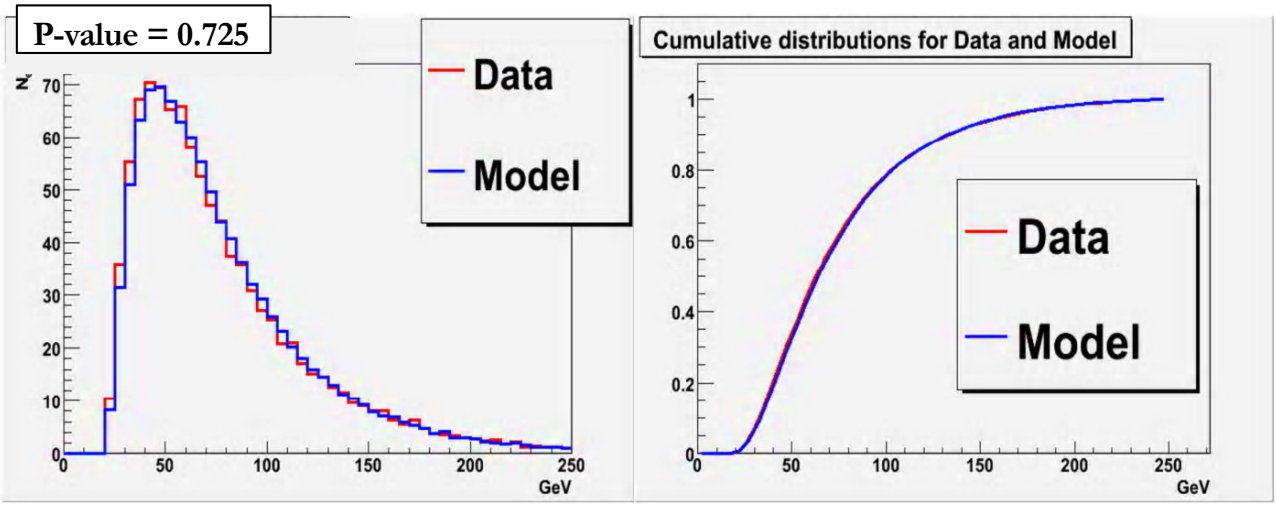
$$dist_K \rightarrow dist_K \sqrt{\frac{nn'}{n+n'}}$$

same when testing if two data sample originate from same distribution !!

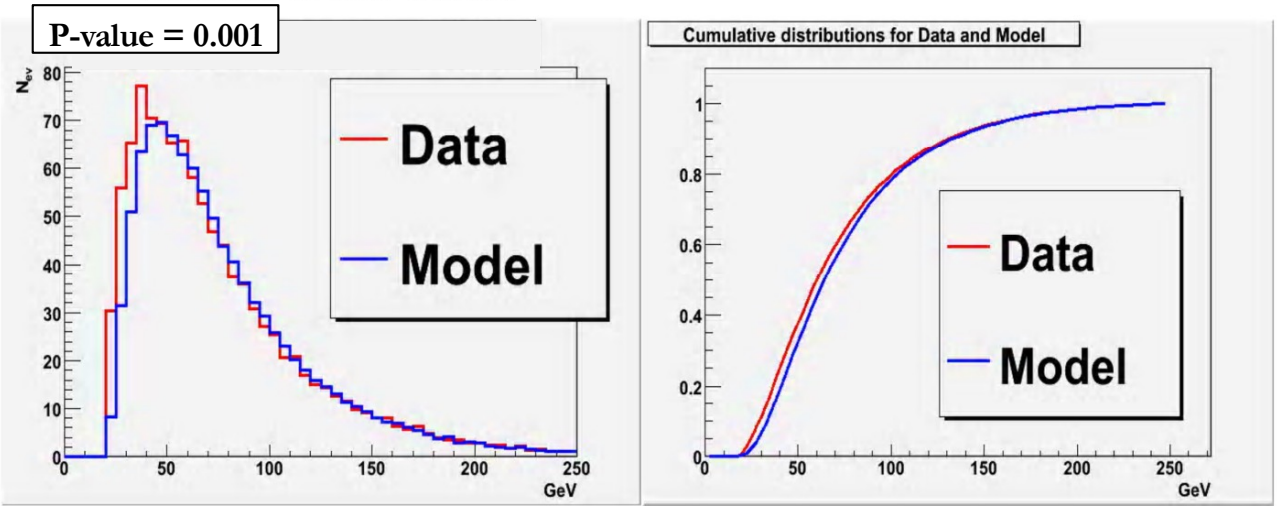
- where n and n' are the number of entries in parent [2] and test distributions.

