Computing and Statistical Data Analysis
Stat 6: MVA (cont.) / Parameter Estimation

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

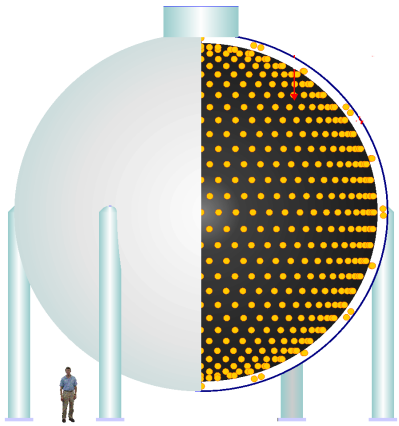
Course web page:

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

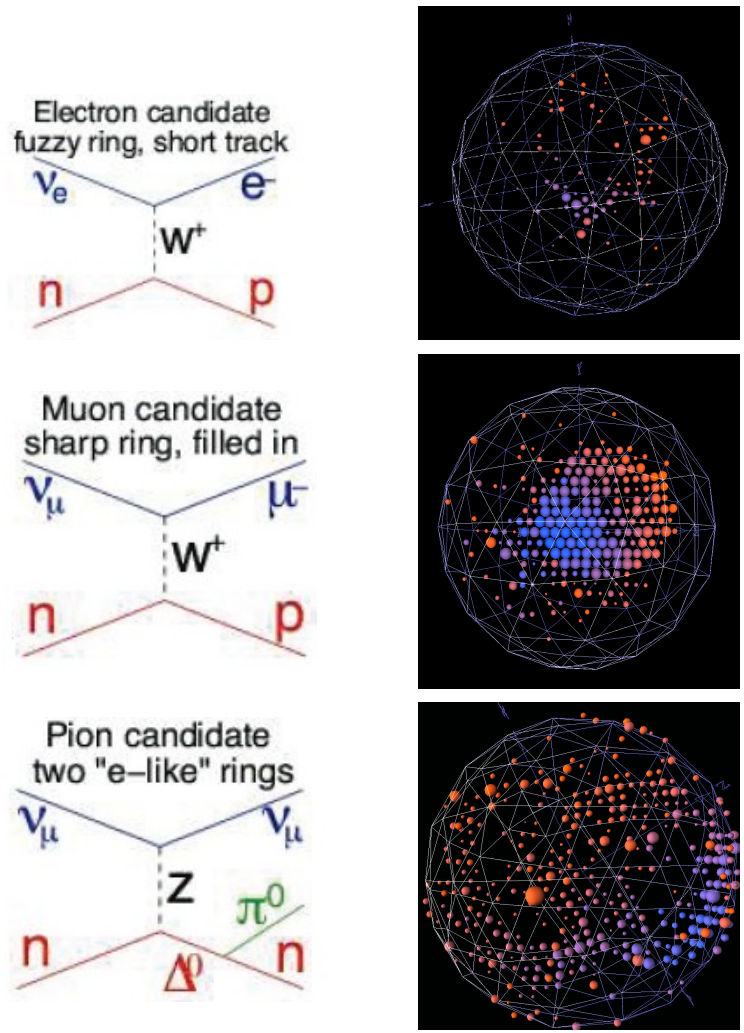
Particle i.d. in MiniBooNE

Detector is a 12-m diameter tank of mineral oil exposed to a beam of neutrinos and viewed by 1520 photomultiplier tubes:

MiniBooNE Detector



Search for ν_μ to ν_e oscillations required particle i.d. using information from the PMTs.



H.J. Yang, MiniBooNE PID, DNP06

Decision trees

Out of all the input variables, find the one for which with a single cut gives best improvement in signal purity:

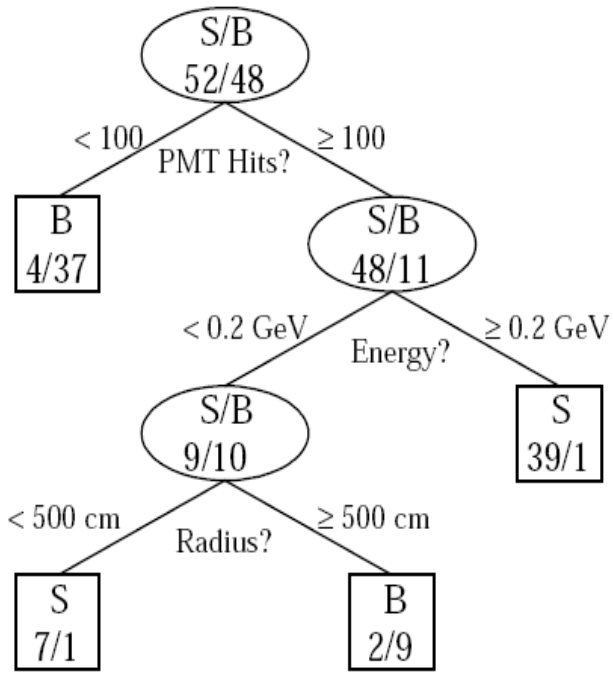
$$P = \frac{\sum_{\text{signal}} w_i}{\sum_{\text{signal}} w_i + \sum_{\text{background}} w_i}$$

where w_i. is the weight of the ith event.

