Multivariate polynomial regression and fit function uncertainty

Or in short, a regression robot

Overview

- Fitting with polynomials
- The uncertainty of the regression
- Finding the right degree
- Minimising the numerical errors
- Comparing distributions
- Variable selection for multivariate methods
- Classification
- Summary

Polynomial regression in a nutshell

- The training inputs are the moments and correlations of the sample

$$h_k = \langle y x^k \rangle \qquad G_{kl} = \langle x^{k+l} \rangle$$

x : input space

y : target space

- The sum of squared residuals will look like

$$\chi^2 = \frac{1}{N_{\text{sample}}} \sum_{i=1}^{N_{\text{sample}}} (y_i - F(x_i))^2$$
$$= \langle y^2 \rangle - 2F_k h_k + F_k G_{kl} F_l$$

- The F_k polynomial coefficients that minimises it are

$$F_k = G^{-1}_{kl} h_l$$

- The number of degrees n can be freely chosen,

$$F_k = 0 \text{ for } k > d_{\max}$$

Input uncertainty estimation

$$p = (\langle yx^0 \rangle, \ldots, \langle yx^d \rangle, \langle x^0 \rangle, \ldots, \langle x^{2d} \rangle)$$

- Unlike the individual x_i points in the sample, $h_k = \langle yx^k \rangle$ and $G_{kl} = \langle x^{k+l} \rangle$ are subject to the central limit theorem
- Their uncertainties, correlations are Gaussians with an error matrix

$$\Sigma_{kl} = \mathrm{Cov}(p_k, p_l)$$

$$\mathrm{Cov}(\langle x^k \rangle, \langle x^l \rangle) = \frac{\langle x^k x^l \rangle - \langle x^k \rangle \langle x^l \rangle}{N}$$

- In short :
  - this is the uncertainty of the sample with N events, the empirical distribution
  - describes what would we measure for the moments, if we had an infinitely large sample: the likelihood where the parameters may converge
  - a sample holds information about its generating distribution and the uncertainty about it

Input uncertainty estimation

$$p = (\langle yx^0 \rangle, \ldots, \langle yx^d \rangle, \langle x^0 \rangle, \ldots, \langle x^{2d} \rangle)$$

- Unlike the individual x_i points in the sample, $h_k = \langle yx^k \rangle$ and $G_{kl} = \langle x^{k+l} \rangle$ are subject to the central limit theorem

- Their uncertainties, correlations are Gaussians with an error matrix

For weighted sample

$$\Sigma_{ij} = \mathrm{Cov}(\langle x^i \rangle, \langle x^j \rangle) = \sum_k w_k^2 x_k^i x_k^j - \frac{1}{N_{\mathrm{eff}}} \langle x^i \rangle \langle x^j \rangle$$

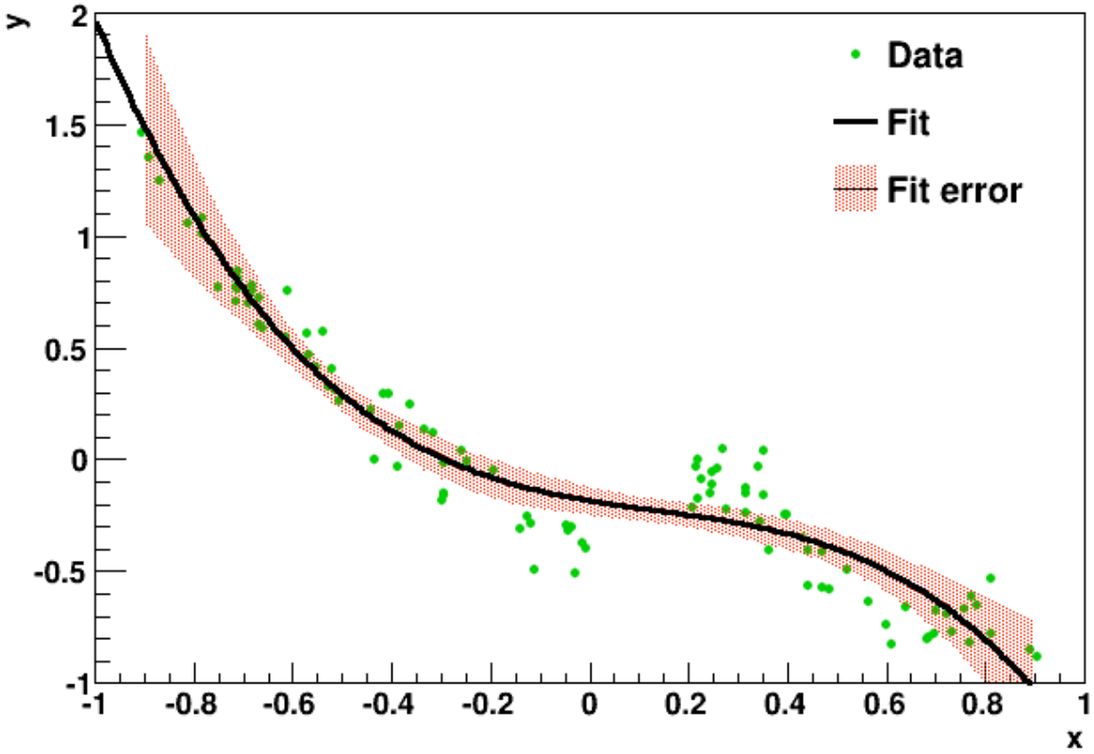
- In short :
  - this is the uncertainty of the sample with N events, the empirical distribution
  - describes what would we measure for the moments, if we had an infinitely large sample: the likelihood where the parameters may converge
  - a sample holds information about its generating distribution and the uncertainty about it

F(x) uncertainty

- The $F_k x^k$ polynomial is known to a finite order, its approximate uncertainty is:

$$\frac{\partial F_k}{\partial p_l} = \begin{pmatrix} G^{-1}_{kl} \\ -h_m G^{-1}_{mn} \frac{\partial G_{no}}{\partial p_l} G^{-1}_{ok} \end{pmatrix} \qquad \sigma^2_{F(x)} = \frac{\partial F_k x^k}{\partial p_l} \sum_{lm} \frac{\partial F_k x^k}{\partial p_m}$$

- The uncertainty of $F_k x^k$ is only useful when we are sure there are no higher degrees



The uncertainty of the residual

- It is possible to calculate the uncertainty of the residual, but note
  - F_k is uncertain, calculated from the uncertain G and h
  - What can be calculated is the residual of the varied F_k on the fixed measured sample

$$\chi^2 = \langle y^2 \rangle^{\text{fixed}} - 2F_k h_k^{\text{fixed}} + F_k G_{kl}^{\text{fixed}} F_l$$

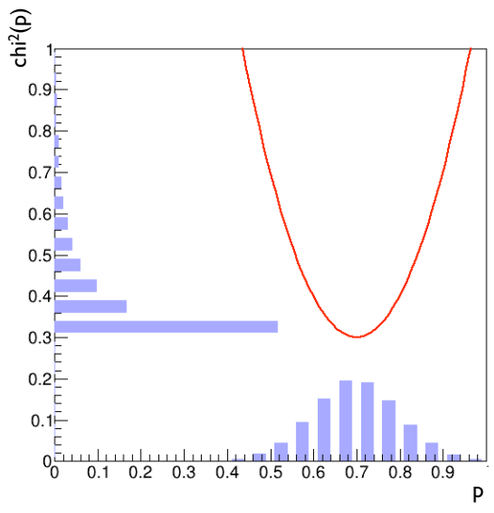
- The first derivative is zero, so for error propagation one has to use the second derivative

$$\frac{\partial \chi^2}{\partial p_k} = 0 \qquad \frac{\partial^2 \chi^2}{\partial p_k \partial p_l} = \begin{pmatrix} 2G_{kl}^{-1} & -4G^{-1}\frac{\partial G}{\partial p_l}G^{-1}h \\ -4G^{-1}\frac{\partial G}{\partial p_k}G^{-1}h & 6hG^{-1}\frac{\partial G}{\partial p_k}G^{-1}\frac{\partial G}{\partial p_k}G^{-1}h \end{pmatrix}$$

- This creates a bias in the expectation value and a deviation

$$b_{\chi^2} = \mathbb{E}[\chi^2_{\text{meas}} - \chi^2_{p+\Delta p}] = \tfrac{1}{2}\Sigma_{kl}\frac{\partial^2 \chi^2}{\partial p_k \partial p_l}$$

$$\sigma^2_{\chi^2} = \mathbb{E}\left[\left(\chi^2_{\text{meas}} - \chi^2_{p+\Delta p}\right)^2\right] - b^2_{\chi^2} = 3\left(\tfrac{1}{2}\Sigma_{kl}\frac{\partial^2 \chi^2}{\partial p_k \partial p_l}\right)^2 - b^2_{\chi^2} = 2b^2_{\chi^2}$$



The uncertainty of the residual

- It is possible to calculate the uncertainty of the residual, but note
  - F_k is uncertain, calculated from the uncertain G and h
  - What can be calculated is the residual of the varied F_k on the fixed measured sample

$$\chi^2 = \langle y^2 \rangle^{\text{fixed}} - 2F_k h_k^{\text{fixed}} + \dots$$

These terms correspond to the randomness of the input - sometimes can be left out

