Computing and Statistical Data Analysis
Stat 4: MC, Intro to Statistical Tests

London Postgraduate Lectures on Particle Physics;

University of London MSci course PH4515

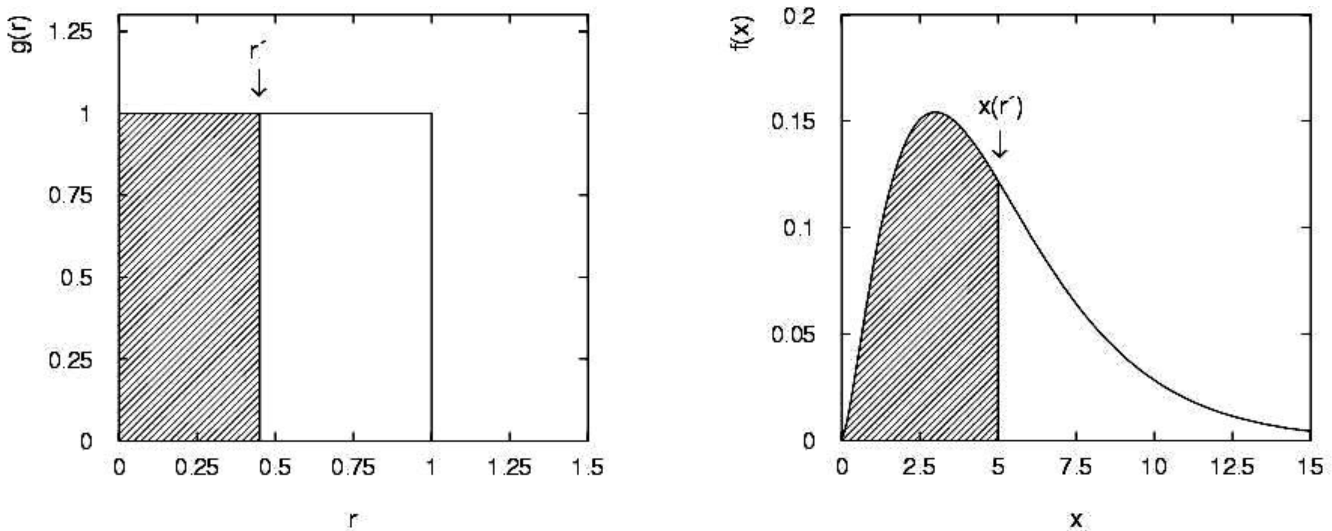
Glen Cowan
Physics Department
Royal Holloway, University of London
g.cowan@rhul.ac.uk
www.pp.rhul.ac.uk/~cowan

Course web page:

www.pp.rhul.ac.uk/~cowan/stat_course.html

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

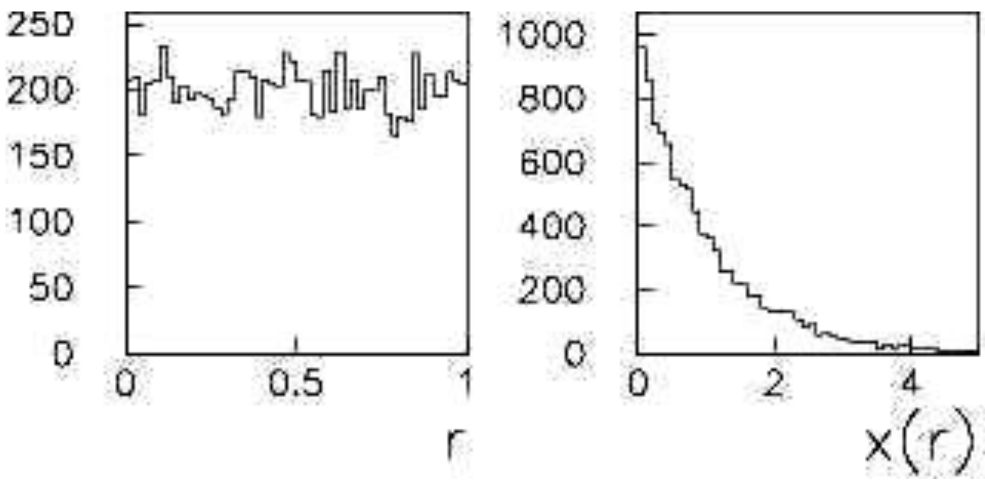
That is, $\boxed{\text{set } F(x) = r \text{ and solve for } x(r).}$

Example of the transformation method

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

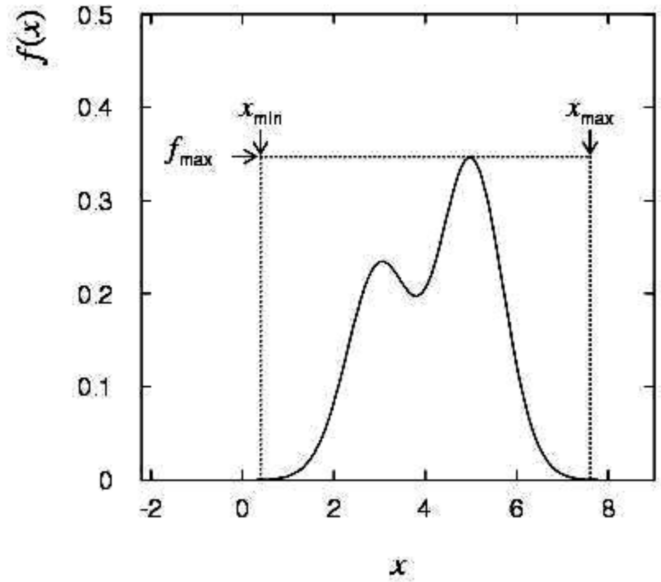
Set $\int_0^x \frac{1}{\xi} e^{-x'/\xi}\, dx' = r$ and solve for $x\,(r)$.