Resulting nodes classified as either signal/background.

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

The set of cuts defines the decision boundary.



Example by MiniBooNE experiment, B. Roe et al., NIM 543 (2005) 577

Finding the best single cut

The level of separation within a node can, e.g., be quantified by the *Gini coefficient*, calculated from the (s or b) purity as:

$$G = p(1-p)$$

For a cut that splits a set of events a into subsets b and c, one can quantify the improvement in separation by the change in weighted Gini coefficients:

$$\Delta = W_a G_a - W_b G_b - W_c G_c \quad \text{where, e.g.,} \quad W_a = \sum_{i \in a} w_i$$

Choose e.g. the cut to the maximize Δ; a variant of this scheme can use instead of Gini e.g. the misclassification rate:

$$\varepsilon = 1 - \max(p, 1-p)$$

Decision trees (2)

The terminal nodes (leaves) are classified a signal or background depending on majority vote (or e.g. signal fraction greater than a specified threshold).

This classifies every point in input-variable space as either signal or background, a decision tree classifier, with discriminant function

$$f(\boldsymbol{x}) = 1 \text{ if } \boldsymbol{x} \text{ in signal region, } -1 \text{ otherwise}$$

Decision trees tend to be very sensitive to statistical fluctuations in the training sample.

Methods such as boosting can be used to stabilize the tree.

Boosting

Boosting is a general method of creating a set of classifiers which can be combined to achieve a new classifier that is more stable and has a smaller error than any individual one.

Often applied to decision trees but, can be applied to any classifier.

Suppose we have a training sample T consisting of N events with

$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ event data vectors (each $\boldsymbol{x}$ multivariate)

$y_1, \ldots, y_N$ true class labels, +1 for signal, −1 for background

$w_1, \ldots, w_N$ event weights

Now define a rule to create from this an ensemble of training samples $T_1, T_2, \ldots$, derive a classifier from each and average them.

Trick is to create modifications in the training sample that give classifiers with smaller error rates than those of the preceding ones.

A successful example is AdaBoost (Freund and Schapire, 1997).

AdaBoost

First initialize the training sample T_1 using the original

$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ event data vectors

$y_1, \ldots, y_N$ true class labels (+1 or -1)

$w_1^{(1)}, \ldots, w_N^{(1)}$ event weights

with the weights equal and normalized such that

$$\sum_{i=1}^{N} w_i^{(1)} = 1$$

Then train the classifier $f_1(\boldsymbol{x})$ (e.g. a decision tree) with a method that incorporates the event weights. For an event with data $\boldsymbol{x}_i$,

$f_1(\boldsymbol{x}_i) > 0$ classify as signal

$f_1(\boldsymbol{x}_i) < 0$ classify as background

Updating the event weights

Define the training sample for step k+1 from that of k by updating the event weights according to

$$w_i^{(k+1)} = w_i^{(k)} \frac{e^{-\alpha_k f_k(\boldsymbol{x}_i) y_i / 2}}{Z_k}$$

i = event index

k = training sample index

where Z_k is a normalization factor defined such that the sum of the weights over all events is equal to one.

Therefore event weight for event i is increased in the k+1 training sample if it was classified incorrectly in sample k.

Idea is that next time around the classifier should pay more attention to this event and try to get it right.

Error rate of the *k*th classifier

At each step the classifiers $f_k(\boldsymbol{x})$ are defined so as to minimize the error rate ε_k,

$$\varepsilon_k = \sum_{i=1}^{N} w_i^{(k)} I\left(y_i f_k(\boldsymbol{x}_i) \leqslant 0\right)$$

where $I(X) = 1$ if X is true and is zero otherwise.

Assigning the classifier score

Assign a score to the *k*th classifier based on its error rate,

$$\alpha_k = \ln \frac{1-\varepsilon_k}{\varepsilon_k}$$

If we define the final classifier as $f(\boldsymbol{x}) = \sum_{k=1}^{K} \alpha_k f_k(\boldsymbol{x}, T_k)$

then one can show that its error rate on the training data satisfies the bound

$$\varepsilon \leqslant \prod_{k=1}^{K} 2\sqrt{\varepsilon_k(1-\varepsilon_k)}$$

AdaBoost error rate

So providing each classifier in the ensemble has $\varepsilon_k < ½$, i.e., better than random guessing, then the error rate for the final classifier on the training data (not on unseen data) drops to zero.

That is, for sufficiently large K the training data will be over fitted.

The error rate on a validation sample would reach some minimum after a certain number of steps and then could rise.