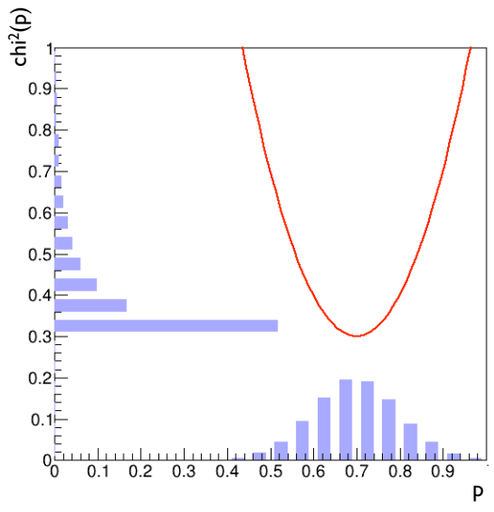
- The first derivative is zero, so for error propagation one has to use the second derivative

$$\frac{\partial \chi^2}{\partial p_k} = 0 \qquad \frac{\partial^2 \chi^2}{\partial p_k \partial p_l} = \begin{pmatrix} 2G^{-1}_{kl} & -4G^{-1}\frac{\partial G}{\partial p_l}G^{-1}h \\ -4G^{-1}\frac{\partial G}{\partial p_k}G^{-1}h & 6hG^{-1}\frac{\partial G}{\partial p_k}G^{-1}\frac{\partial G}{\partial p_k}G^{-1}h \end{pmatrix}$$

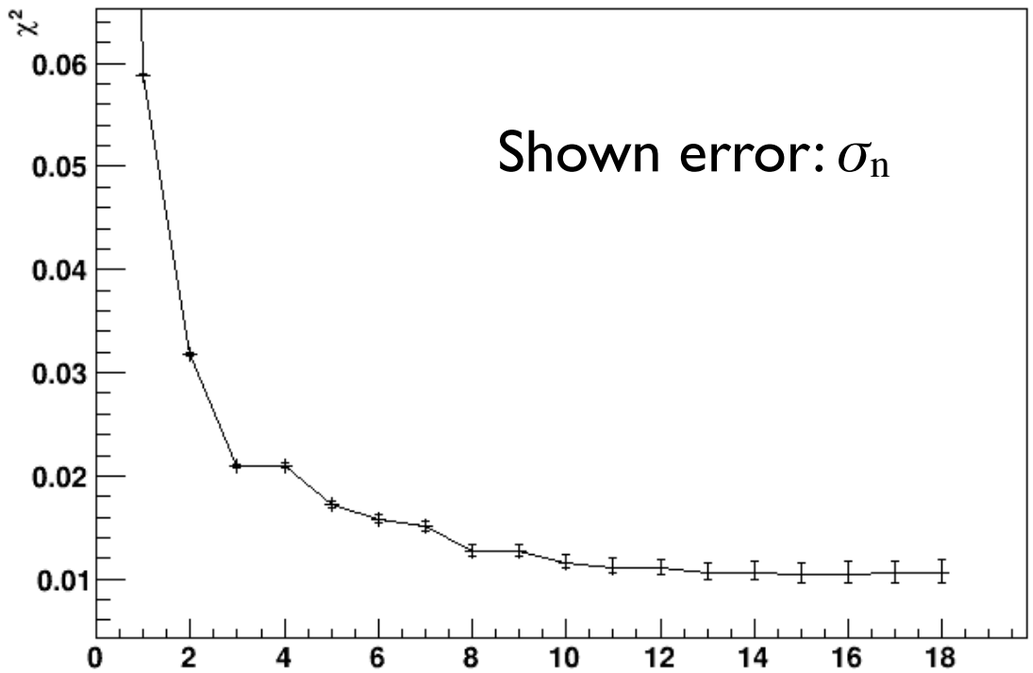
- This creates a bias in the expectation value and a deviation

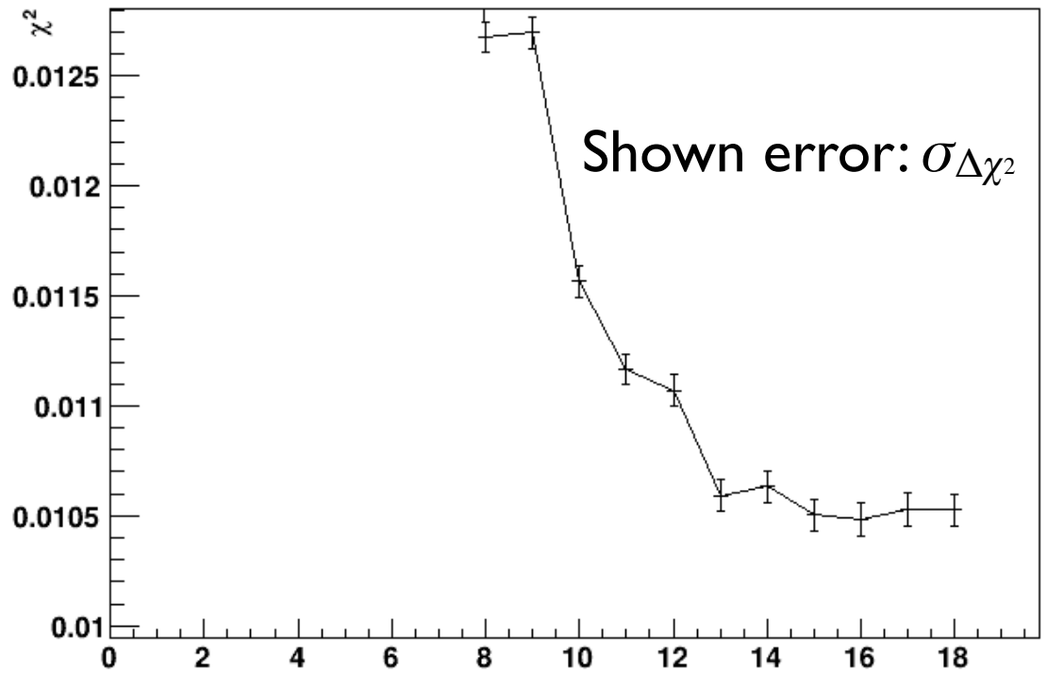
$$b_{\chi^2} = \mathbb{E}[\chi^2_{\text{meas}} - \chi^2_{p+\Delta p}] = \tfrac{1}{2}\Sigma_{kl}\frac{\partial^2 \chi^2}{\partial p_k \partial p_l}$$

$$\sigma^2_{\chi^2} = \mathbb{E}\left[\left(\chi^2_{\text{meas}} - \chi^2_{p+\Delta p}\right)^2\right] - b^2_{\chi^2} = 3\left(\tfrac{1}{2}\Sigma_{kl}\frac{\partial^2 \chi^2}{\partial p_k \partial p_l}\right)^2 - b^2_{\chi^2} = 2b^2_{\chi^2}$$

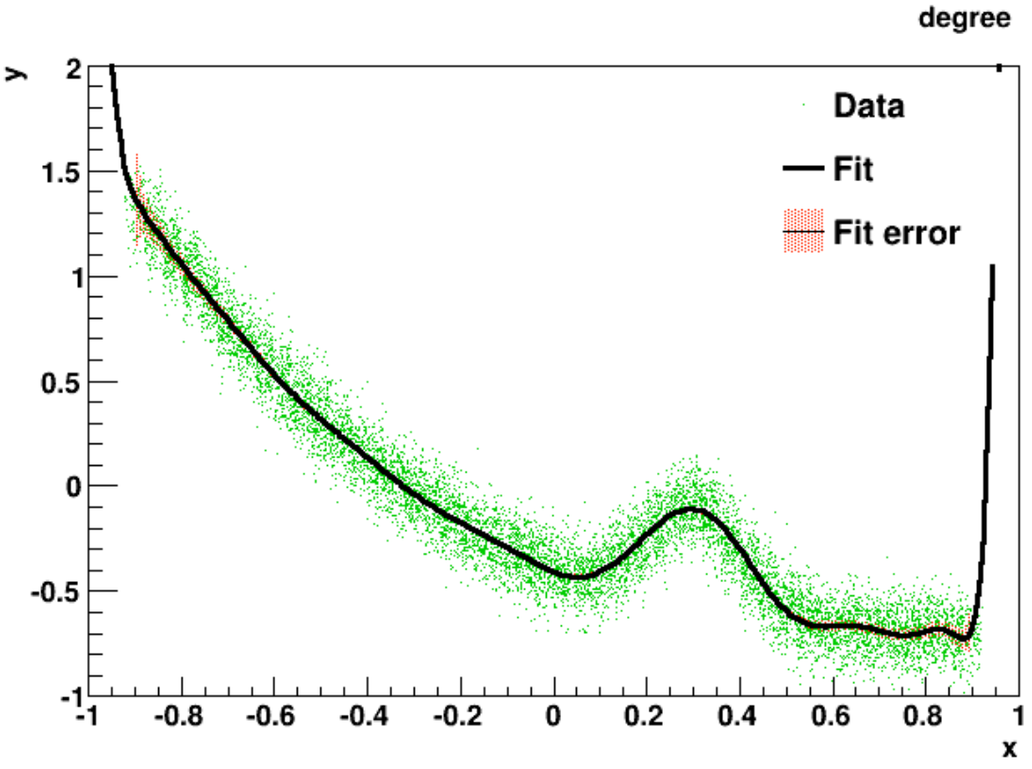


Using the expected residual





- The expected residual, χ^2+b will start to increase with the degree
- Possible to calculate the uncertainty of residual differences
  $\sigma_{\Delta\chi^2} = \sigma_n - \sigma_{n-1} < \sigma_n \rightarrow$ minimise $\chi^2+b+\sigma$
- Regularisation is possible with:
  - Checking significance of $\Delta\chi^2$
  - Finding smallest expected χ^2