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

$$x = x_{\text{min}} + r_1(x_{\text{max}} - x_{\text{min}})\ ,\quad r_1 \text{ is uniform in } [0,1].$$

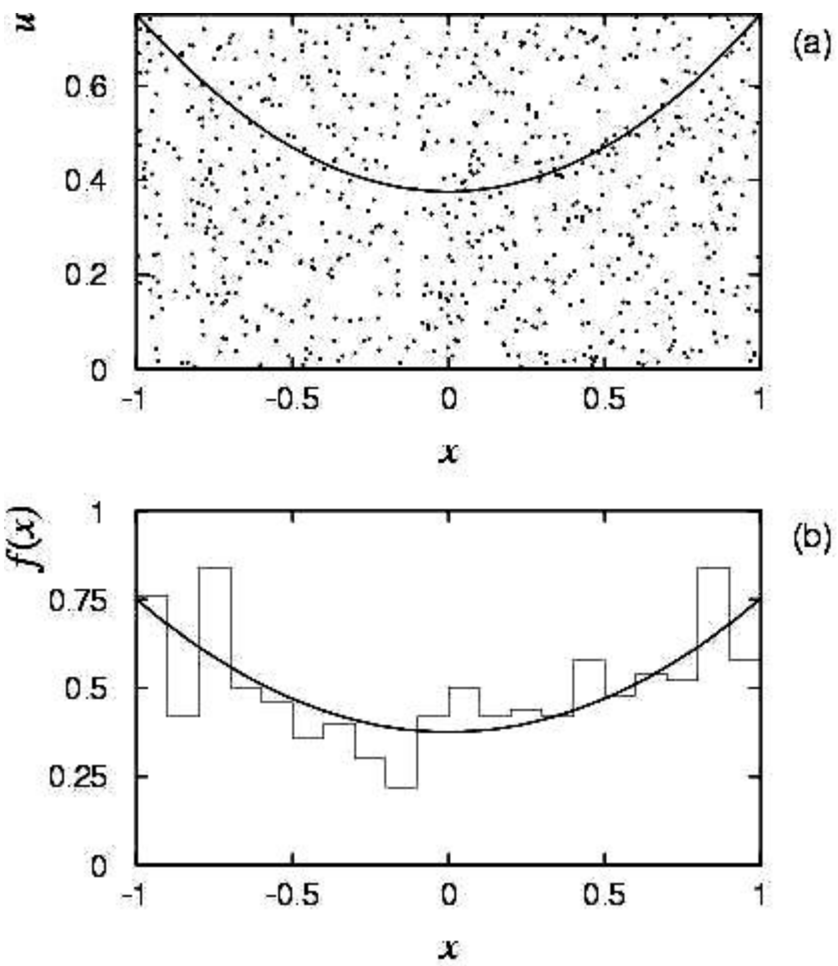
(2) Generate a 2nd independent random number u uniformly distributed between 0 and f_{max}, i.e. $u = r_2 f_{\text{max}}$.

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

Example with acceptance-rejection method

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

$$(-1 \le x \le 1)$$



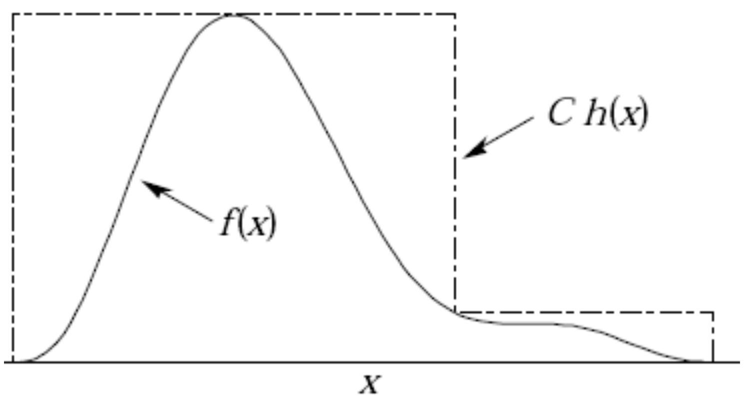
If dot below curve, use x value in histogram.

Improving efficiency of the acceptance-rejection method

The fraction of accepted points is equal to the fraction of the box's area under the curve.

For very peaked distributions, this may be very low and thus the algorithm may be slow.

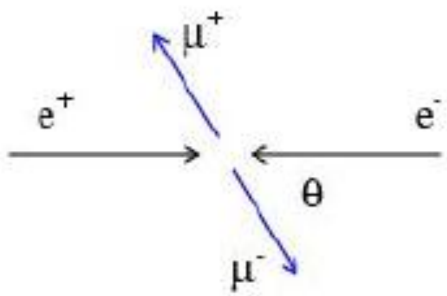
Improve by enclosing the pdf $f(x)$ in a curve $C\,h(x)$ that conforms to $f(x)$ more closely, where $h(x)$ is a pdf from which we can generate random values and C is a constant.



Generate points uniformly over $C\,h(x)$.

If point is below $f(x)$, accept x.

Monte Carlo event generators



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

Generate $\cos\theta$ and ϕ:

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

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

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

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

pp $\rightarrow$ 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.