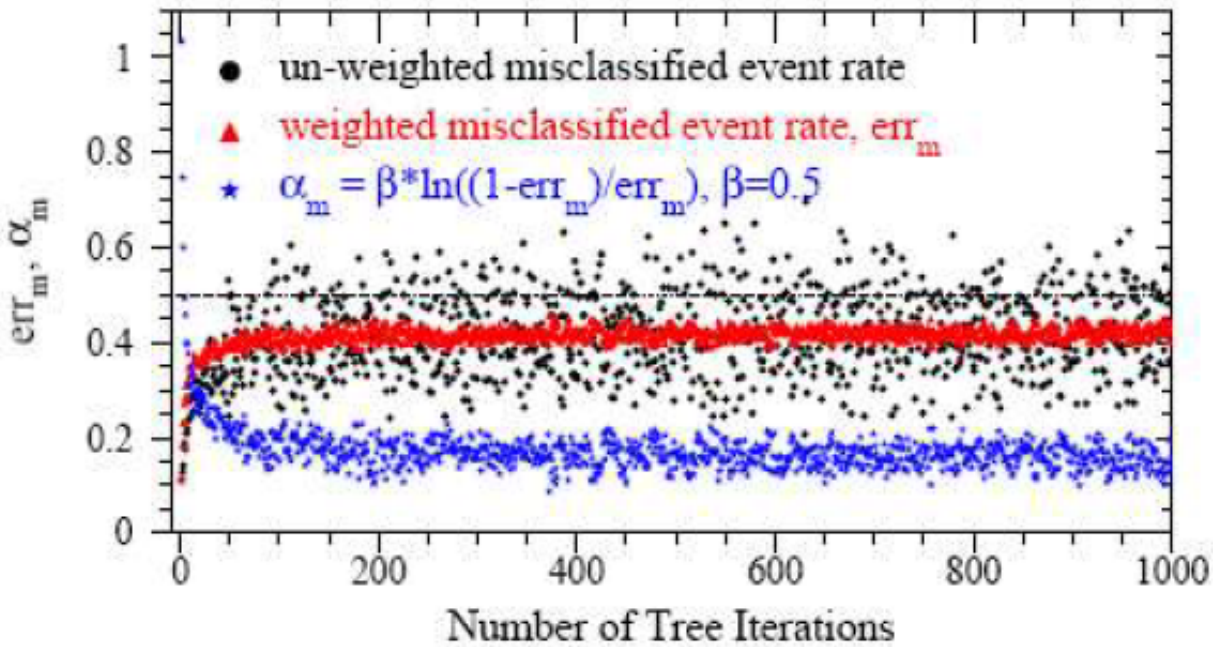
So the procedure is to monitor the error rate of the combined classifier at each step with a validation sample and to stop before it starts to rise.

Although in principle AdaBoost must overfit, in practice following this procedure overtraining is not a big problem.

BDT example from MiniBooNE

~200 input variables for each event (ν interaction producing e, μ or π).

Each individual tree is relatively weak, with a misclassification error rate ~ 0.4 – 0.45