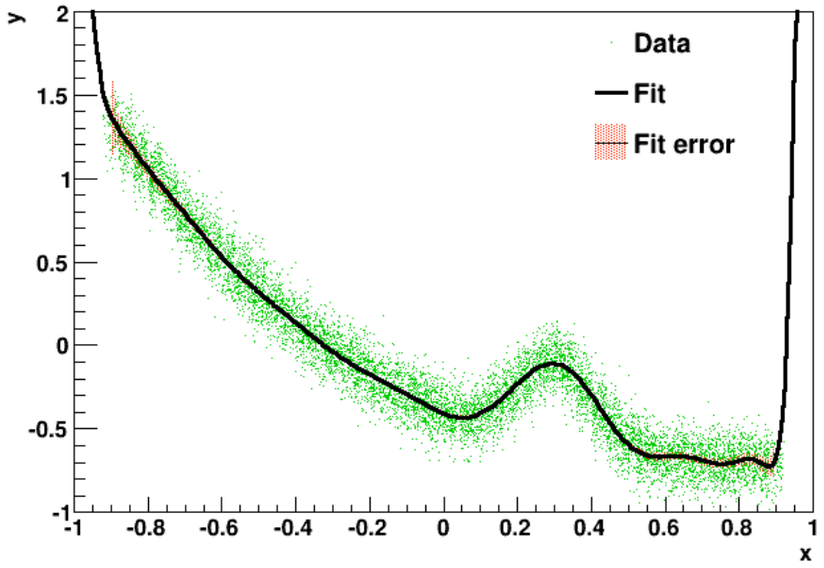


Minimising numerical errors

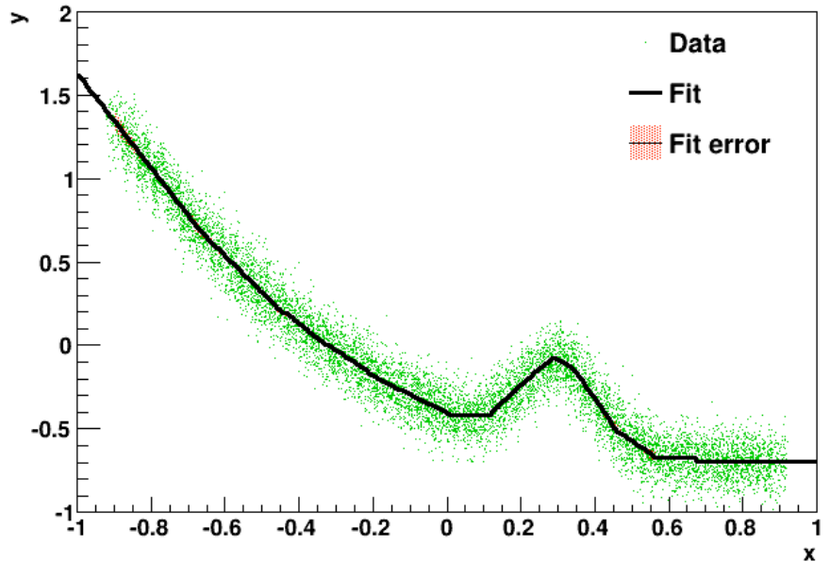
- Calculating higher degree momenta introduces numerical errors
  - many multiplications and subtractions
  - different degrees will produce numbers on different order of magnitude
  - large matrices have to be inverted (especially in higher dimensional input)
- One can split the input phase space for smaller function complexity
  - The different decision tree algorithms only weakly influence the result
  - One simple method is:
    Cut perpendicular to the principal eigenvector of the x input distribution at the mean - this ‘equalises’ the length in the different input directions and produces small, compact regions

Regression Robot

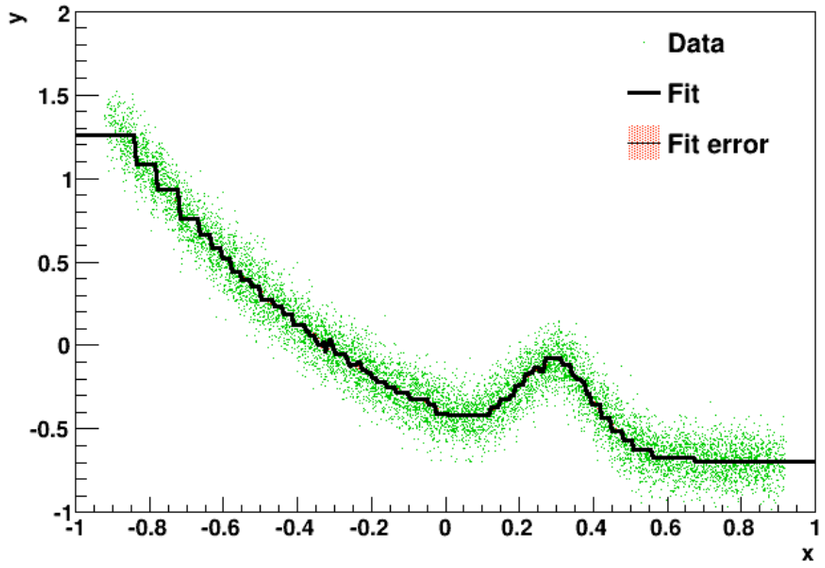
max.degree = 16



max.degree = 1



max.degree = 0



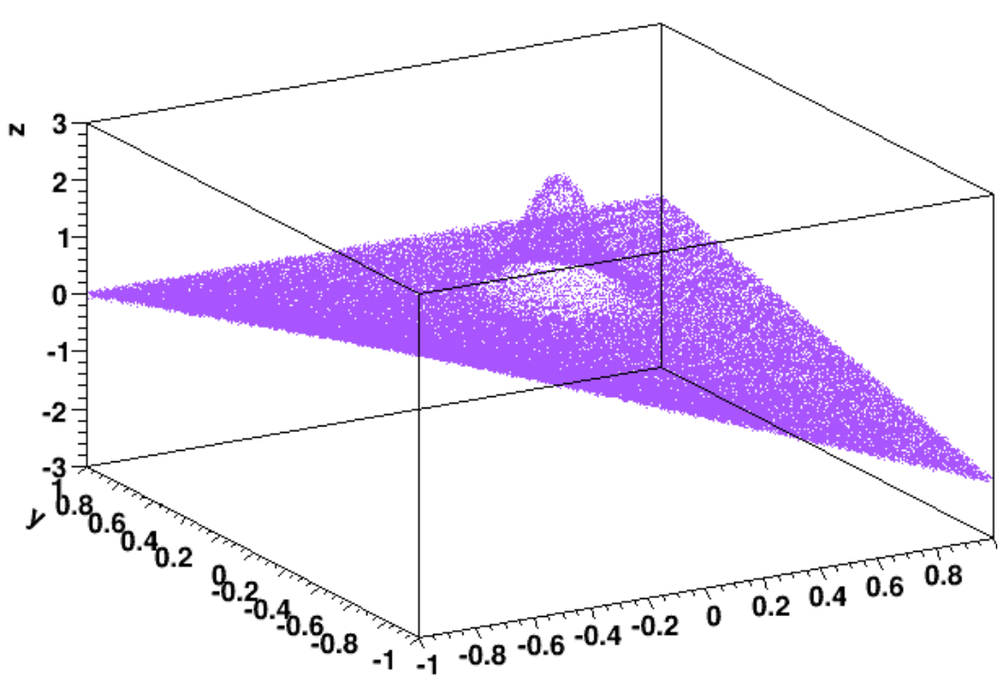
- If a given segment is more complex than the maximal allowed degree, it is split
- Using different max.degrees have little effect at the bulk of the sample, but it differs how they extrapolate

Multivariate input

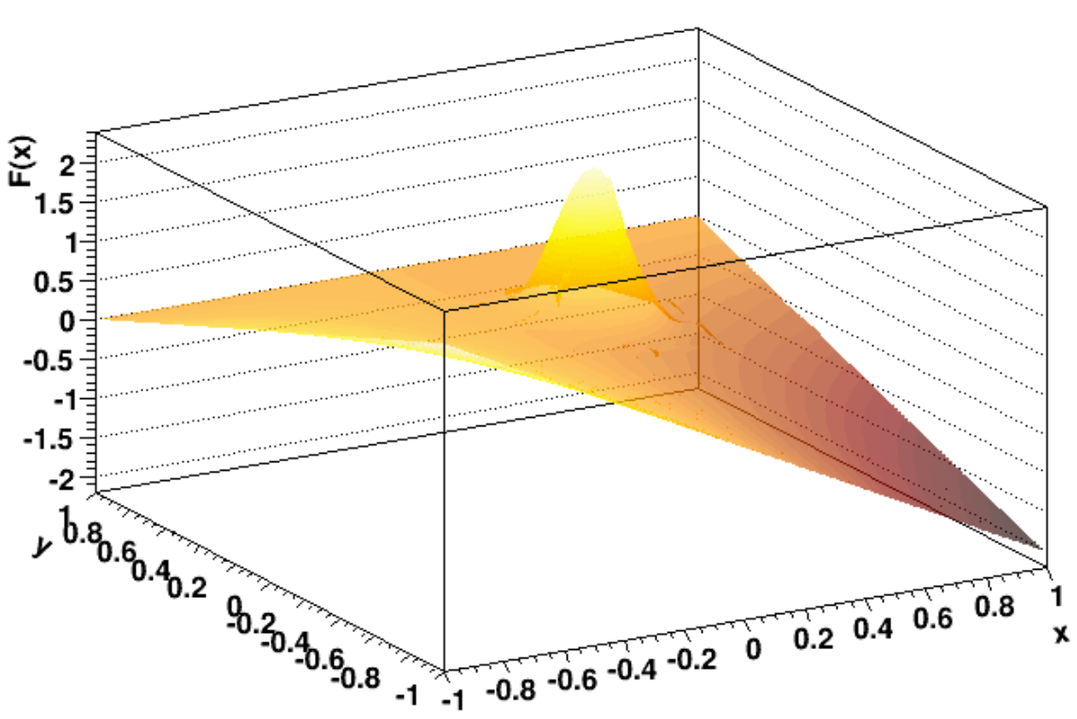
- The polynomial coefficients are symmetric tensors

$$F(x) = F_0 + F_{1\mu}x_\mu + F_{2\mu\nu}x_\mu x_\nu \ldots$$

- The n^{th} degree d dimensional tensor has $\binom{n+d-1}{n}$ free parameters
- It is a large reduction, but still goes rapidly with n and d