KS-test

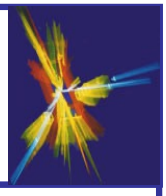
- P-value is calculated comparing how often simulated experiment has larger K_{dist} than K-distance is in data vs model.



- WARNING: Few bad bins can make your KS-value really bad! Check your distributions, understand your measurements! One bin with 4-sigma deviation can make your KS close to zero.
- Eg. Low energy region badly modelled. Otherwise distribution identical to one above.

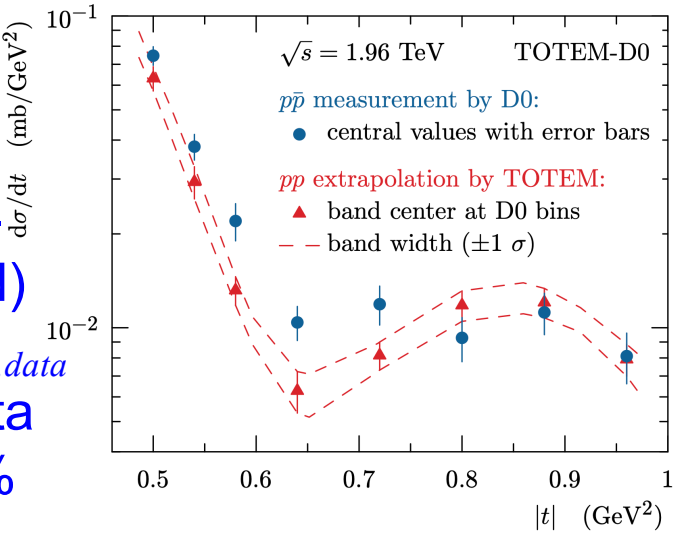


Kolmogorov-Smirnov test

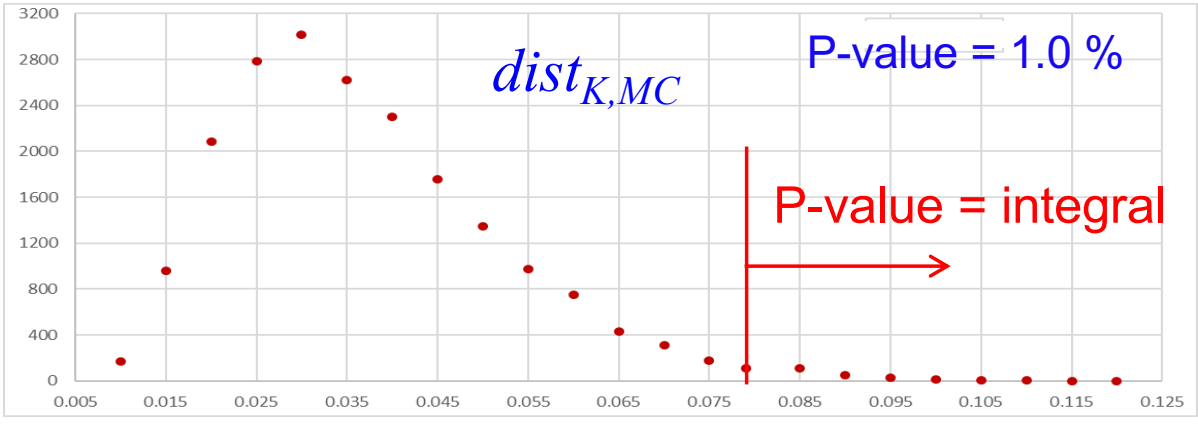


An additional usage of Kolmogorov-Smirnov is to test the compatibility of two sets of measurements with uncertainties using MC methods. In this case one tests whether the shape of the distributions of the two data sets are compatible with each other.

