Goodness of fit.

Assuming the predictions correctly model the data, and assuming the measurements are normally distributed with covariance matrix V, the χ^2 statistic will have a p.d.f. given by

$$f(x) = \frac{1}{\Gamma(k/2)} x^{k/2-1} e^{-x/2}$$

where N is the number of degrees of freedom. If there are N measurements that are uncorrelated and M parameters in the model that are fit by minimizing χ^2 then $k = N - M$.

The probability of obtaining χ^2 as large as the one observed purely by chance is

$$P(\chi^2; k) = \int_{\chi^2}^{\infty} f(x)\,dx$$

$$= 1 - \int_0^{\chi^2} f(x)\,dx$$

$$= 1 - \gamma\left(\frac{k}{2}, \frac{\chi^2}{2}\right)$$

where γ is the normalized γ-function:

$$\gamma(a, x) = \frac{1}{\Gamma(a)} \int_0^x t^{a-1} e^{-t}\,dt$$

Is adding a parameter to a model significant?

$\bar{\chi}^2 = k$.

If we add a parameter then $\chi^{2\prime}$ must not be greater than the previous value.

In general, if $\Delta\chi^2 \gg 1$ then adding the parameter is justified.

The model with more parameters might be correct but is not distinguisable from the model with fewer parameters.

With additional measurements it may become possible to make a distinction.

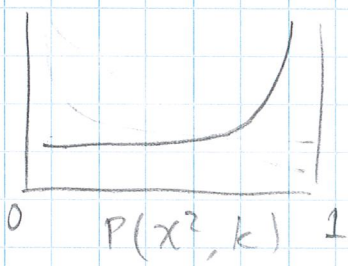
Similarly,

$$D = 2\log\left(\frac{\text{likelihood for alternate model}}{\text{likelihood for null hypothesis}}\right)$$

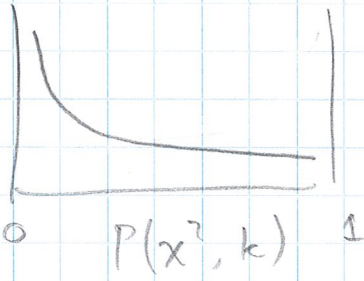
is distributed as a χ^2 distribution with k = difference in the numbers of degrees of freedom in the two hypotheses.

If the model correctly describes the data and all other assumptions are satisfied then $P(\chi^2, k)$ will be uniformly distributed between 0 and 1.

If the model were fitting statistical fluctuations, then χ^2 would be smaller than expected



If the model did not describe the data then χ^2 would be larger than expected



For a given single measurement we can't tell what this distribution would look like

But if we can simulate experiments then we can at least test the fitting algorithm.

Monte Carlo Methods

1. Calculate integrals of non-analytic functions.
2. Simulate random processes

Pseudorandom number generation:

$$X_{n+1} = (a X_n + b) \bmod m$$

a, b, m are large integers. They must be carefully chosen to avoid short periods. But the sequence is deterministic given a seed X_n (which also may need to be chosen carefully).

Other algorithms in Root:

TRandom
TRandom1
TRandom3

Generate pseudorandom numbers on $[0, 1)$.
⊛ Uniformly distributed.

Generating other distributions.

Let $f(x)$ be a p.d.f. and $F(x)$ be its cumulative distribution.

$$F(x) = \int_{-\infty}^{x} f(x)\,dx \quad ; \quad 0 \leq F(x) < 1 .$$

1. Generate u uniformly on $(0, 1)$
2. $x = F^{-1}(u)$.

Example: Exponential distribution

$$f(x) = \frac{1}{\tau} e^{-x/\tau}$$

$$F(x) = \frac{1}{\tau} \int_0^x e^{-t/\tau} dt$$

$$= -e^{-t/\tau} \Big|_0^x$$

$$= 1 - e^{-x/\tau} = u$$

$$e^{-x/\tau} = 1 - u$$

equivalent to $e^{-x/\tau} = u$

$$\log u = -x/\tau$$

$$x = -\tau \log u$$

Acceptance - Rejection method

Let $h(x)$ be an easily generated distribution forexample uniform over (a, b).

Pick C so that $Ch(x) > f(x)$ for all x.

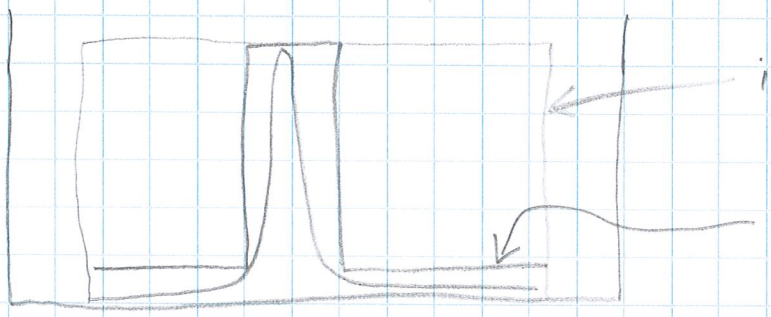
1. Generate x according to $h(x)$
2. Generate u uniformly over $[0, Ch(x)]$
3. If $Ch(x) > f(x)$ then start over, otherwise, accept x.

x will be distributed as $f(x)$.

Acceptance-Rejection can be more efficient than the inverse method.

Adaptive algorithms (eg VEGAS)

Find regions where $f(x)$ is large, divide (a, b) more finely in these regions:



Algorithms generalized to large dimensions. Well suited for matrix elements expressed as functions of momenta of multiple final state particles.

Other algorithms:

Normal distribution:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \qquad f(y) = \frac{1}{\sqrt{2\pi}} e^{-y^2/2}$$

$$f(x,y) = \frac{1}{2\pi} e^{-r^2/2} \qquad r^2 = x^2 + y^2.$$

1. Generate $u_{1,2}$ uniformly over $(0,1)$
2. $r = \sqrt{-2 \log u_2}$
3. $\theta = 2\pi u_1$

$$x = r \cos\theta \qquad y = r \sin\theta$$

Gaussian with mean μ and rms σ:

Let $f(x)$ be normally distributed.

Then $g(x;\mu,\sigma) = \mu + \sigma f(x)$.

Correlated Gaussian distribution:

$$V = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$

Let u_1 and u_2 be normally distributed, then let

$$x_1 = \mu_1 + \sigma_1 u_1$$

$$x_2 = \mu_2 + \rho\sigma_2 u_1 + \sqrt{1-\rho^2}\,\sigma_2 u_2$$

In general $x_i = \mu_i + \sum_j L_{ij} u_j$

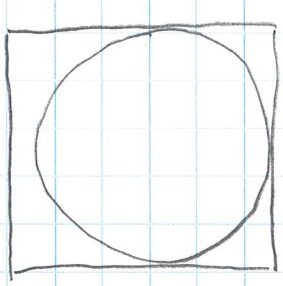
Where L is a lower triangular matrix that satisfies

$$V = LL^T$$

(Cholesky decomposition).

Most of these methods are already implemented in ROOT in the TRandom class.

Monte Carlo integration:



Area of circle:
Generate x, y uniformly on $(-1, 1)$.
If $x^2 + y^2 > 1$, reject the point. Otherwise count it

$$\lim_{n\to\infty} \frac{n}{N} = \pi.$$