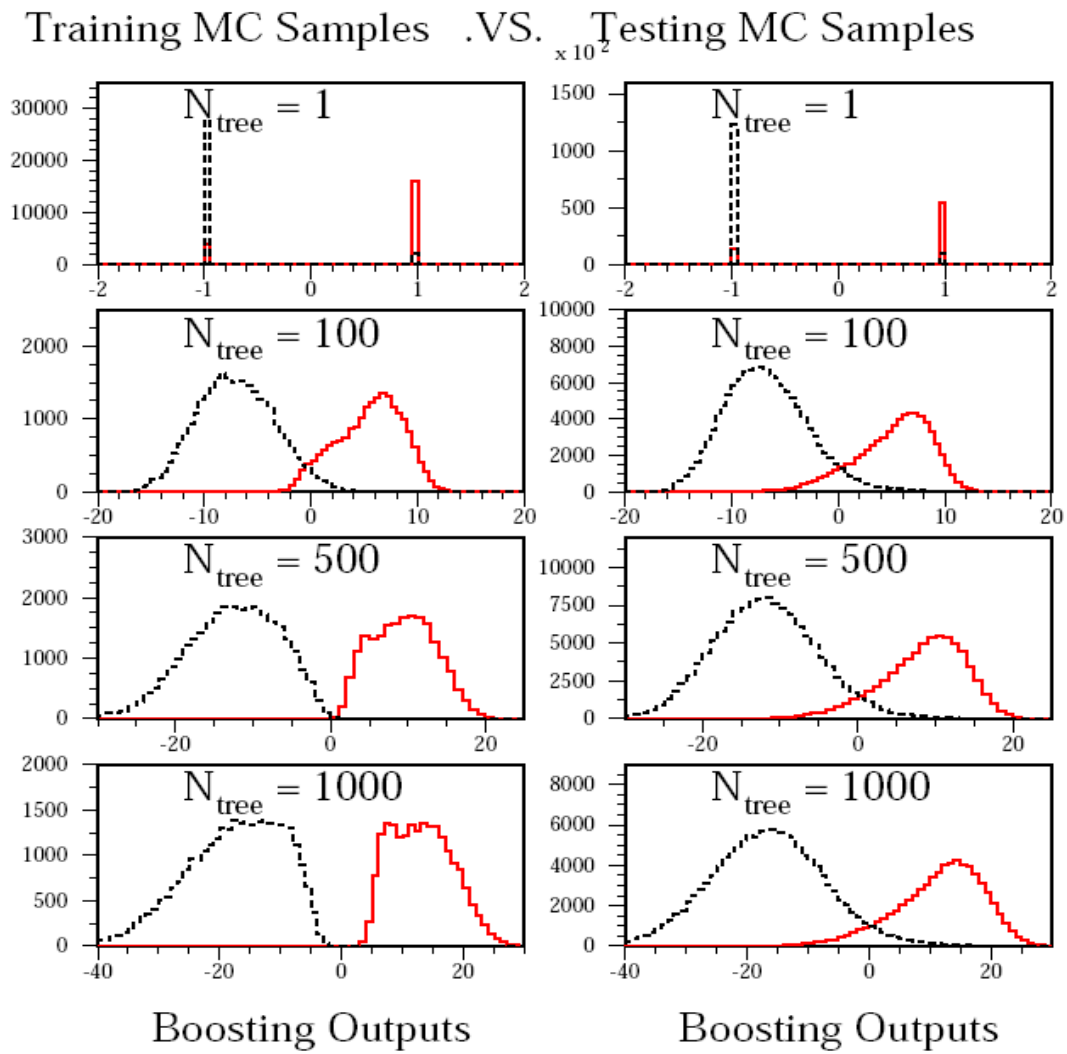


B. Roe et al., NIM 543 (2005) 577

Monitoring overtraining

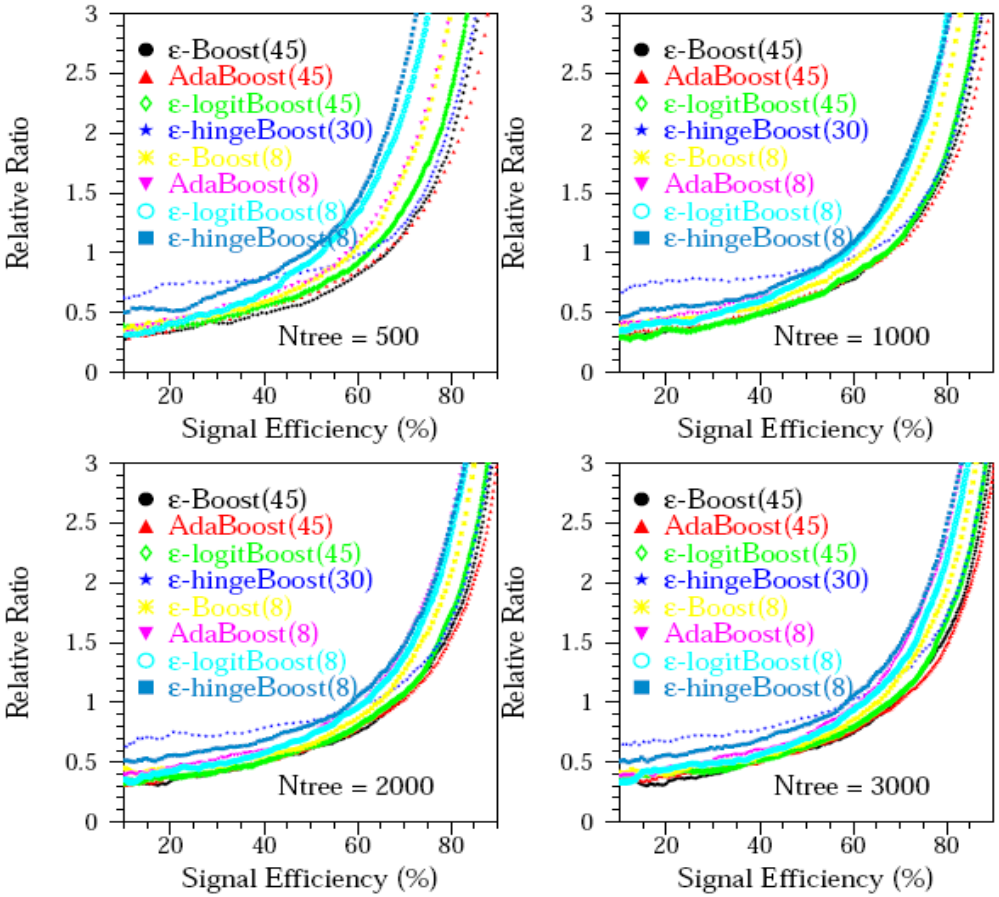
From MiniBooNE example:

Performance stable after a few hundred trees.



Comparison of boosting algorithms

A number of boosting algorithms on the market; differ in the update rule for the weights.