A simulated event

⋮

Event listing (summary)

I	particle/jet	KS	KF	orig	p_x	p_y	p_z	E	m
1	!p+!	21	2212	0	0.000	0.000	7000.000	7000.000	0.938
2	!p+!	21	2212	0	0.000	0.000	-7000.000	7000.000	0.938
3	!g!	21	21	1	0.863	-0.323	1739.862	1739.862	0.000
4	!ubar!	21	-2	2	-0.621	-0.163	-777.415	777.415	0.000
5	!g!	21	21	3	-2.427	5.486	1487.857	1487.	
6	!g!	21	21	4	-62.910	63.357	-463.274	471.	
7	!~g!	21	1000021	0	314.363	544.843	498.897	979.	
8	!~g!	21	1000021	0	-379.700	-476.000	525.686	980.	
9	!~chi_1-!	21	-1000024	7	130.058	112.247	129.860	263.	
10	!sbar!	21	-3	7	259.400	187.468	83.100	330.	
11	!c!	21	4	7	-79.403	242.409	283.026	381.	
12	!~chi_20!	21	1000023	8	-326.241	-80.971	113.712	385.	
13	!b!	21	5	8	-51.841	-294.077	389.853	491.	
14	!bbar!	21	-5	8	-0.597	-99.577	21.299	101.	
15	!~chi_10!	21	1000022	9	103.352	81.316	83.457	175.	
16	!s!	21	3	9	5.451	38.374	52.302	65.	
17	!cbar!	21	-4	9	20.839	-7.250	-5.938	22.	
18	!~chi_10!	21	1000022	12	-136.266	-72.961	53.246	181.	
19	!nu_mu!	21	14	12	-78.263	-24.757	21.719	84.	
20	!nu_mubar!	21	-14	12	-107.801	16.901	38.226	115.	
21	gamma	1	22	4	2.636	1.357	0.125	2.	
22	(~chi_1-)	11	-1000024	9	129.643	112.440	129.820	262.	
23	(~chi_20)	11	1000023	12	-322.330	-80.817	113.191	382.	
24	~chi_10	1	1000022	15	97.944	77.819	80.917	169.	
25	~chi_10	1	1000022	18	-136.266	-72.961	53.246	181.	
26	nu_mu	1	14	19	-78.263	-24.757	21.719	84.	
27	nu_mubar	1	-14	20	-107.801	16.901	38.226	115.	
28	(Delta++)	11	2224	2	0.222	0.012	-2734.287	2734.	

⋮

397	pi+	1	211	209	0.006	0.398	-308.296	308.297	0.140
398	gamma	1	22	211	0.407	0.087	-1695.458	1695.458	0.000
399	gamma	1	22	211	0.113	-0.029	-314.822	314.822	0.000
400	(pi0)	11	111	212	0.021	0.122	-103.709	103.709	0.135
401	(pi0)	11	111	212	0.084	-0.068	-94.276	94.276	0.135
402	(pi0)	11	111	212	0.267	-0.052	-144.673	144.674	0.135
403	gamma	1	22	215	-1.581	2.473	3.306	4.421	0.000
404	gamma	1	22	215	-1.494	2.143	3.051	4.016	0.000
405	pi-	1	-211	216	0.007	0.738	4.015	4.085	0.140
406	pi+	1	211	216	-0.024	0.293	0.486	0.585	0.140
407	K+	1	321	218	4.382	-1.412	-1.799	4.968	0.494
408	pi-	1	-211	218	1.183	-0.894	-0.176	1.500	0.140
409	(pi0)	11	111	218	0.955	-0.459	-0.590	1.221	0.135
410	(pi0)	11	111	218	2.349	-1.105	-1.181	2.855	0.135
411	(Kbar0)	11	-311	219	1.441	-0.247	-0.472	1.615	0.498
412	pi-	1	-211	219	2.232	-0.400	-0.249	2.285	0.140
413	K+	1	321	220	1.380	-0.652	-0.361	1.644	0.494
414	(pi0)	11	111	220	1.078	-0.265	0.175	1.132	0.135
415	(K_S0)	11	310	222	1.841	0.111	0.894	2.109	0.498
416	K+	1	321	223	0.307	0.107	0.252	0.642	0.494
417	pi-	1	-211	223	0.266	0.316	-0.201	0.480	0.140
418	nbar0	1	-2112	226	1.335	1.641	2.078	3.111	0.940
419	(pi0)	11	111	226	0.899	1.046	1.311	1.908	0.135
420	pi+	1	211	227	0.217	1.407	1.356	1.971	0.140
421	(pi0)	11	111	227	1.207	2.336	2.767	3.820	0.135
422	n0	1	2112	228	3.475	5.324	5.702	8.592	0.940
423	pi-	1	-211	228	1.856	2.606	2.808	4.259	0.140
424	gamma	1	22	229	-0.012	0.247	0.421	0.489	0.000
425	gamma	1	22	229	0.025	0.034	0.009	0.043	0.000
426	pi+	1	211	230	2.718	5.229	6.403	8.703	0.140
427	(pi0)	11	111	230	4.109	6.747	7.597	10.961	0.135
428	pi-	1	-211	231	0.551	1.233	1.945	2.372	0.140
429	(pi0)	11	111	231	0.645	1.141	0.922	1.608	0.135
430	gamma	1	22	232	-0.383	1.169	1.208	1.724	0.000
431	gamma	1	22	232	-0.201	0.070	0.060	0.221	0.000

PYTHIA Monte Carlo

pp → gluino-gluino

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`**

Hypotheses

A hypothesis H specifies the probability for the data, i.e., the outcome of the observation, here symbolically: x.

x could be uni-/multivariate, continuous or discrete.

E.g. write $x \sim f(x|H)$.

x could represent e.g. observation of a single particle, a single event, or an entire "experiment".

Possible values of x form the sample space S (or "data space").

Simple (or "point") hypothesis: $f(x|H)$ completely specified.

Composite hypothesis: H contains unspecified parameter(s).

The probability for x given H is also called the likelihood of the hypothesis, written $L(x|H)$.