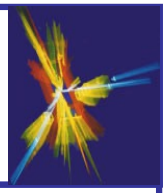
What is the probability that data set 1 (blue, taking its uncertainties into account) with uncer-tainties of data set 2 (red) would give $dist_K$ >= $dist_{K,data}$ i.e. the $dist_K$ between data set 1 & 2? Answer: 1.0 %



1. determine $dist_{K\,data}$ between data 1 and 2 (= 0.079)
2. generate data set 1' (1") by adding to the central value of each data set 1 point, a random gaussian according to point uncertainty in data set 1 (2)
3. calculate $dist_{K,MC}$ between data set 1' and 1"
4. Repeat step 2 and 3 to get a distribution of $dist_{K,MC}$
5. Calculate probability that $dist_{K,MC} >= dist_{K,data}$



Pearson's χ^2 test



Let's examine another goodness-of-fit test applicable to distribution of a variable x. Assume histogram of observed x-values in N bins. Suppose number of entries in bin i is n_i & number of expected entries ν_i, then the test statistic

$$\chi^2 = \sum_{i=1}^{N} (n_i - \nu_i)^2 / \nu_i$$

reflects level of agreement between observed & expected histogram. Above test based on **Pearson's χ^2 statistic.**
If data $\bar{n} = (n_1, \ldots, n_N)$ consists of N independent Poisson variables with mean values $\bar{\nu} = (\nu_1, \ldots, \nu_N)$ & all ν_i not too small (rule of thumb: all $\nu_i \geq 5$) then test statistic χ^2 follow chi-square pdf for N degrees of freedom. Holds regardless of distribution for variable x (χ^2 test distribution free).

Standard deviation of Poisson distribution $\sqrt{\nu_i}$ so χ^2 test sum of difference squared between observed & expected values, measured in units of standard deviations squared.

Corresponding P-value given by the observed χ^2 using

$$P = \int_{\chi^2}^{\infty} f(z; N)\, dz$$

where $f(z;N)$ chi-square pdf for N degrees of freedom.

(google "chi square calculator" to get cumulative χ^2 function)
Recall that for chi-square pdf, expectation value $E[z] = N$ $\rightarrow$ often give χ^2/N as a measure of level of agreement. However better to give χ^2 & N separately...

$\chi^2 = 15$, $N = 10$ $\rightarrow$ P-value = 13 %
$\chi^2 = 150$, $N = 100$ $\rightarrow$ P-value = 0.09 %

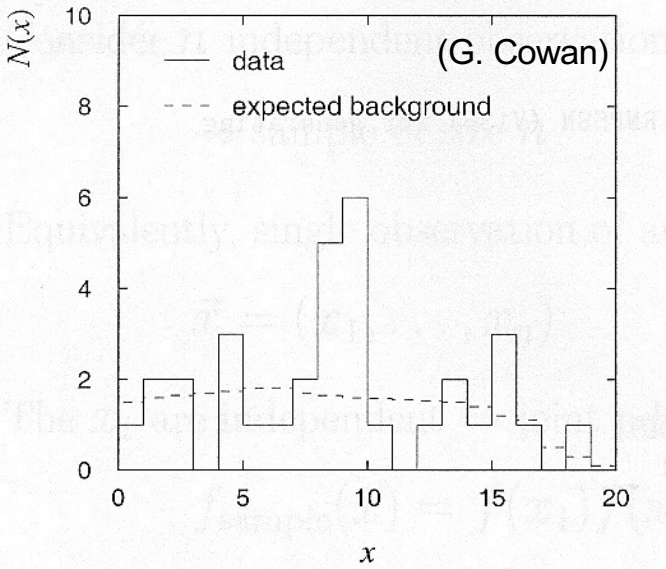
If $n_{\text{tot}} = \sum_{i=1}^{N} n_i$ fixed then n_i's multinomial with $\nu_i = p_i\, n_{\text{tot}}$ &

$$\chi^2 = \sum_{i=1}^{N} (n_i - p_i n_{\text{tot}})^2 / p_i n_{\text{tot}}$$

follows chi-square pdf with N–1 dof (if all $p_i\, n_{\text{tot}} \gg 1$).