3D histogram

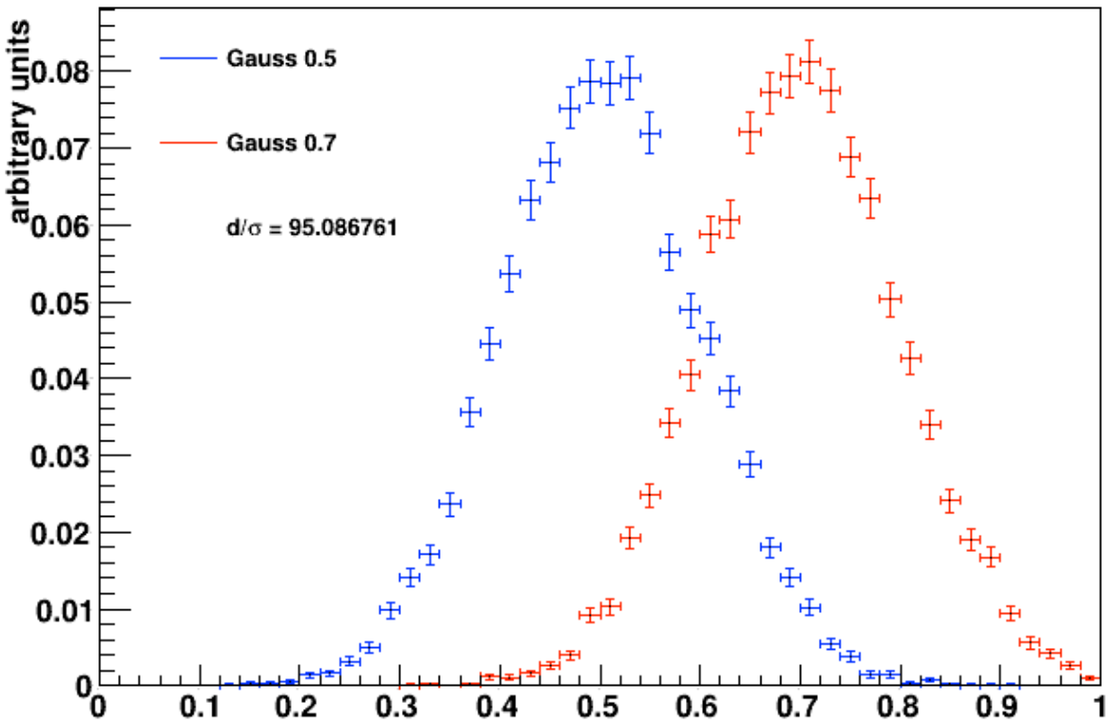


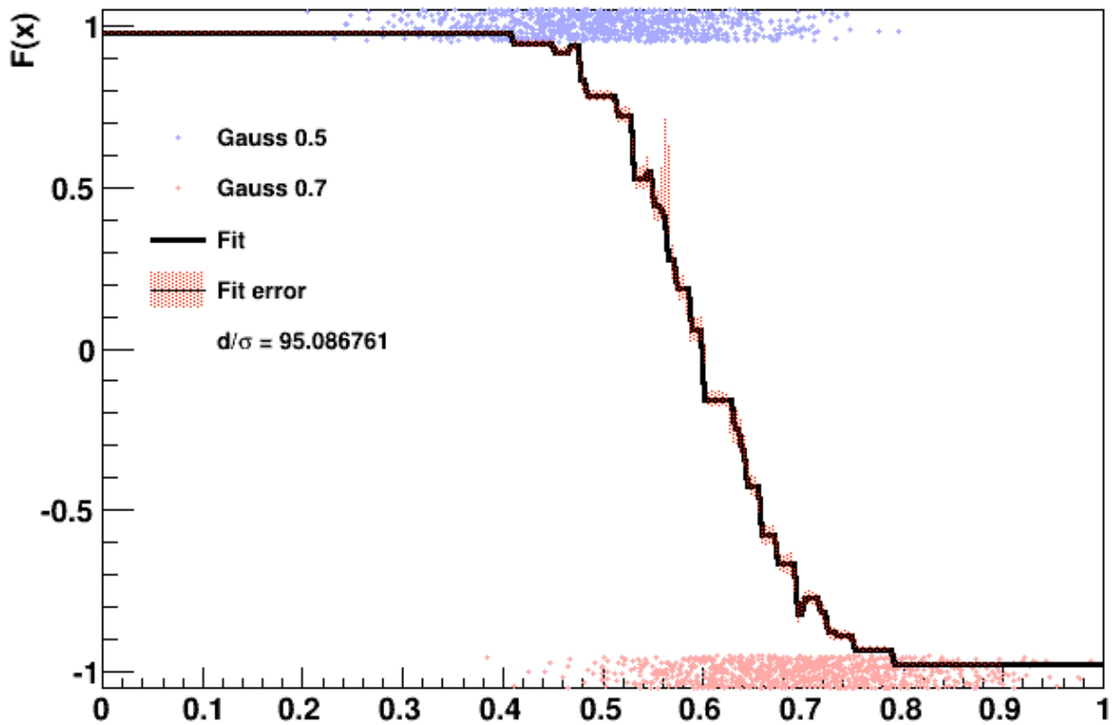
2D function with max.degree = 3

Efficient feature discovery

- Non-random 3D data, but interpreted as random
- Used $z=I$ intensity and $\sqrt{I}$ weights and weight 1 for $I=0$, max.degree = 1 + second layer
- Many details were found, automatically

Distribution comparison





- Using max.degree = 0 results in automatic variable binning
- It can be used for distribution comparison
- Regression between the samples with $y=\pm 1$
- Comparison with null hypothesis (significance): $\left(\frac{d}{\sigma}\right)^2 = \frac{1}{N}\sum_{i\in\{\text{bins}\}} N_i \frac{F(x_i)^2}{\sigma_F^2(x_i)}$
- The $(y\text{-}F(x))^2$ measure, polynomials and the slicing method are not ideal for fitting discrete y

Variable selection

- The task of variable selection is maximising the significance of a variable set
- A simplified way of distribution comparison:
  checking whether the moments of sample s and b are significantly different

$$p_k = \langle x^k \rangle_s - \langle x^k \rangle_b \stackrel{?}{=} 0 \ , \forall k$$

- Likelihood of this vector:

$$\mathcal{L}(p) = \mathrm{Prob}(s, b|p) = \frac{1}{\mathcal{N}} \exp\left(-\frac{1}{2} \left(p_k - \langle x^k \rangle_s + \langle x^k \rangle_b \right) \left(\Sigma^s_{kl} + \Sigma^s_{kl} \right)^{-1} \left(p_l - \langle x^l \rangle_s + \langle x^l \rangle_b \right) \right)$$

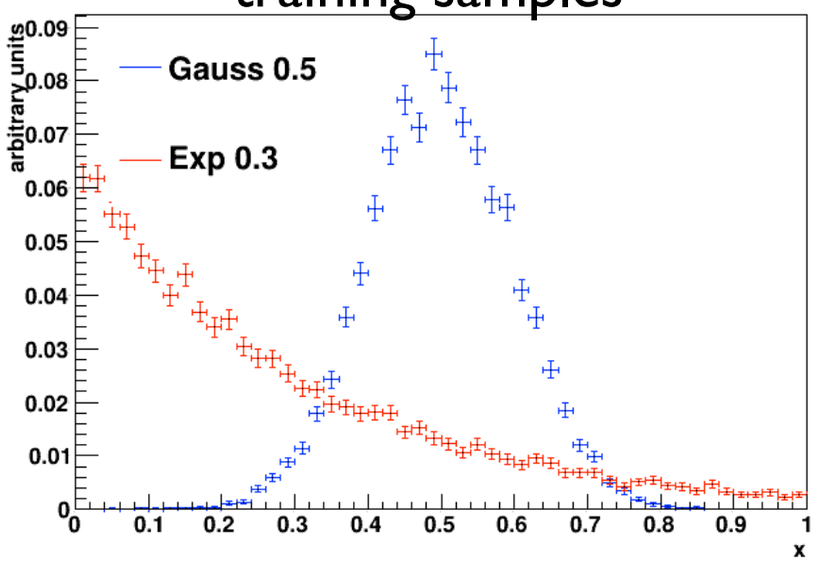
- Testing whether p was zero

$$\left(\frac{d}{\sigma} \right)^2 = \left(\langle x^k \rangle_s - \langle x^k \rangle_b \right) \left[\Sigma^{\mathrm{s}}_{kl} + \Sigma^{\mathrm{b}}_{kl} \right]^{-1} \left(\langle x^l \rangle_s - \langle x^l \rangle_b \right)$$