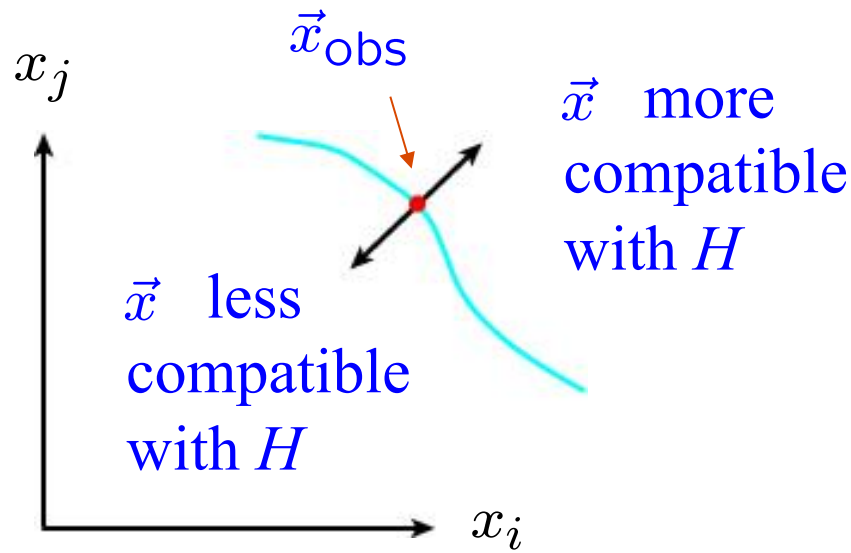
Testing significance / goodness-of-fit

Suppose hypothesis H predicts pdf $f(\vec{x}|H)$ for a set of observations $\vec{x} = (x_1, \ldots, x_n)$.

We observe a single point in this space: $\vec{x}_{\text{obs}}$

What can we say about the validity of H in light of the data?

Decide what part of the data space represents less compatibility with H than does the point $\vec{x}_{\text{obs}}$. (Not unique!)



p-values

Express 'goodness-of-fit' by giving the p-value for H:

p = probability, under assumption of H, to observe data with equal or lesser compatibility with H relative to the data we got.

This is not the probability that H is true!

In frequentist statistics we don't talk about $P(H)$ (unless H represents a repeatable observation). In Bayesian statistics we do; use Bayes' theorem to obtain

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

where $\pi(H)$ is the prior probability for H.

For now stick with the frequentist approach;
result is p-value, regrettably easy to misinterpret as $P(H)$.

p-value example: testing whether a coin is 'fair'

Probability to observe n heads in N coin tosses is binomial:

$$P(n; p, N) = \frac{N!}{n!(N-n)!} p^n (1-p)^{N-n}$$

Hypothesis H: the coin is fair ($p = 0.5$).

Suppose we toss the coin $N = 20$ times and get $n = 17$ heads.

Region of data space with equal or lesser compatibility with H relative to $n = 17$ is: $n = 17, 18, 19, 20, 0, 1, 2, 3$. Adding up the probabilities for these values gives:

$$P(n = 0, 1, 2, 3, 17, 18, 19, \text{ or } 20) = 0.0026 \ .$$

i.e. $p = 0.0026$ is the probability of obtaining such a bizarre result (or more so) 'by chance', under the assumption of H.

The significance of an observed signal

Suppose we observe n events; these can consist of:

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

If n_s, n_b are Poisson r.v.s with means s, b, then $n = n_s + n_b$ is also Poisson, mean = $s + b$:

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

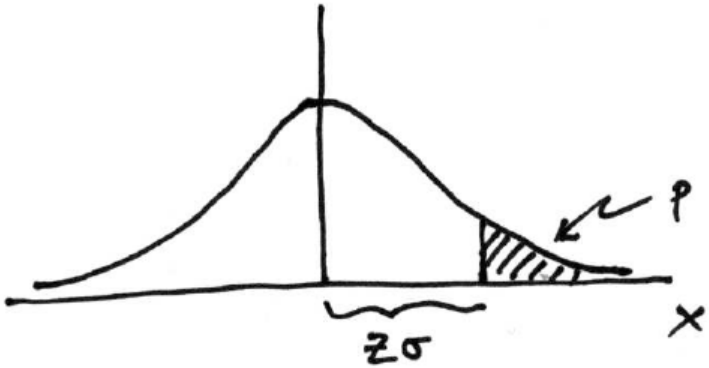
Suppose $b = 0.5$, and we observe $n_{obs} = 5$. Should we claim evidence for a new discovery?

Give p-value for hypothesis $s = 0$:

$$\begin{aligned} p\text{-value} &= P(n \geq 5; b = 0.5, s = 0) \\ &= 1.7 \times 10^{-4} \neq P(s = 0)! \end{aligned}$$

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.