Pearson's χ^2 test

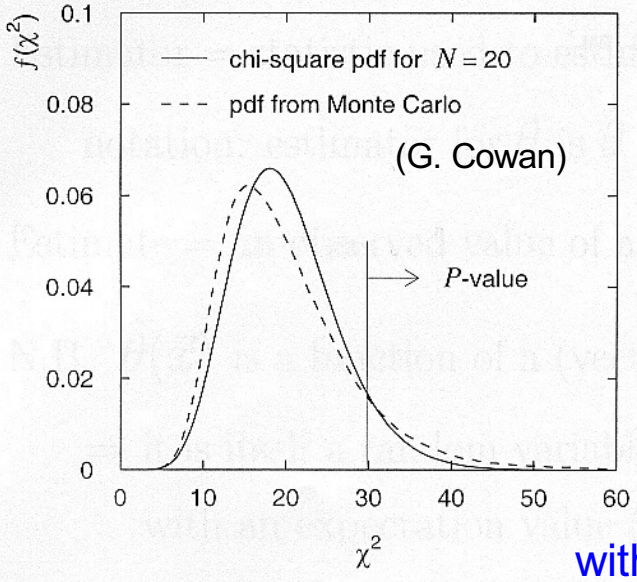
Even if amount of data too small for requirement all $\nu_i \geq 5$ to be fulfilled, one can still construct χ^2 statistic as long as all ν_i's > 0. Will no longer follow chi-square distribution & χ^2 statistic distribution will depend on pdf of variable x. e.g.



histogram (to the left) with peak gives χ^2 = 29.8 for N = 20 dof.

but ...all bins have ν_i < 5. here χ^2 statistic will not follow a normal chi-square distribution.

Pearson's χ^2 still used as test statistic, but must generate corresponding $f(\chi^2)$ using MC methods to get P-value.
- generate n_i from Poissonian with mean ν_i, for i = 1, ..., N,
- compute & record χ^2 value into a histogram,
- repeat steps above many times to get $f(\chi^2)$,
- compute P-value as probability of $\chi^2 \geq \chi^2_{obs}$ using $f(\chi^2)$.



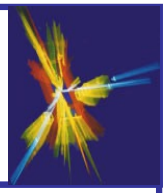
using $f(\chi^2)$ from MC:
P-value = 11 %

using chi-square pdf:
P-value = 7.3 %

The χ^2 test not very sensitive to presence of peaks.

NB! binning ambiguity of χ^2 test. P-value can vary significantly with binning for small data samples.

Significance of a signal



A simple type of goodness-of-fit test carried out to judge whether a discrepancy between data & expectation is sufficiently significant to merit a claim for a new discovery.

One observes n events; these can then consist of:

n_b events from a known process (background)
n_s events from a possible new process (signal)

If n_b & n_s Poisson random variables with means ν_b & ν_s $\Rightarrow$ $n = n_s + n_b$ also Poissonian with mean $\nu = \nu_b + \nu_s$:

$$P(n;\nu_s,\nu_b) = \frac{(\nu_s+\nu_b)^n}{n!} e^{-(\nu_s+\nu_b)}$$

Suppose $\nu_b = 0.5$ & one observes $n_{\text{obs}} = 5$.
Should one claim evidence for a new discovery?
hypothesis H: $\nu_s = 0$, i.e. only background events present.

$$P\text{-value} = P(n \geq n_{\text{obs}}) = \sum_{n=n_{\text{obs}}}^{\infty} P(n;\nu_s = 0,\nu_b) =$$

$$1 - \sum_{n=0}^{n_{\text{obs}}-1} \frac{\nu_b^n}{n!} e^{-\nu_b} = 1.7 \cdot 10^{-4} \quad (\neq P(\nu_s = 0)!)$$

probability for $\nu_s = 0$ hypothesis

Typical misunderstandings:
a misleading (but often occuring) estimate...

estimate for mean ν: $n_{\text{obs}} = 5$
estimated standard deviation of n: $\sqrt{n} = 2.2$ $\Rightarrow$

"measured signal" = $n_{\text{obs}} - \nu_b = 4.5 \pm 2.2$ i.e. ~ 2σ from 0.