Definition of a test

Goal is to make some statement based on the observed data x as to the validity of the possible hypotheses.

Consider e.g. a simple hypothesis H_0 and alternative H_1.

A test of H_0 is defined by specifying a critical region W of the data space such that there is no more than some (small) probability α, assuming H_0 is correct, to observe the data there, i.e.,

$$P(x \in W \mid H_0) \leq \alpha$$

If x is observed in the critical region, reject H_0.

α is called the size or significance level of the test.

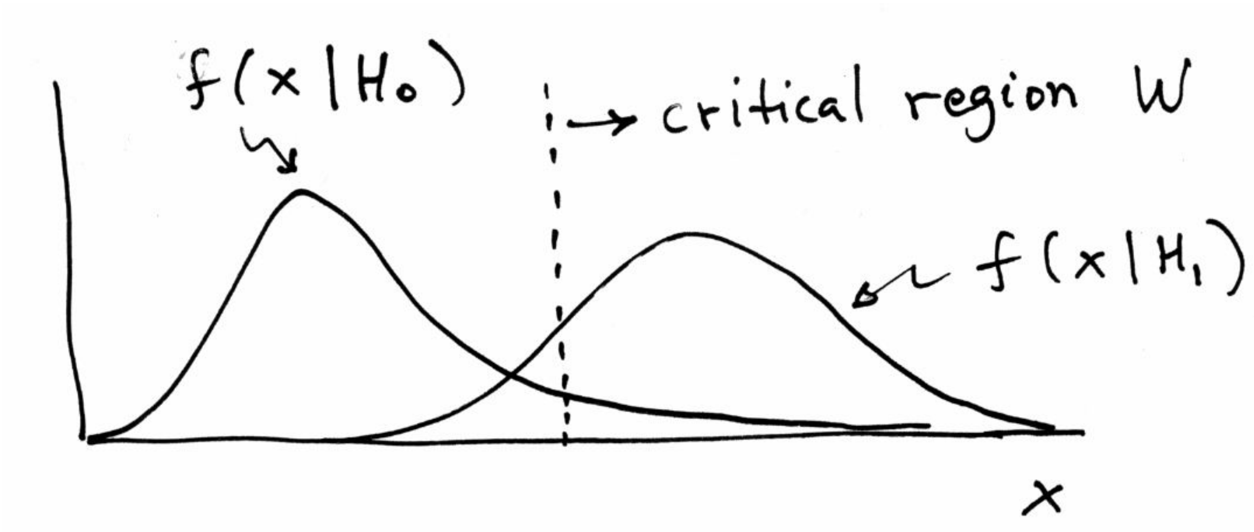
Critical region also called "rejection" region; complement is acceptance region.

Definition of a test (2)

But in general there are an infinite number of possible critical regions that give the same significance level α.

So the choice of the critical region for a test of H_0 needs to take into account the alternative hypothesis H_1.

Roughly speaking, place the critical region where there is a low probability to be found if H_0 is true, but high if H_1 is true:



Rejecting a hypothesis

Note that rejecting H_0 is not necessarily equivalent to the statement that we believe it is false and H_1 true. In frequentist statistics only associate probability with outcomes of repeatable observations (the data).

In Bayesian statistics, probability of the hypothesis (degree of belief) would be found using Bayes' theorem:

$$P(H|x) = \frac{P(x|H)\pi(H)}{\int P(x|H)\pi(H)\,dH}$$

which depends on the prior probability $\pi(H)$.

What makes a frequentist test useful is that we can compute the probability to accept/reject a hypothesis assuming that it is true, or assuming some alternative is true.

Type-I, Type-II errors

Rejecting the hypothesis H_0 when it is true is a Type-I error.

The maximum probability for this is the size of the test:

$$P(x \in W \mid H_0) \le \alpha$$

But we might also accept H_0 when it is false, and an alternative H_1 is true.

This is called a Type-II error, and occurs with probability

$$P(x \in S - W \mid H_1) = \beta$$

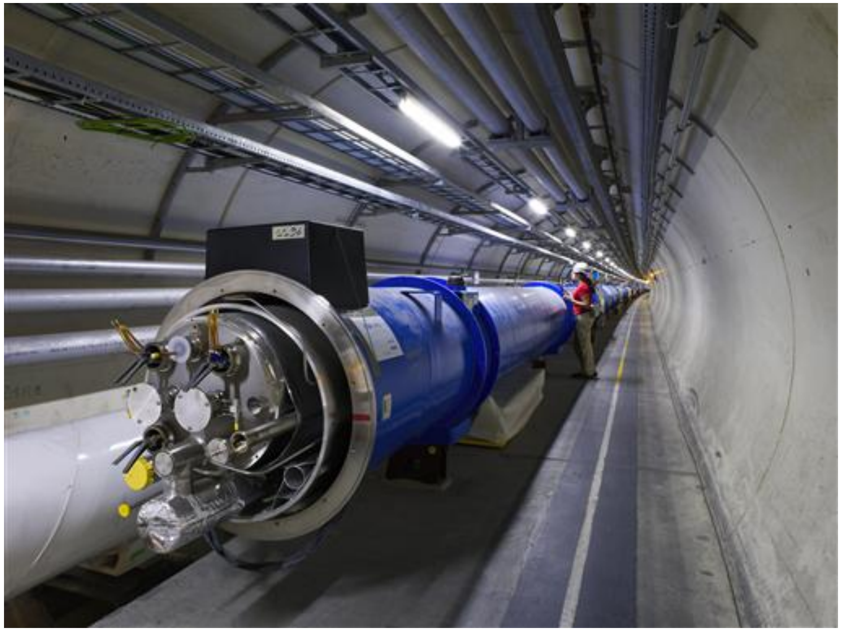
One minus this is called the power of the test with respect to the alternative H_1:

$$\text{Power} = 1 - \beta$$

Example setting for statistical tests: the Large Hadron Collider

Counter-rotating proton beams
in 27 km circumference ring

pp centre-of-mass energy 14 TeV

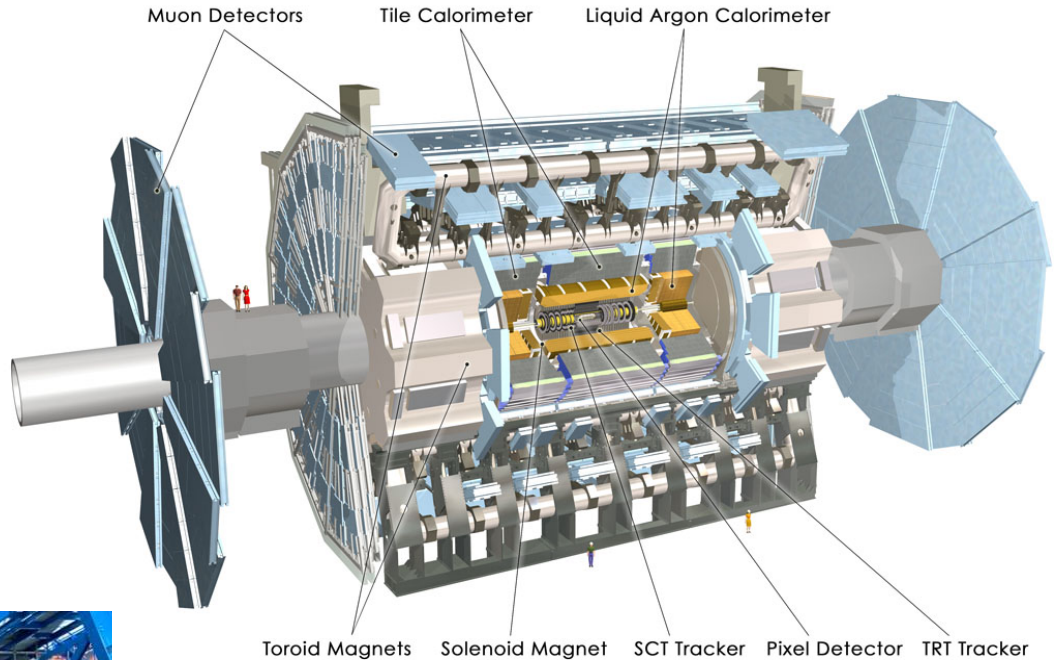


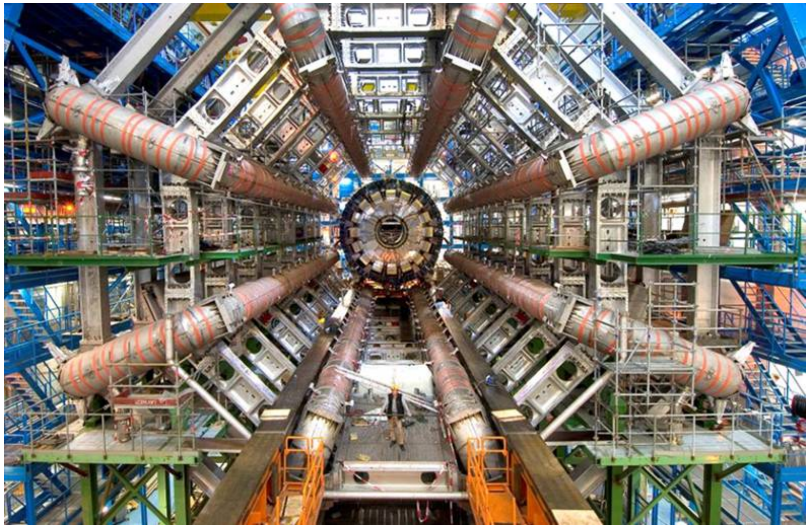
Detectors at 4 pp collision points:

ATLAS
CMS general purpose
LHCb (b physics)
ALICE (heavy ion physics)

The ATLAS detector

2100 physicists
37 countries
167 universities/labs





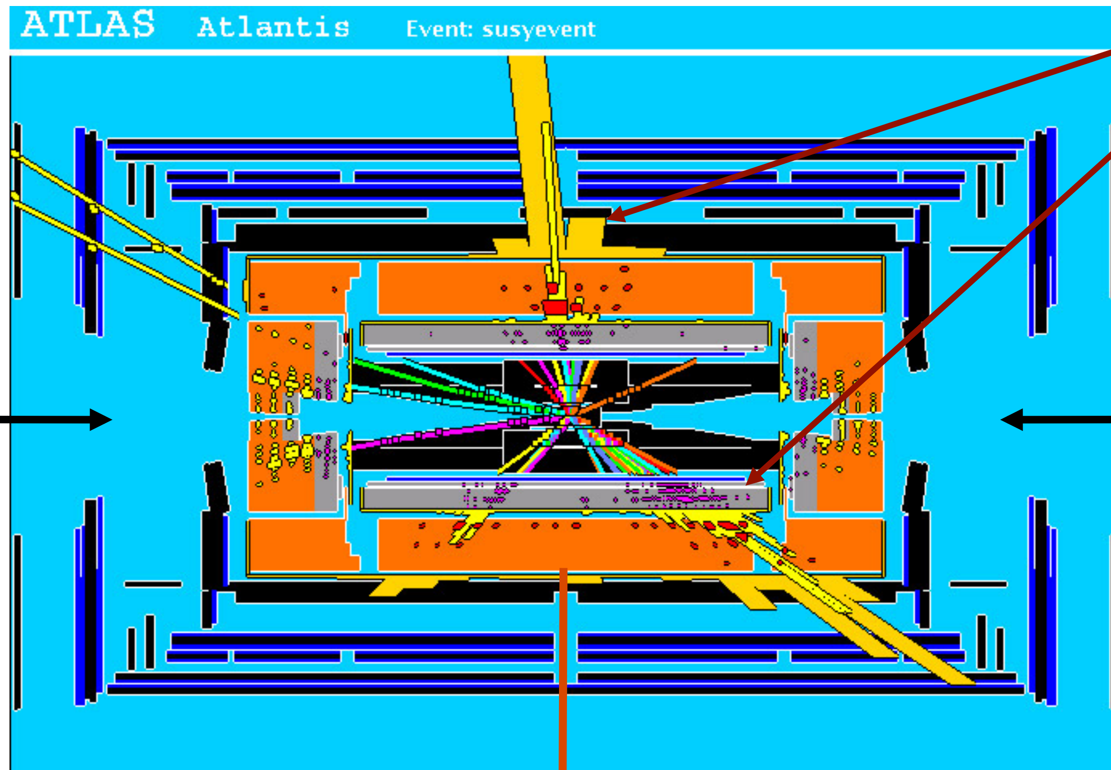
25 m diameter
46 m length
7000 tonnes
$\sim 10^8$ electronic channels