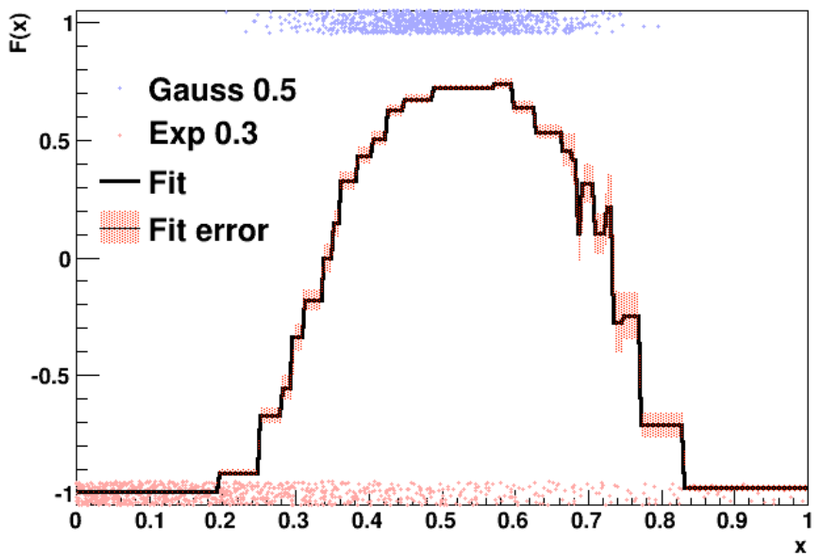
- The first few moments can still give a good approximation
- Heuristic solution: start with the most significant variable then add more

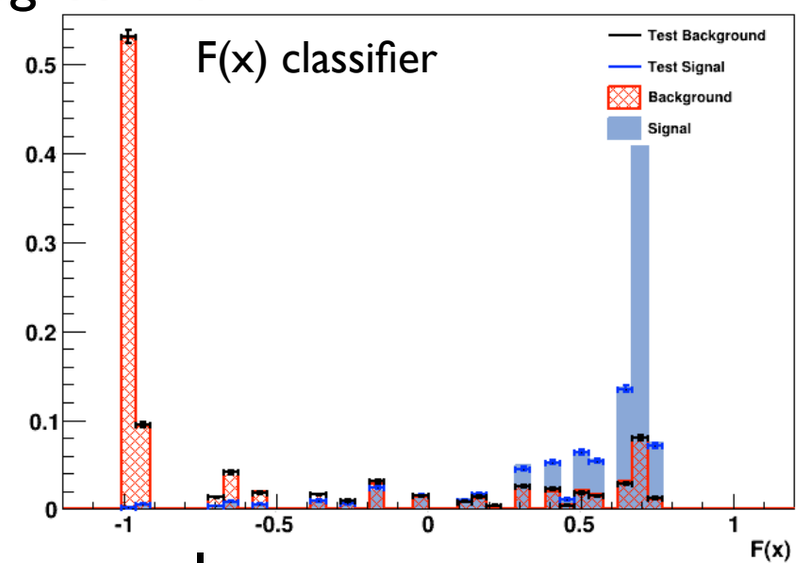
Classification

training samples

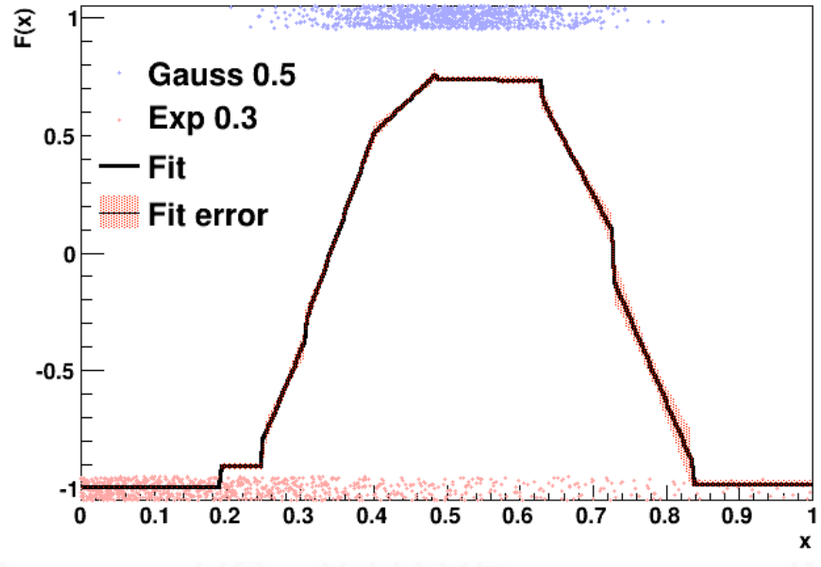


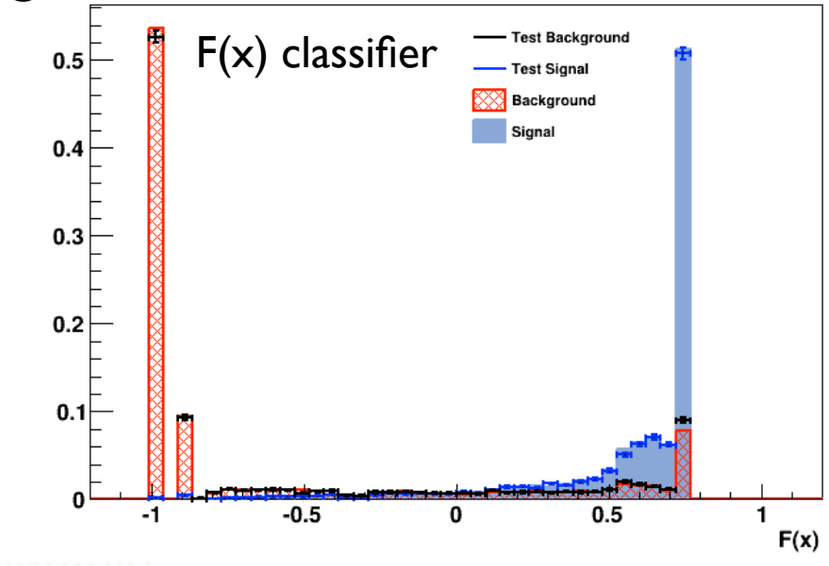
max.degree = 0





max.degree = 1





- Regression of target $y=\pm 1$
- The difference of the expected residuals had to be more than $1\ \sigma_{\text{std}}$
- The local max.degree can be nonzero, as the d/σ of the sample difference is not important, only the significance of the residuals

Overview of the steps

- Parametrisation viewpoint: using the most important averages

$$p = (\langle yx^0 \rangle, \ldots, \langle yx^d \rangle, \langle x^0 \rangle, \ldots, \langle x^{2d} \rangle)$$

- Central Limit Theorem: tells us the likelihood of the true parametrisation

$$\mathcal{L}(p^{\text{true}}) = \text{Prob}(p|p^{\text{true}}) = \frac{1}{\mathcal{N}} \exp\left(-\frac{1}{2}\left(p_k - p_k^{\text{true}}\right)\left(\Sigma_{kl}^{\text{est.}}\right)^{-1}\left(p_l - p_l^{\text{true}}\right)\right)$$

- Bayesian step: using the derived likelihood to calculate the probability of the residual (the divergence of the fit from the minimum) and residual differences:

$$\text{Prob}(\chi^2 \text{ of } F^{\text{true}} \text{ at } p|p)$$

- Decision step: using the polynomial degree that is
  - expected to be the best
  - significantly better than the others
- Repeat the steps on smaller input space if needed

Summary

- Fast polynomial regression is done based on the moments of the distribution
- The uncertainty of the fit function is estimated from data analytically
- No need for :
  - separating training and testing samples
  - bootstrapping
  - parametric knowledge of the distribution (as for Fisher information)
- Multivariate analysis can be done with given (approx.) significance level
- Variable preselection for the MVA is possible
- There is still room for improvement for speed and accuracy

Some references

- The Hamburger moment problem
  http: //www.encyclopediaofmath.org/index.php?title=Hankel_matrix&oldid=23850

- Fisher information, polynomial regression
  Statistical Methods in Data Analysis, W. J. Metzger
  HEN-343

- Bootstrap Confidence Intervals
  DiCiccio, Efron Statistical Science, 1996, Vol. 11, No. 3, 189–228
  http://www.jstor.org/stable/2246110

- Support vector machines
  Pattern Recognition and Machine Learning, Christopher M. Bishop
  ISBN:0387310738