Wanted: probability for Poisson variable with mean ν_b = 0.5 to give 5 or more (answer: 0.017 %)
Misunderstanding implied: probability for variable with μ = 4.5 & σ = 2.2 to give 0 or less (Gaussian answer: 2.2 %)

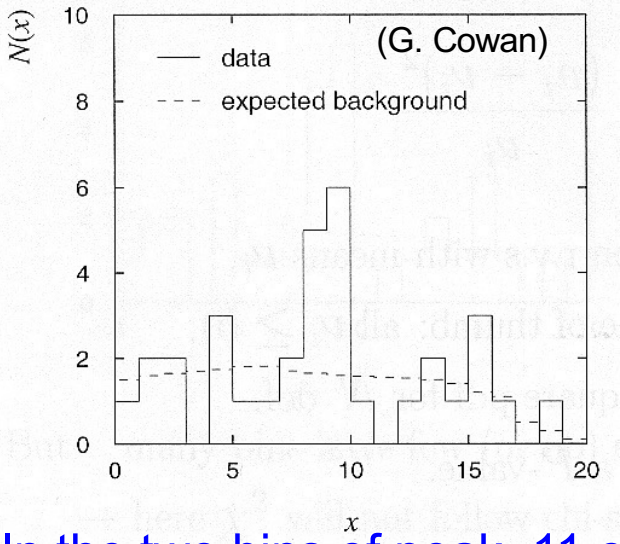
NB! difference disappear when ν_b »1, i.e. Poisson ~ Gaussian.

Significance of a signal

another pitfall: believe ν_b to be God-given; in reality must study influence of systematic uncertainties on ν_b. if e.g. $\sigma(\nu_b \text{ syst.}) = 0.3 \Rightarrow P(n \geq 5;\ \nu_b = 0.8,\ \nu_s = 0) = 0.14$ %
$\Rightarrow$ conservative approach: report P-value corresponding to $\pm 1\sigma$ in disfavourable direction of result, here $P = 0.14$ %

When can one claim observation? Often require P-value corresponding to 5σ (3σ) of a Gaussian distribution for new ("known") phenomena i.e. $\leq 2.9 \cdot 10^{-7}$ ($\leq 1.35 \cdot 10^{-3}$).
Particle physics convention: $>5\sigma$ = discovery, $>3\sigma$ = evidence

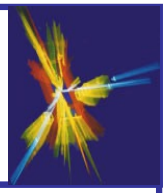
Significance of a peak:
In addition to counting events, one measures also x for each event; data viewed as a histogram as function of x:



histogram of observed data & theoretical expectation. Each bin a Poisson variable, since each x value independent (or multinomial if the expectation is normalised to the total number of observed events).

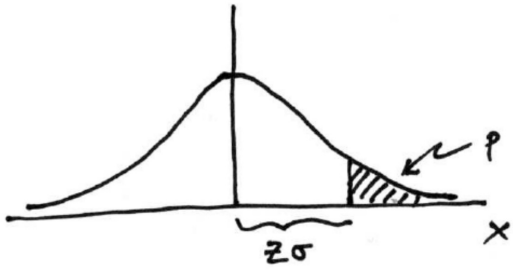
Is the peak significant?

In the two bins of peak, 11 observed events, $\nu_b = 3.2$,
$P(n \geq 11;\ \nu_b = 3.2,\ \nu_s = 0) = 0.05$ %
but ... did one know where to look for the peak? $\rightarrow$ give $P(n \geq 11)$ in any 2 adjacent bins ("look elsewhere" effect).
Is observed width consistent with expected x resolution?
How many bins $\times$ distributions did one examine?
Did one adjust selections to "enhance" the peak?
What about the side-bands of the peak, are they too low?



Significance from *p*-value

Often define significance Z as the number of standard deviations that a Gaussian variable would fluctuate in one direction to give the same p-value.



$$p = \int_Z^\infty \frac{1}{\sqrt{2\pi}} e^{-x^2/2}\, dx = 1 - \Phi(Z)$$ `1 - TMath::Freq`

$$Z = \Phi^{-1}(1-p)$$ `TMath::NormQuantile`

E.g. $Z = 5$ (a "5 sigma effect") corresponds to $p = 2.9 \times 10^{-7}$.

Search for the signal process

But what if the signal process is not known to exist and we want to search for it. The relevant hypotheses are therefore

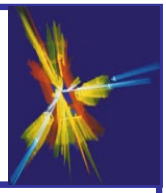
H_0: all events are of the background type
H_1: the events are a mixture of signal and background

Rejecting H_0 with $Z > 5$ constitutes "discovering" new physics.

Suppose that for a given integrated luminosity, the expected number of signal events is s, and for background b.