A simulated SUSY event

high p_T muons

high p_T jets of hadrons

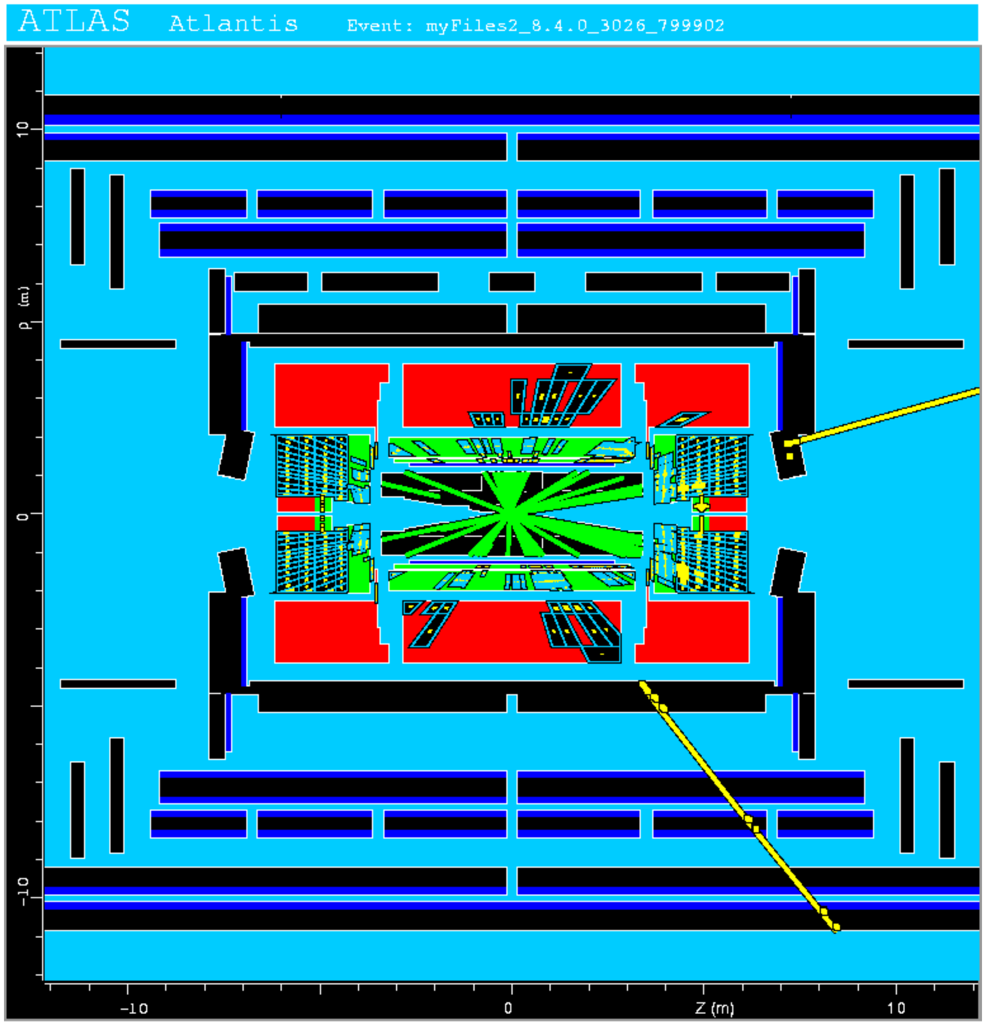
p



p

missing transverse energy

Background events



This event from Standard Model ttbar production also has high p_T jets and muons, and some missing transverse energy.

→ can easily mimic a SUSY event.

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

$$\mathrm{pp} \to t\bar{t} \;, \quad \mathrm{pp} \to \tilde{g}\tilde{g} \;, \ldots$$

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.

E.g. call H_0 the background hypothesis (the event type we want to reject); H_1 is signal hypothesis (the type we want).