- My thesis
  https://cds.cern.ch/record/1530811?ln=en

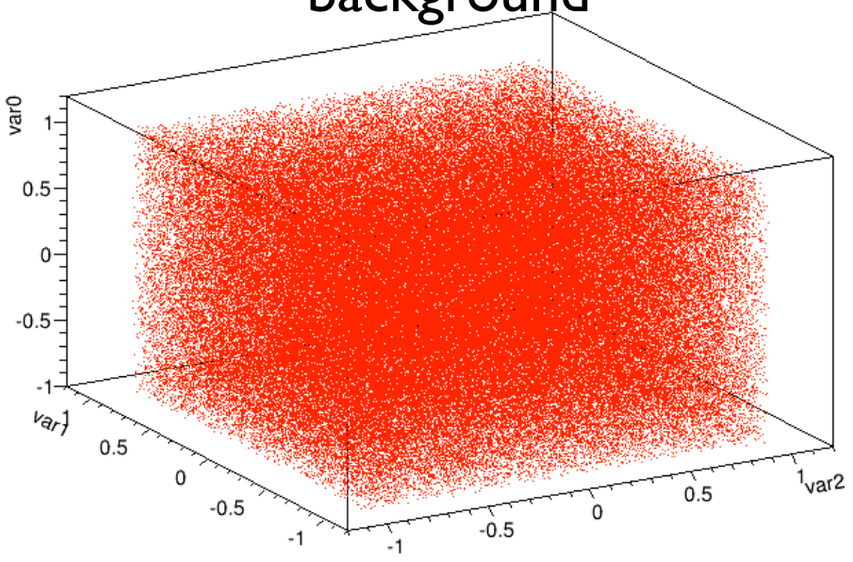
- Polynomial regression in classification
  http://arxiv.org/abs/1203.5647

Thank you!

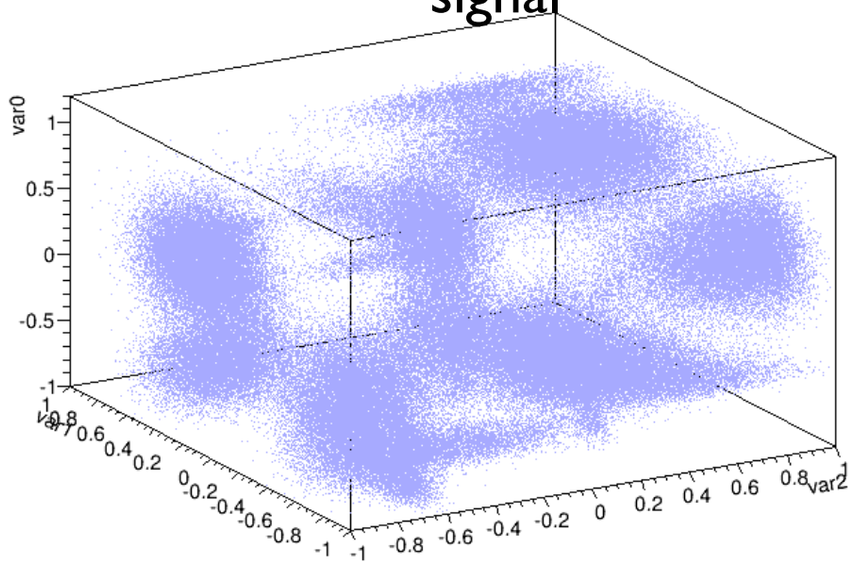
Backup

Multivariate analysis

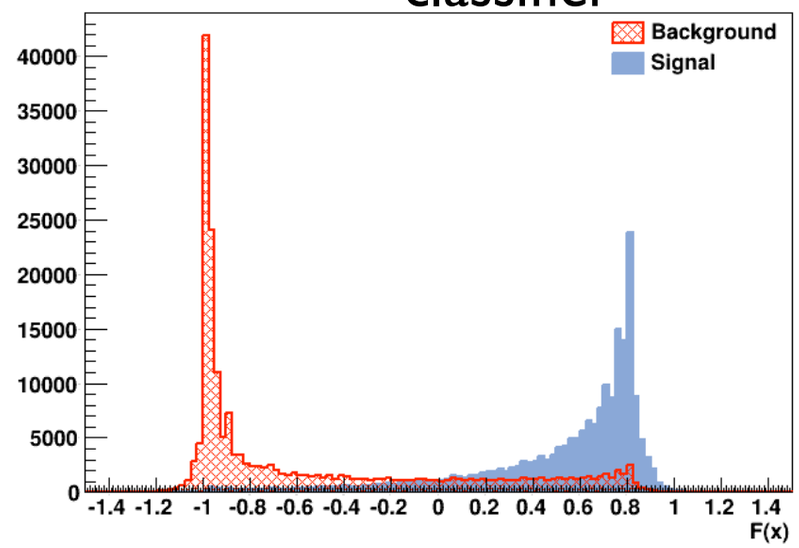
background



signal



classifier



- 3D multivariate analysis
- flat background, 12 Gaussian signal, $2*10^5$ events each

Polynomial Regression

- The ideal classification function :

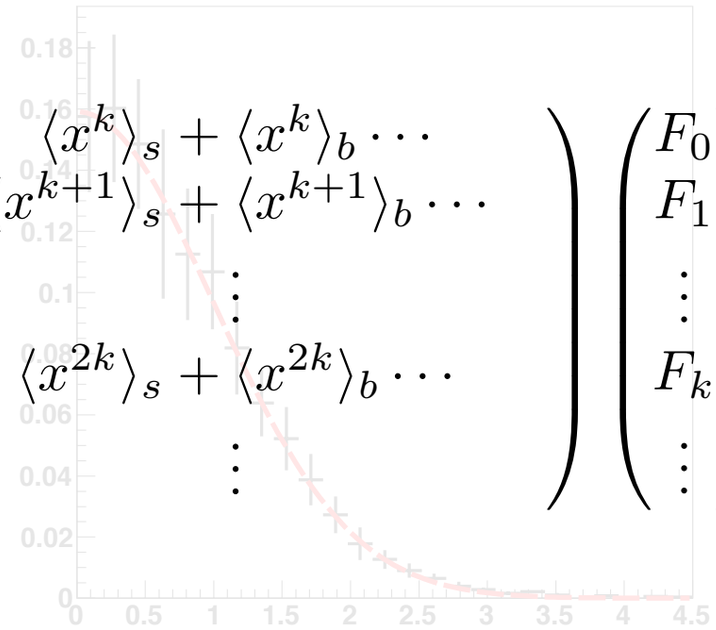
$$F(x) = \sum_{k=0}^{\infty} F_k x^k = \frac{s(x) - b(x)}{s(x) + b(x)}$$

- Its Taylor series, being Fourier transformed:

$$\sum_{k=0}^{\infty} F_k \int_{\mathbb{R}} x^k \left(s(x) + b(x)\right) e^{i\omega x} \, \mathrm{d}x = \int_{\mathbb{R}} \left(s(x) - b(x)\right) e^{i\omega x} \, \mathrm{d}x$$

- F_k can be obtained by solving a linear equation, no optimisation is needed

$$\begin{pmatrix} \langle x^0 \rangle_s - \langle x^0 \rangle_b \\ \langle x^1 \rangle_s - \langle x^1 \rangle_b \\ \vdots \\ \langle x^k \rangle_s - \langle x^k \rangle_b \\ \vdots \end{pmatrix} = \begin{pmatrix} \langle x^0 \rangle_s + \langle x^0 \rangle_b & \langle x^1 \rangle_s + \langle x^1 \rangle_b & \cdots & \langle x^k \rangle_s + \langle x^k \rangle_b \cdots \\ \langle x^1 \rangle_s + \langle x^1 \rangle_b & \langle x^2 \rangle_s + \langle x^2 \rangle_b & \cdots & \langle x^{k+1} \rangle_s + \langle x^{k+1} \rangle_b \cdots \\ \vdots & \vdots & \ddots & \vdots \\ \langle x^k \rangle_s + \langle x^k \rangle_b & \langle x^{k+1} \rangle_s + \langle x^{k+1} \rangle_b & \cdots & \langle x^{2k} \rangle_s + \langle x^{2k} \rangle_b \cdots \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix} \begin{pmatrix} F_0 \\ F_1 \\ \vdots \\ F_k \\ \vdots \end{pmatrix}$$



Polynomial Regression

- The ideal classification function :

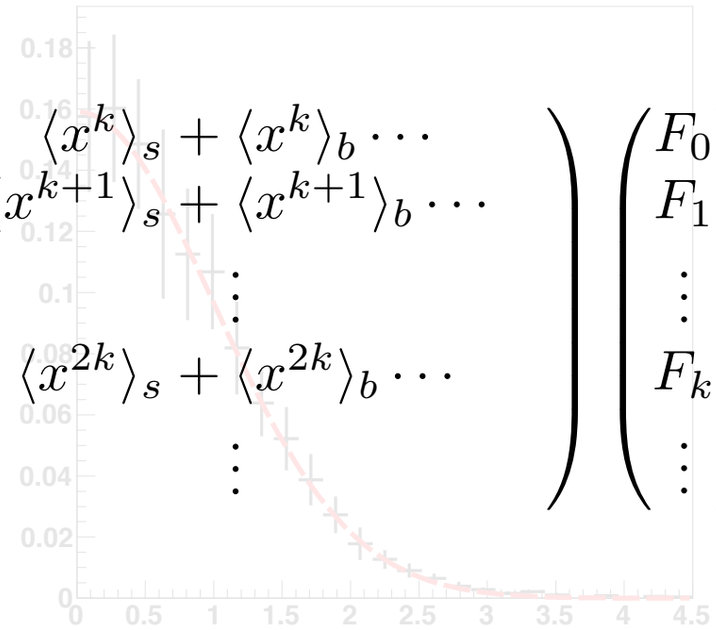
$$F(x) = \sum_{k=0}^{\infty} F_k x^k = \frac{s(x) - b(x)}{s(x) + b(x)}$$