The observed number of events n will follow a Poisson distribution:

$$P(n|b) = \frac{b^n}{n!} e^{-b} \qquad P(n|s+b) = \frac{(s+b)^n}{n!} e^{-(s+b)}$$



Likelihoods for full experiment

We observe n events, and thus measure n instances of $\boldsymbol{x} = (x_1, x_2)$.

The likelihood function for the entire experiment assuming the background-only hypothesis (H_0) is

$$L_b = \frac{b^n}{n!} e^{-b} \prod_{i=1}^{n} f(\mathbf{x}_i|\mathrm{b})$$

and for the "signal plus background" hypothesis (H_1) it is

$$L_{s+b} = \frac{(s+b)^n}{n!} e^{-(s+b)} \prod_{i=1}^{n} (\pi_\mathrm{s} f(\mathbf{x}_i|\mathrm{s}) + \pi_\mathrm{b} f(\mathbf{x}_i|\mathrm{b}))$$

where π_s and π_b are the (prior) probabilities for an event to be signal or background, respectively.

Likelihood ratio for full experiment

We can define a test statistic Q monotonic in the likelihood ratio as

$$Q = -2 \ln \frac{L_{s+b}}{L_b} = -s + \sum_{i=1}^{n} \ln \left(1 + \frac{s}{b} \frac{f(\mathbf{x}_i|\mathrm{s})}{f(\mathbf{x}_i|\mathrm{b})}\right)$$

To compute p-values for the b and s+b hypotheses given an observed value of Q we need the distributions $f(Q|\mathrm{b})$ and $f(Q|\mathrm{s+b})$.

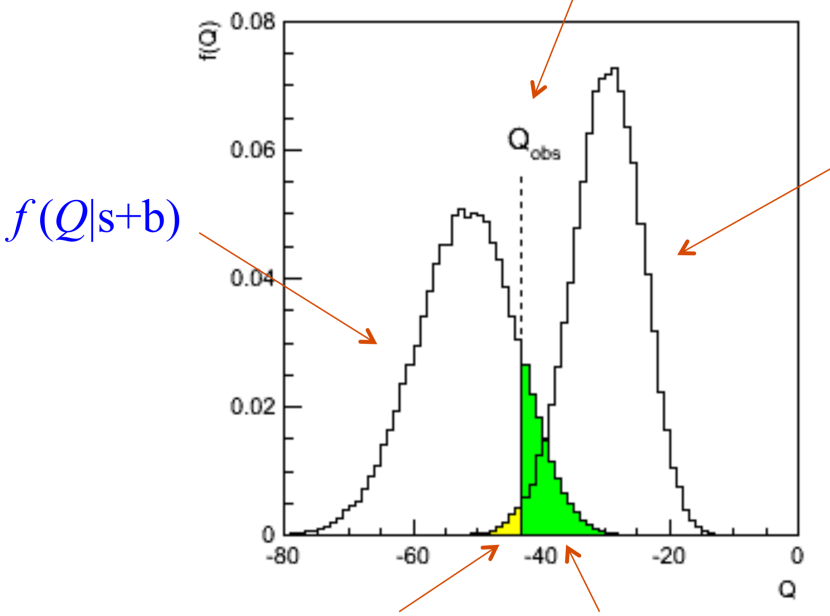
Note that the term $-s$ in front is a constant and can be dropped.

The rest is a sum of contributions for each event, and each term in the sum has the same distribution.

Distribution of Q

Take e.g. b = 100, s = 20.

Suppose in real experiment Q is observed here.



$f(Q|\text{s+b})$

$f(Q|\text{b})$

p-value of b only

p-value of s+b

Systematic uncertainties

Up to now we assumed all parameters were known exactly.

In practice they have some (systematic) uncertainty.

Suppose e.g. uncertainty in expected number of background events b is characterized by a (Bayesian) pdf $\pi(b)$.

Maybe take a Gaussian, i.e.,

$$\pi(b) = \frac{1}{\sqrt{2\pi}\sigma_b} e^{-(b-b_0)^2/2\sigma_b^2}$$

where b_0 is the nominal (measured) value and σ_b is the estimated uncertainty.

Distribution of Q with systematics

To get the desired p-values we need the pdf $f(Q)$, but this depends on b, which we don't know exactly.