Selecting events

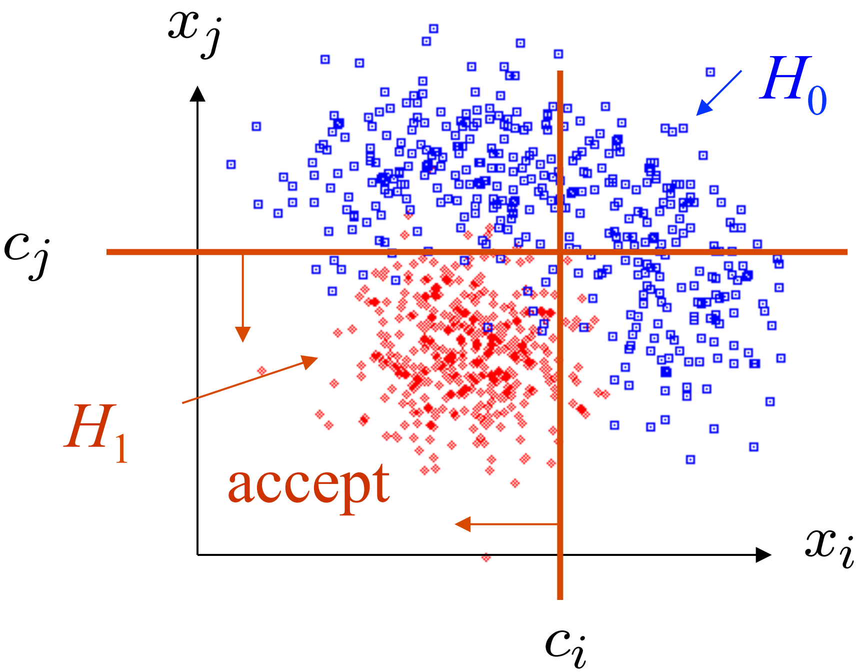
Suppose we have a data sample with two kinds of events, corresponding to hypotheses H_0 and H_1 and we want to select those of type H_1.

Each event is a point in $\vec{x}$ space. What 'decision boundary' should we use to accept/reject events as belonging to event types H_0 or H_1?

Perhaps select events with 'cuts':

$$x_i < c_i$$

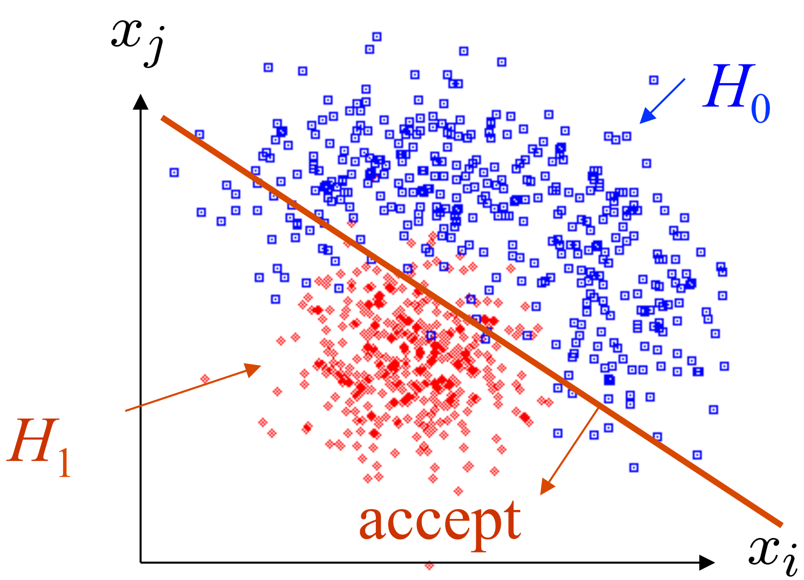
$$x_j < c_j$$



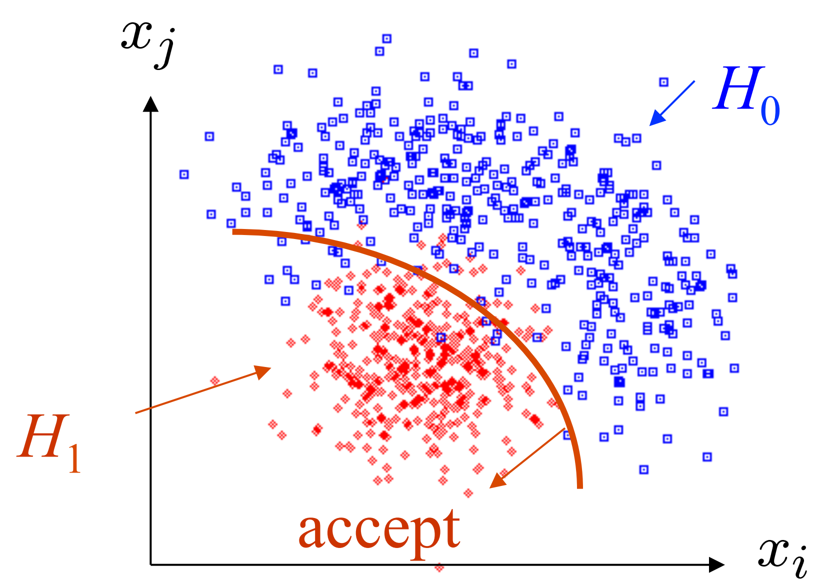
Other ways to select events

Or maybe use some other sort of decision boundary:

linear



or nonlinear



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

Test statistics

The decision boundary can be defined by an equation of the form

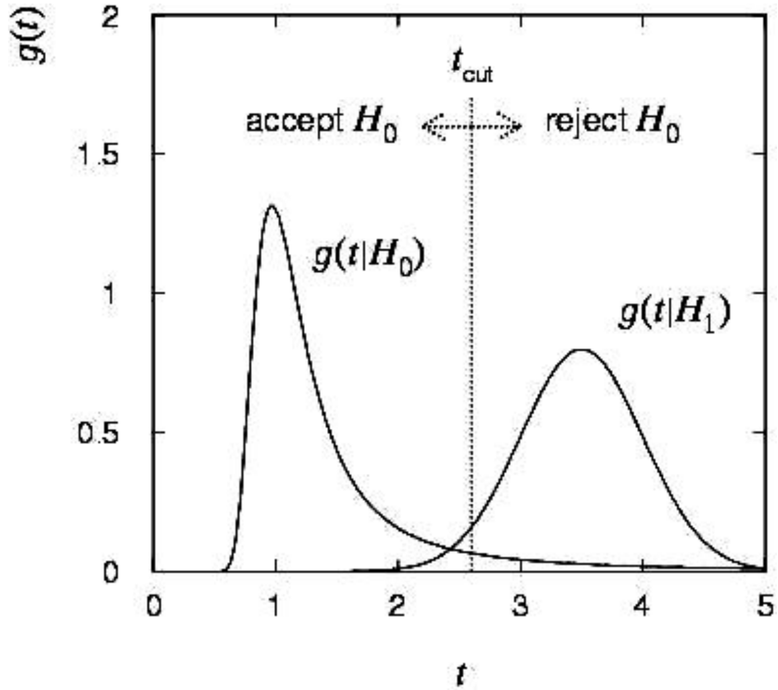
$$t(x_1, \ldots, x_n) = t_{\mathrm{cut}}$$

where $t(x_1,\ldots, x_n)$ is a scalar test statistic.

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

Decision boundary is now a single 'cut' on t, which divides the space into the critical (rejection) region and acceptance region.

This defines a test. If the data fall in the critical region, we reject H_0.



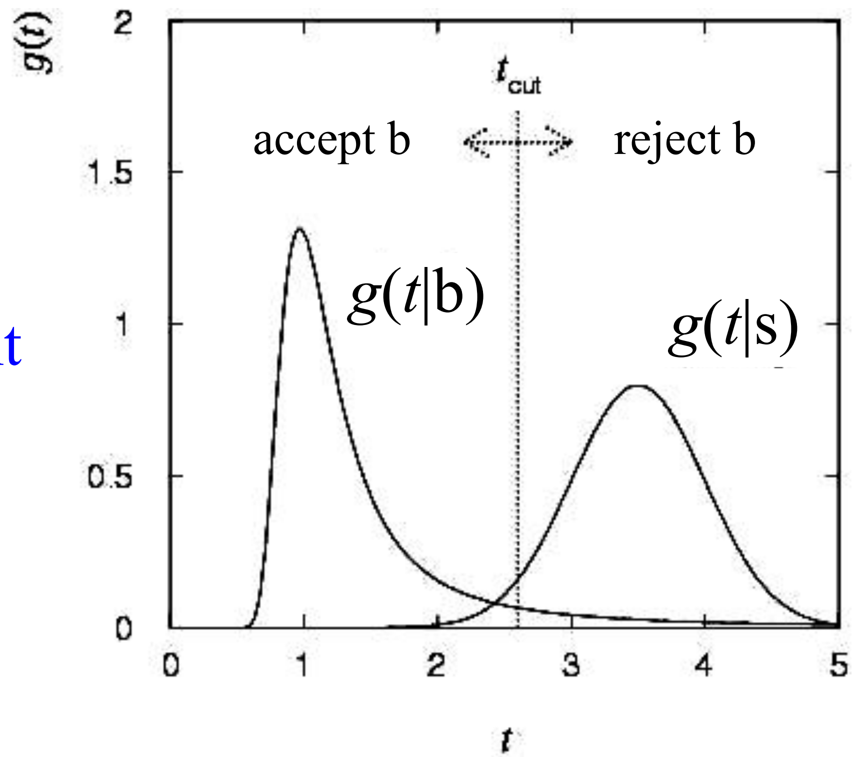
Signal/background efficiency

Probability to reject background hypothesis for background event (background efficiency):

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

Probability to accept a signal event as signal (signal efficiency):

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



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 signal 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:

$$
\begin{aligned}
P(\mathrm{s}|t > t_{\mathrm{cut}}) &= \frac{P(t > t_{\mathrm{cut}}|\mathrm{s})\pi_{\mathrm{s}}}{P(t > t_{\mathrm{cut}}|\mathrm{s})\pi_{\mathrm{s}} + P(t > t_{\mathrm{cut}}|\mathrm{b})\pi_{\mathrm{b}}} \\
&= \frac{\varepsilon_{\mathrm{s}}\pi_{\mathrm{s}}}{\varepsilon_{\mathrm{s}}\pi_{\mathrm{s}} + \varepsilon_{\mathrm{b}}\pi_{\mathrm{b}}}
\end{aligned}
$$

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

Constructing a test statistic

How can we choose a test's critical region in an 'optimal way'?

Neyman-Pearson lemma states:

To get the highest power for a given significance level in a test of H_0, (background) versus H_1, (signal) the critical region should have

$$\frac{P(\mathbf{x}|H_1)}{P(\mathbf{x}|H_0)} > c$$

inside the region, and $\leq c$ outside, where c is a constant which determines the power.

Equivalently, optimal scalar test statistic is

$$\boxed{t(\mathbf{x}) = \frac{P(\mathbf{x}|H_1)}{P(\mathbf{x}|H_0)}}$$

N.B. any monotonic function of this is leads to the same test.

Why Neyman-Pearson doesn't always help

The problem is that we usually don't have explicit formulae for the pdfs $P(\mathbf{x}|H_0)$, $P(\mathbf{x}|H_1)$.

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, → 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.

Multivariate methods

Many new (and some old) methods:

- Fisher discriminant
- Neural networks
- Kernel density methods
- Support Vector Machines
- Decision trees
  - Boosting
  - Bagging

New software for HEP, e.g.,

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

`StatPatternRecognition`, I. Narsky, physics/0507143