- Its Taylor series, being Fourier transformed:

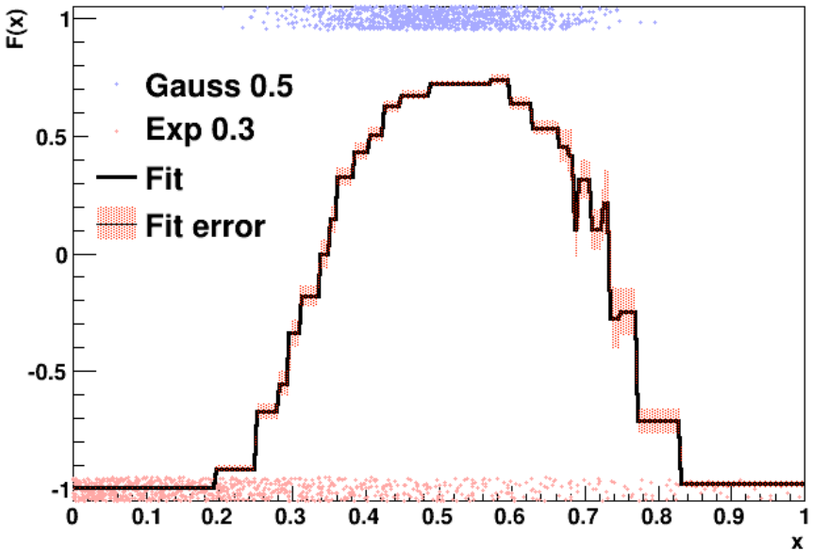
$$\sum_{k=0}^{\infty} F_k \int_{\mathbb{R}} x^k \left(s(x) + b(x)\right) e^{i\omega x} \, \mathrm{d}x = \int_{\mathbb{R}} \left(s(x) - b(x)\right) e^{i\omega x} \, \mathrm{d}x$$

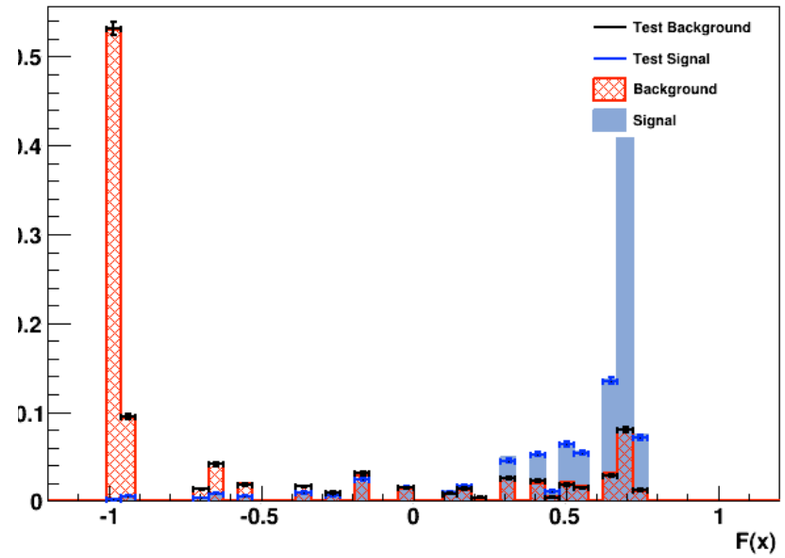
- F_k can be obtained by solving a linear equation, no optimisation is needed

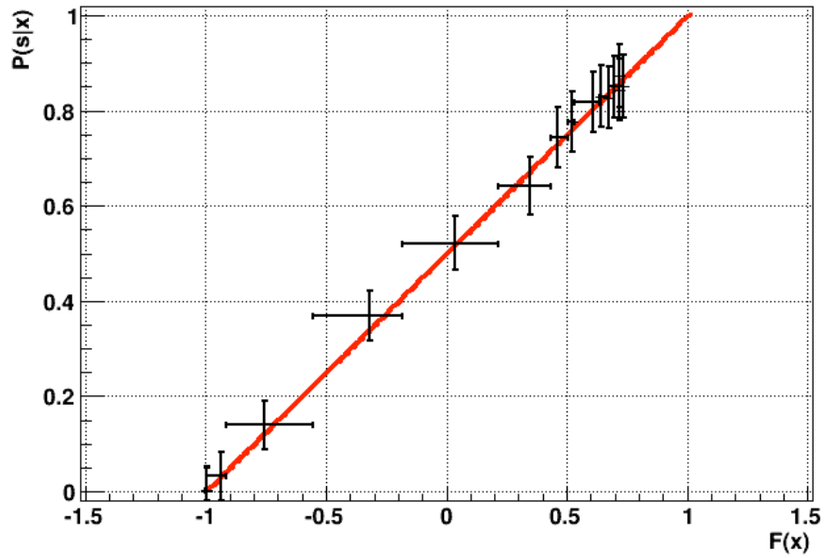
$$\begin{pmatrix} \langle x^0 \rangle_s - \langle x^0 \rangle_b \\ \langle x^1 \rangle_s - \langle x^1 \rangle_b \\ \vdots \\ \langle x^k \rangle_s - \langle x^k \rangle_b \\ \vdots \end{pmatrix} = \begin{pmatrix} \langle x^0 \rangle_s + \langle x^0 \rangle_b & \langle x^1 \rangle_s + \langle x^1 \rangle_b & \cdots & \langle x^k \rangle_s + \langle x^k \rangle_b \cdots \\ \langle x^1 \rangle_s + \langle x^1 \rangle_b & \langle x^2 \rangle_s + \langle x^2 \rangle_b & \cdots & \langle x^{k+1} \rangle_s + \langle x^{k+1} \rangle_b \cdots \\ \vdots & \vdots & \ddots & \vdots \\ \langle x^k \rangle_s + \langle x^k \rangle_b & \langle x^{k+1} \rangle_s + \langle x^{k+1} \rangle_b & \cdots & \langle x^{2k} \rangle_s + \langle x^{2k} \rangle_b \cdots \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix} \begin{pmatrix} F_0 \\ F_1 \\ \vdots \\ F_k \\ \vdots \end{pmatrix}$$