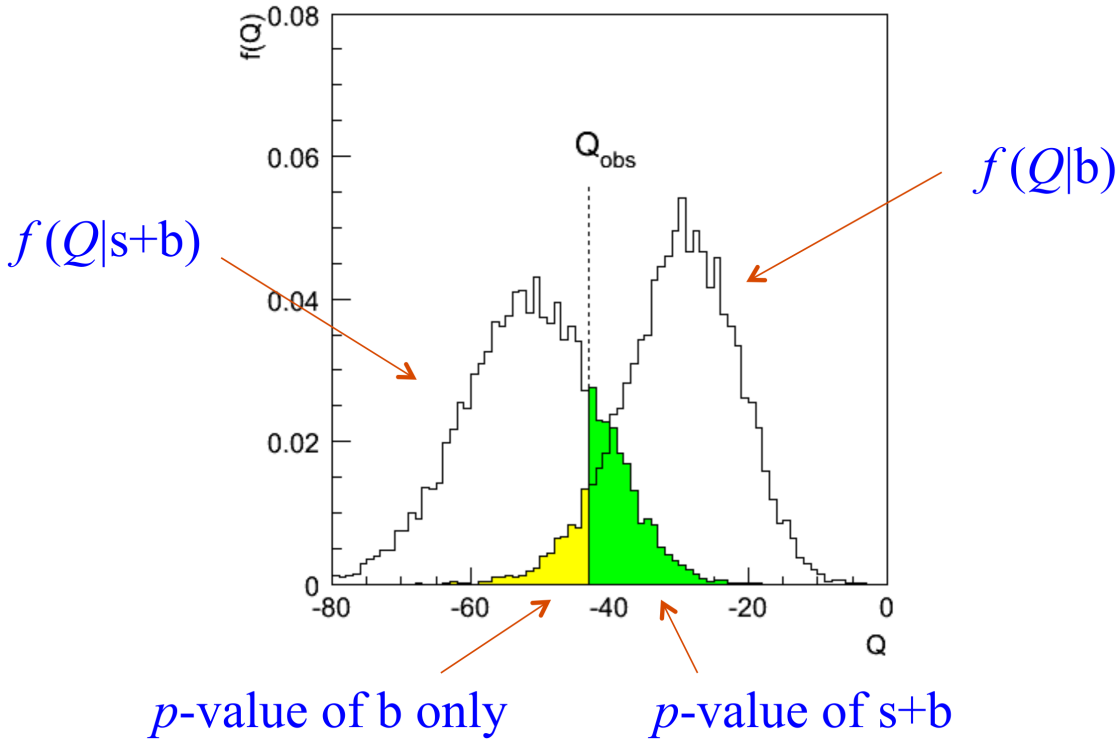
But we can obtain the Bayesian model average:

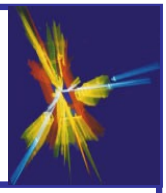
$$f(Q) = \int f(Q|b)\pi(b)\,db$$

With Monte Carlo, sample b from $\pi(b)$, then use this to generate Q from $f(Q|b)$, i.e., a new value of b is used to generate the data for every simulation of the experiment.

This broadens the distributions of Q and thus increases the p-value (decreases significance Z) for a given Q_{obs}.

For $s = 20$, $b_0 = 100$, $\sigma_b = 10$ this gives





What if number of data samples to compare is large?

Can test each sample pair with t– (basic method to test if two samples have same mean or not), χ^2 or KS-test… but easily get into trouble since probability of (at least) one pair to differ significantly naturally gets high (like ”peak” search in large # of histograms, ”peaks” are found).

Analysis of Variance (ANOVA): collection of statistical methods to treat comparison of 3 or more data samples. Originally developed by R. Fisher in 1920’s and 30’s.

Comparison can be made on one or several variables (one-way ANOVA and two(three)-way or factorial ANOVA)

Typical assumptions: independent samples, normal distributed samples, equality of variances…

Logic of ANOVA:

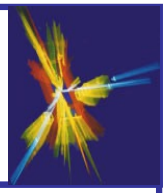
$$V_{tot}[x] = V_{stat}[x] + V_{diff\ sample}[x]$$

$$\mathrm{ndf}_{tot}[x] = \mathrm{ndf}_{stat}[x] + \mathrm{ndf}_{diff\ sample}[x]$$

ndf of χ^2 distribution describing sum of variances

$$F = \frac{\text{variance of sample means}}{\text{mean of within-sample variances}}$$

variable F follows F–distribution.Cumulative F–distribution to get P–value. If replace ”mean of within-sample variances” with ”expected variation of sample means” then F follows χ^2 distribution with ndf = # of samples –1