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

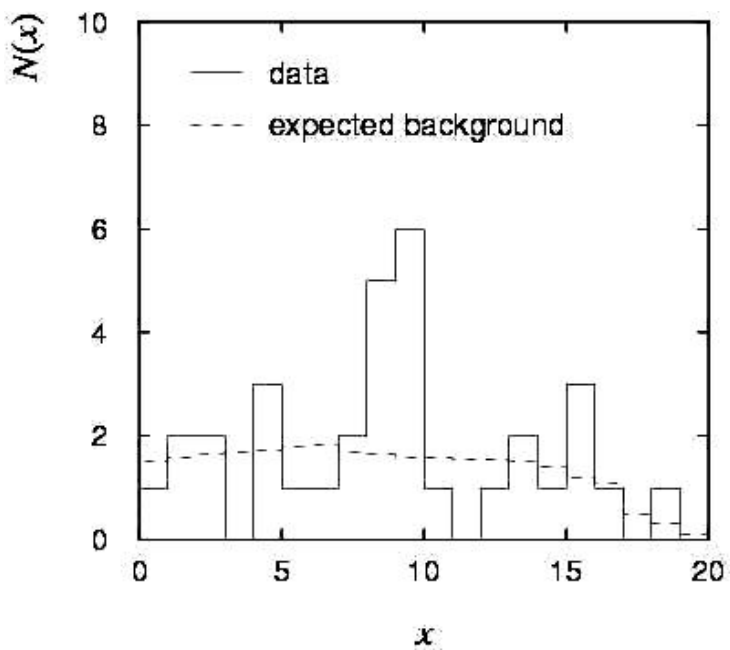
`1 - TMath::Freq`

$$Z = \Phi^{-1}(1-p)$$

`TMath::NormQuantile`

The significance of a peak

Suppose we measure a value x for each event and find:



Each bin (observed) is a Poisson r.v., means are given by dashed lines.

In the two bins with the peak, 11 entries found with $b = 3.2$. The p-value for the $s = 0$ hypothesis is:

$$P(n \geq 11; b = 3.2, s = 0) = 5.0 \times 10^{-4}$$

The significance of a peak (2)

But... did we know where to look for the peak?

→ give $P(n \geq 11)$ in any 2 adjacent bins

Is the observed width consistent with the expected x resolution?

→ take x window several times the expected resolution

How many bins × distributions have we looked at?

→ look at a thousand of them, you'll find a 10^{-3} effect

Did we adjust the cuts to 'enhance' the peak?

→ freeze cuts, repeat analysis with new data

How about the bins to the sides of the peak... (too low!)

Should we publish????

When to publish

HEP folklore is to claim discovery when $p = 2.9 \times 10^{-7}$, corresponding to a significance $Z = 5$.

This is very subjective and really should depend on the prior probability of the phenomenon in question, e.g.,

phenomenon	reasonable p-value for discovery
D^0D^0 mixing	~0.05
Higgs	$\sim 10^{-7}$ (?)
Life on Mars	$\sim 10^{-10}$
Astrology	$\sim 10^{-20}$

One should also consider the degree to which the data are compatible with the new phenomenon, not only the level of disagreement with the null hypothesis; p-value is only first step!

Distribution of the p-value

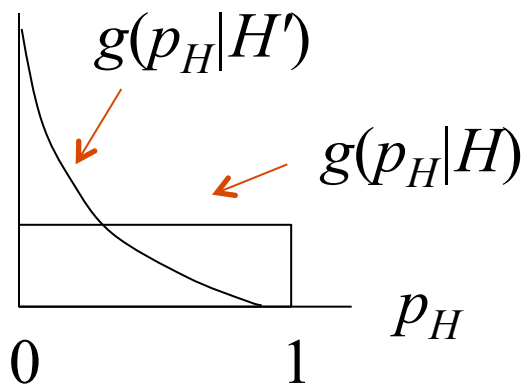
The p-value is a function of the data, and is thus itself a random variable with a given distribution. Suppose the p-value of H is found from a test statistic $t(\boldsymbol{x})$ as

$$p_H = \int_t^\infty f(t'|H)dt'$$

The pdf of p_H under assumption of H is

$$g(p_H|H) = \frac{f(t|H)}{|\partial p_H/\partial t|} = \frac{f(t|H)}{f(t|H)} = 1 \quad (0 \le p_H \le 1)$$

In general for continuous data, under assumption of H, $p_H \sim$ Uniform[0,1] and is concentrated toward zero for Some (broad) class of alternatives.



Using a p-value to define test of H_0

So the probability to find the p-value of H_0, p_0, less than α is

$$P(p_0 \leq \alpha | H_0) = \alpha$$

We started by defining critical region in the original data space ($\boldsymbol{x}$), then reformulated this in terms of a scalar test statistic $t(\boldsymbol{x})$.

We can take this one step further and define the critical region of a test of H_0 with size α as the set of data space where $p_0 \leq \alpha$.

Formally the p-value relates only to H_0, but the resulting test will have a given power with respect to a given alternative H_1.

Pearson's χ^2 statistic

Test statistic for comparing observed data $\vec{n} = (n_1, \ldots, n_N)$ (n_i independent) to predicted mean values $\vec{\nu} = (\nu_1, \ldots, \nu_N)$:

$$\chi^2 = \sum_{i=1}^{N} \frac{(n_i - \nu_i)^2}{\sigma_i^2} \,, \quad \text{where } \sigma_i^2 = V[n_i] \,.$$

(Pearson's χ^2 statistic)

χ^2 = sum of squares of the deviations of the ith measurement from the ith prediction, using σ_i as the 'yardstick' for the comparison.