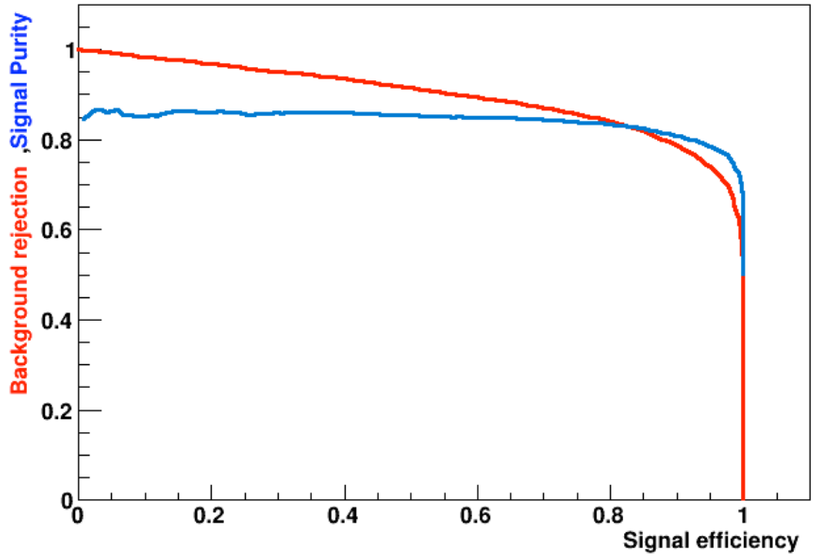


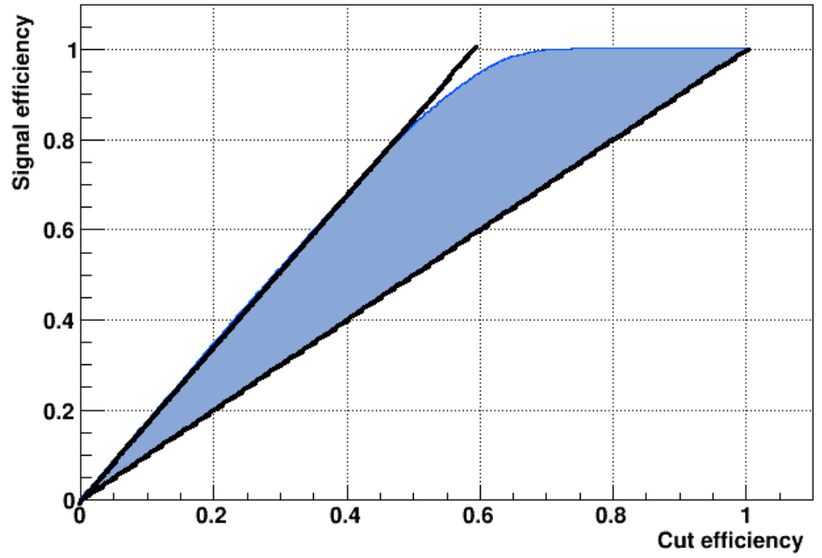
1D classification





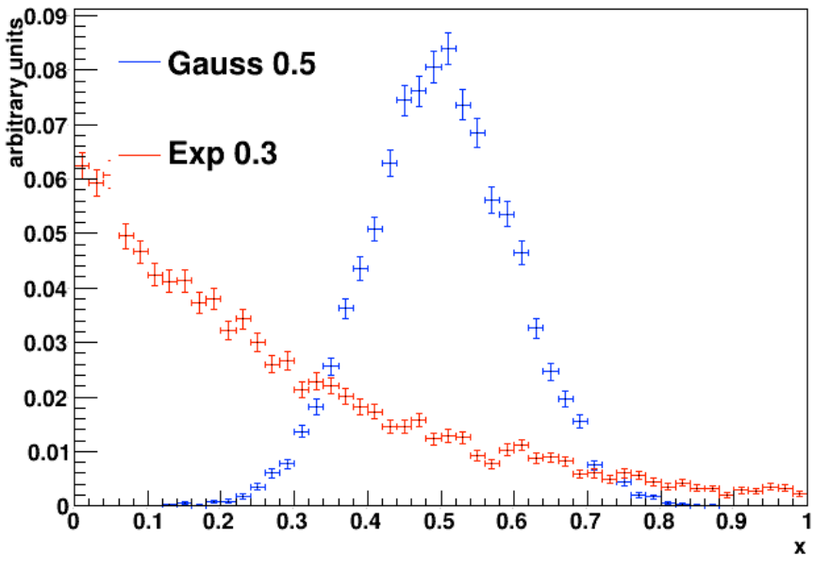


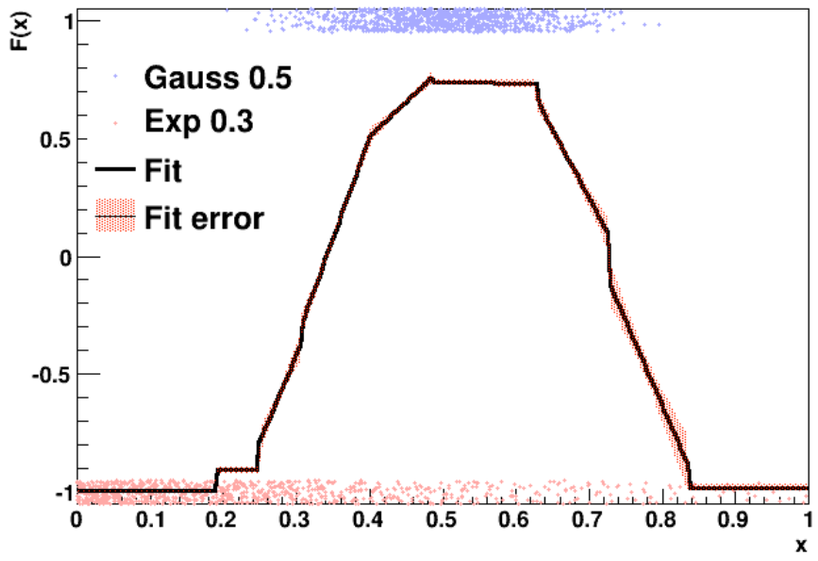


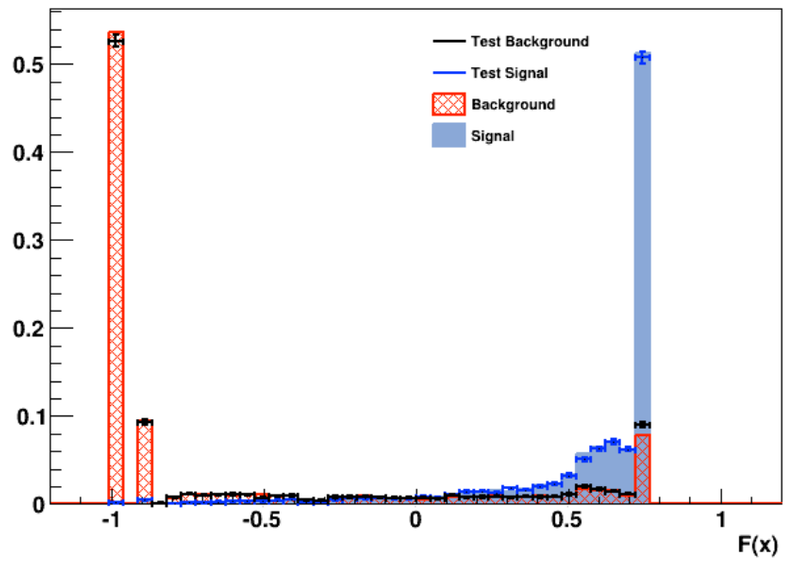


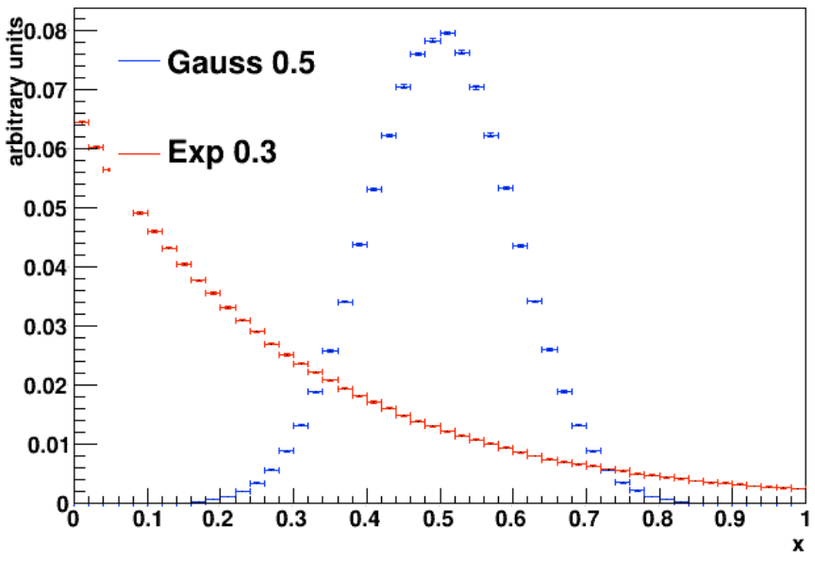
- Practically no overtraining
- The code calculates ROC, Lorenz curve and signal probability

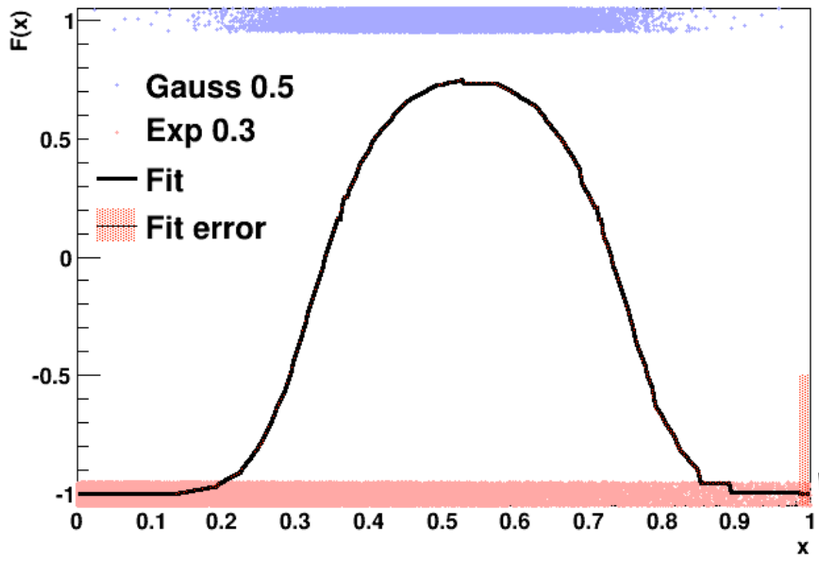
Different sample sizes

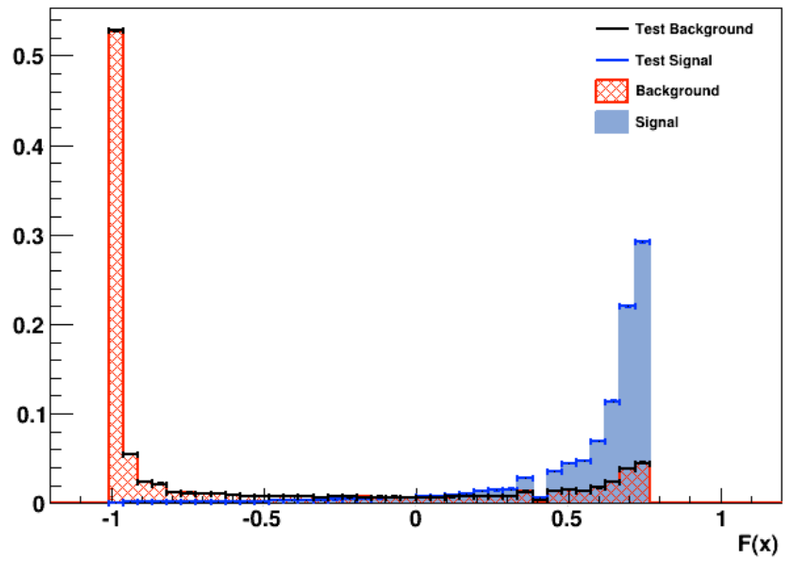












Fit σ was NaN

- Training with 10^4 and 10^6 events