Example 11.1 *A barmaid at the Bull's Head is working to pay her way through university. To get a little intellectual exercise while pulling pints, she keeps track of the drinking habits of the various UMIST departments whose staff drink in the pub. Here are her findings from the previous week:*

Physics	Maths	Chemistry
21	6	8
16	10	6
17	13	4
18	13	5
	4	2

The data are usually arranged in columns, as in the table above:

Total number of staff:

$$N = \sum_{j=1}^{r} n_j.$$

Column 1	Column 2	...	Column r
$x_{1,1}$	$x_{1,2}$	...	$x_{1,r}$
$x_{2,1}$	$x_{2,2}$	...	$x_{2,r}$
$\vdots$	$\vdots$		$\vdots$
$\boldsymbol{x_{n1,1}}$	$\boldsymbol{x_{n2,2}}$		$\boldsymbol{x_{nr,r}}$

Then one does the following.

a) Compute the mean of each column separately:

$$m_j = \frac{\sum_{k=1}^{n_j} x_{k,j}}{n_j}.$$

"column" = sample

b) Compute the mean of all the entries in the table

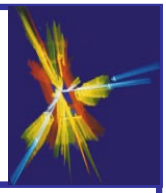
$$m = \frac{\sum_{j=1}^{r} \left[\sum_{k=1}^{n_j} x_{k,j}\right]}{N} = \frac{\sum_{j=1}^{r} n_j m_j}{N}.$$

c) Compute a measure of the variation *between* the columns. It is called SS_b, the "sum of squares between columns", and defined by

$$SS_b = \sum_{j=1}^{r} n_j (m_j - m)^2.$$

d) Compute a measure of the variation *within* the various columns. It is called SS_w, the "sum of squares within the columns", and is given by

$$SS_w = \sum_{j=1}^{r} \left[\sum_{k=1}^{n_j} (x_{k,j} - m_j)^2\right]$$



It's messy, but not difficult, to show that the two sums-of-squares account for all the variation in the whole table:

$$SS_t = SS_b + SS_w = \sum_{j=1}^{r}\left[\sum_{k=1}^{n_j}(x_{k,j}-m)^2\right]$$

Here the subscript 't' stands for "total"; SS_t is the sum-of-squared-deviations from the mean for the entire table.

One then assembles these results into a standard table that looks like:

Type of Variation	*Sum of Squares*	*deg. of freedom*	*Average sum of squares (estimated variances)*
Between Samples	SS_b	$(r-1)$	$SS_b/(r-1)$
Within Samples	SS_w	$\sum_{j=1}^{r}(n_j-1)$ $=(N-r)$	$SS_w/(N-r)$
Total	SS_t	$(N-1)$	—

Finally, one does a test to see whether the mean variation between columns, $SS_b/(r-1)$ is significantly bigger than the mean variation within columns $SS_w/(N-r)$. In the usual way one computes a statistic,

$$F = \frac{SS_b/(r-1)}{SS_w/(N-r)},$$

Find P-value: google "F-distribution applet"

then looks the value up (in special tables) to see if it is large enough to reject the null hypothesis. These tables depend on the number of degrees of freedom in the sums, so one consults the table for $F((r-1),(N-r))$.

Example of beer drinking:

$N = 14$, $r = 3$

Type of Variation	*Sum of Squares*	*deg. of freedom*	*Average sum of squares (estimated variances)*
Between Samples	386.9	2	193.5
Within Samples	102	11	8.5
Total	488.9	13	—

$\Rightarrow$ P-value = $1.2 \cdot 10^{-4}$

and the test stat. is $F \approx 22.8$. The attached table shows that if $F(2,12)$ exceeds 3.89 we can reject the null with 95% confidence, so we can safely conclude that the members the various groups do not *drink the same amount in the Bull's Head.*

Scheffe test statistic: expected maximal statistical difference (MSD) of two samples (if the underlying distribution is the same):

$$\text{MSD} = \sqrt{(r-1) * \overline{V[x]} * F\,(P=0.05) * \left(\frac{1}{n_1}+\frac{1}{n_2}\right)}$$