For $n_i \sim$ Poisson(ν_i) we have $V[n_i] = \nu_i$, so this becomes

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

Pearson's χ^2 test

If n_i are Gaussian with mean ν_i and std. dev. σ_i, i.e., $n_i \sim \mathrm{N}(\nu_i\,,\,\sigma_i^2)$, then Pearson's χ^2 will follow the χ^2 pdf (here for $\chi^2 = z$):

$$f_{\chi^2}(z;N) = \frac{1}{2^{N/2}\Gamma(N/2)} z^{N/2-1} e^{-z/2}$$

If the n_i are Poisson with $\nu_i >> 1$ (in practice OK for $\nu_i > 5$) then the Poisson dist. becomes Gaussian and therefore Pearson's χ^2 statistic here as well follows the χ^2 pdf.

The χ^2 value obtained from the data then gives the p-value:

$$p = \int_{\chi^2}^{\infty} f_{\chi^2}(z;N)\,dz\ .$$

The 'χ^2 per degree of freedom'

Recall that for the chi-square pdf for N degrees of freedom,

$$E[z] = N\,, \quad V[z] = 2N\,.$$

This makes sense: if the hypothesized ν_i are right, the rms deviation of n_i from ν_i is σ_i, so each term in the sum contributes ~ 1.

One often sees χ^2/N reported as a measure of goodness-of-fit. But... better to give χ^2 and N separately. Consider, e.g.,

$$\begin{aligned} \chi^2 &= 15,\ N = 10 \ \rightarrow\ p-\text{value} = 0.13\,, \\ \chi^2 &= 150,\ N = 100 \ \rightarrow\ p-\text{value} = 9.0 \times 10^{-4}\,. \end{aligned}$$

i.e. for N large, even a χ^2 per dof only a bit greater than one can imply a small p-value, i.e., poor goodness-of-fit.

Pearson's χ^2 with multinomial data

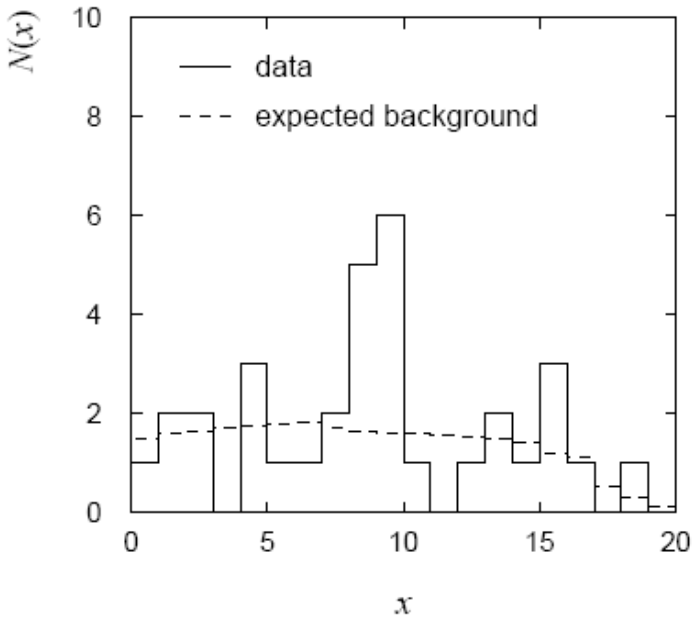
If $n_{\text{tot}} = \sum_{i=1}^{N}$ is fixed, then we might model $n_i \sim$ binomial with $p_i = n_i / n_{\text{tot}}$. I.e. $\vec{n} = (n_1, \ldots, n_N)$ $\sim$ multinomial.

In this case we can take Pearson's χ^2 statistic to be

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

If all $p_i\, n_{\text{tot}} >> 1$ then this will follow the chi-square pdf for $N-1$ degrees of freedom.

Example of a χ^2 test



← This gives

$$\chi^2 = \sum_{i=1}^{N} \frac{(n_i - \nu_i)^2}{\nu_i} = 29.8$$

for $N = 20$ dof.

Now need to find p-value, but... many bins have few (or no) entries, so here we do not expect χ^2 to follow the chi-square pdf.

Using MC to find distribution of χ^2 statistic

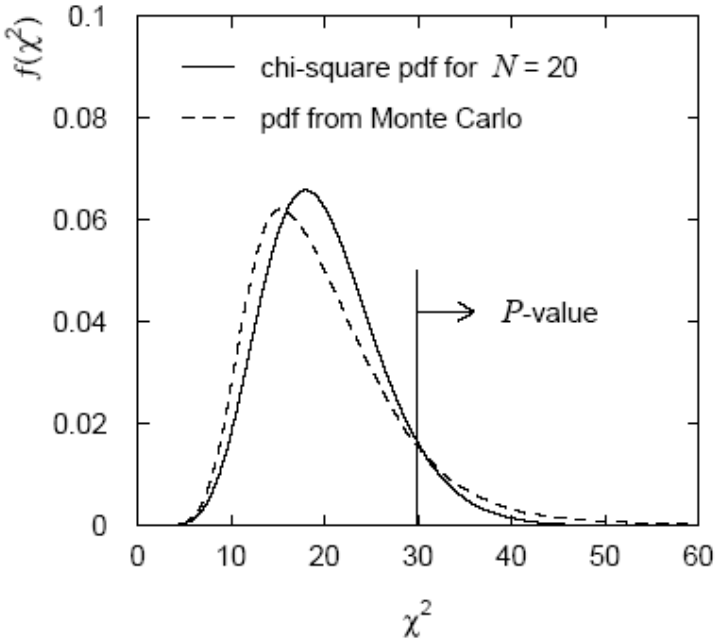
The Pearson χ^2 statistic still reflects the level of agreement between data and prediction, i.e., it is still a 'valid' test statistic.

To find its sampling distribution, simulate the data with a Monte Carlo program: $n_i \sim \mathsf{Poisson}(\nu_i)\,,\ i = 1, N\,.$

Here data sample simulated 10^6 times. The fraction of times we find $\chi^2 > 29.8$ gives the p-value:

$$p = 0.11$$

If we had used the chi-square pdf we would find $p = 0.073$.



Parameter estimation

The parameters of a pdf are constants that characterize its shape, e.g.

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

r.v. parameter

Suppose we have a sample of observed values: $\vec{x} = (x_1, \ldots, x_n)$

We want to find some function of the data to estimate the parameter(s):

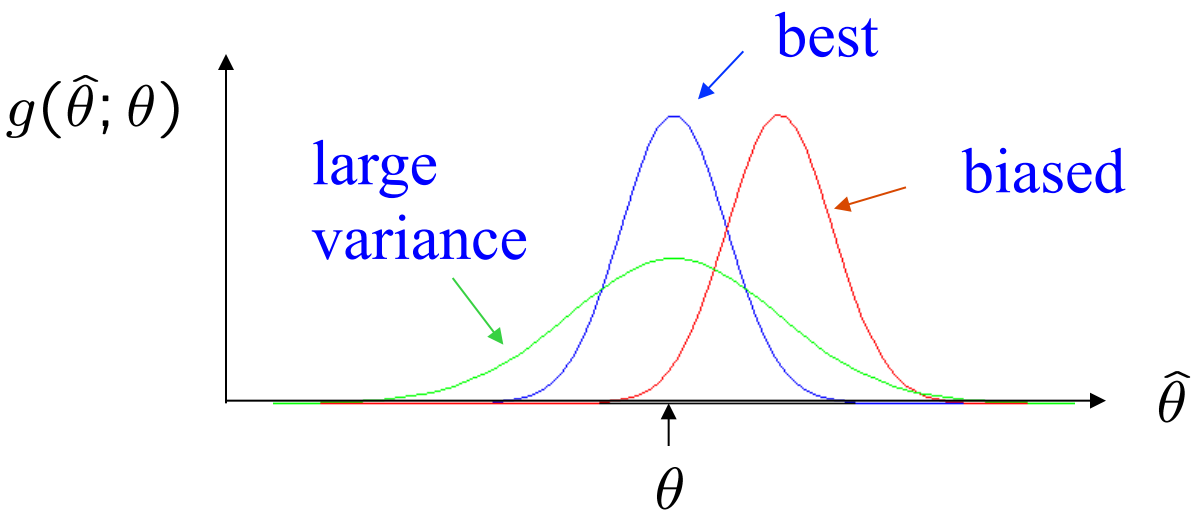
$$\boxed{\hat{\theta}(\vec{x})}$$

← estimator written with a hat

Sometimes we say 'estimator' for the function of x_1, ..., x_n; 'estimate' for the value of the estimator with a particular data set.

Properties of estimators

If we were to repeat the entire measurement, the estimates from each would follow a pdf:



We want small (or zero) bias (systematic error): $b = E[\hat{\theta}] - \theta$

→ average of repeated measurements should tend to true value.

And we want a small variance (statistical error): $V[\hat{\theta}]$

→ small bias & variance are in general conflicting criteria

An estimator for the mean (expectation value)

Parameter: $\mu = E[x]$

Estimator: $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \equiv \overline{x}$ ('sample mean')

We find: $b = E[\hat{\mu}] - \mu = 0$

$$V[\hat{\mu}] = \frac{\sigma^2}{n} \qquad \left(\sigma_{\hat{\mu}} = \frac{\sigma}{\sqrt{n}} \right)$$

An estimator for the variance

Parameter: $\sigma^2 = V[x]$

Estimator: $\widehat{\sigma^2} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2 \equiv s^2$ ('sample variance')

We find:

$$b = E[\widehat{\sigma^2}] - \sigma^2 = 0 \quad \text{(factor of } n\text{−1 makes this so)}$$

$$V[\widehat{\sigma^2}] = \frac{1}{n}\left(\mu_4 - \frac{n-3}{n-1}\mu_2\right) \ , \quad \text{where}$$

$$\mu_k = \int (x-\mu)^k f(x)\, dx$$