Point estimation and RooFit

Rahul Tiwary

TIFR, Mumbai

Outline

- Statistical basis of estimation
- OLS and MLE
- RooFit (model, dataset)
- Example fits
- Toy study and fit bias

Sources:

- RooFit Manual, Quick Start Guide
- Online tutorials (Verkerke, Rembser, Hehn)
- Others: F. James, Statistical Methods in Experimental Physics; Luca Lista, Statistical Methods for Data Analysis; Practical Statistics for the LHC [arXiv:1503.07622].

Statistical basis of estimation

- `Statistic`: A function of random variables.
- `Estimator`: A function of random variables used to estimate an unknown parameter of a PDF.
- `Point estimator` ($\hat{\theta}$): A point estimator is a statistical estimator whose value can be represented geometrically in the form of a point in the same space as the values of the unknown parameters.
- `Consistent estimator`: Estimates converge toward the true value of the parameter as the number of observations increases.
- `Bias`: The deviation of the expectation of $\hat{\theta}$, from the true value θ_0 $\Rightarrow b_N(\hat{\theta})) = E(\hat{\theta}) - \theta_0$. Here, N is number of observations.
- Unbiased estimator $\Rightarrow b_\theta = 0\ \forall N$ and θ_0.

Statistical basis of estimation: Homework

- If we have N observations of a quantity X, then the arithmetic mean of the observations is an unbiased estimator of the mean ($\mu_0 = E(X)$) of the distribution.
- $\hat{\mu}_N(\text{estimator}) = \frac{X_1+X_2 \ldots X_N}{N}$
- $b_N(\hat{\mu}) = E(\hat{\mu}) - \mu_0 = \frac{\sum_{i=1}^{N} E(X_i)}{N} - \mu_0 = \frac{\sum_{i=1}^{N} \mu_0}{N} - \mu_0 = 0$
- `Homework`: Show that the sample variance $\hat{\sigma}^2 = \frac{\sum_{i=1}^{N}(X_i - \hat{\mu})^2}{N}$ is not an unbiased estimator of the variance σ_0^2 of the distribution.

Ordinary least square estimator (OLS)

- Let random variates X be distributed according to the PDF $Y = F(X, \theta)$, with n unknown parameters θ.
- If we have N observations of the form $(X_i, Y_i, \sigma(Y_i))$
- One can write an OLS as: $\chi^2 = \sum_{i=1}^{N} \frac{(F(X_i,\theta)-Y_i)^2}{\sigma(Y_i)^2}$
- The estimates $(\hat{\theta})$ of the unknown parameters (θ) are determined by solving n equations of the form $\frac{\partial \chi^2}{\partial \theta_i} = 0$.
- If $F = \theta_1 \times X + \theta_2$, $\chi^2 = \sum_{i=1}^{N} \frac{(\theta_1 \times X + \theta_2 - Y_i)^2}{\sigma(Y_i)^2}$
- We get two equations to solve:
  $\frac{\partial \chi^2}{\partial \theta_1} = \sum_{i=1}^{N} \frac{2\times(\theta_1 \times X_i + \theta_2 - Y_i)\times X_i}{\sigma(Y_i)^2} = 0$ and
  $\frac{\partial \chi^2}{\partial \theta_2} = \sum_{i=1}^{N} \frac{2\times(\theta_1 \times X_i + \theta_2 - Y_i)}{\sigma(Y_i)^2} = 0$

Ordinary least square estimator (OLS)

- The covariance matrix is defined as: $U^{-1} = \frac{1}{2}\frac{\partial^2 \chi^2}{\partial\theta_i \partial\theta_j}|_{\theta=\hat{\theta}}$
- $U = \begin{bmatrix} \sigma(\theta_1)^2 & \rho\sigma(\theta_1)\sigma(\theta_2) \\ \rho\sigma(\theta_1)\sigma(\theta_2) & \sigma(\theta_2)^2 \end{bmatrix}$.
- ρ is the correlation between θ_1 and θ_2.

Maximum likelihood estimator (MLE)

- Let random variates X be distributed according to the PDF $Y = F(X, \theta)$, with n unknown parameters θ.
- If we have N observations X_i.
- The likelihood is written as: $L = \prod_{i=1}^{N} F(X_i, \theta)$
- MLE: The parameters θ which maximise the likelihood.
- Hence, we solve n equations $\partial L / \partial \theta_i = 0$
- In many of the cases, for numerical simplicity on takes minimizes the negative log of likelihood (NLL), $\partial(-Log(L))/\partial\theta_i$
- The inverse covariance matrix is given by: $U^{-1} = -\frac{\partial^2 Log(L)}{\partial\theta_i \partial\theta_j}|_{\theta=\hat{\theta}}$
- `Homework`: Show that if F is gaussian, minimizing NLL is equivalent to the OLS method.

RooFit: Basics

Mathematical concept	RooFit class
Variable x	`RooRealVar`
Function $f(x)$	`RooAbsReal`
Pdf $p(x)$	`RooAbsPdf`
Space point $\overrightarrow{x}$	`RooArgSet`
Integral $\int_{x_{min}}^{x^{max}} f(x)dx$	`RooRealIntegral`
List of space points	`RooAbsData`

Operation	RooFit class
Addition	`RooAddPdf / RooAddition` for functions
Product	`RooProdPdf / RooProduct` for functions
Convolution	`RooFFTConvPdf`
PDF or function from histogram	`RooHistPdf / RooHistFunc`
Kernel estimation	`RooKeysPdf`
Morphing PDFs for sys. variations	`RooMomentMorph / RooMomentMorphFunc`

RooFit: Example

```
RooRealVar x("x","x",-10,10) ;
RooRealVar mean("mean","mean of gaussian",
                1,-10,10) ;
RooRealVar sigma("sigma","width of gaussian",
                 1,0.1,10) ;
```

1. define 3 variables:
   - *observable* x
   - free *parameters* mean, sigma

```
RooGaussian gauss("gauss","gaussian PDF",
                  x,mean,sigma) ;
```

2. create *PDF* model with these variables

```
RooDataSet* data = gauss.generate(x,10000) ;
```

3. generate 10^4 toy events

```
gauss.fitTo(*data) ;
```

4. fit PDF and all floating parameters to data

```
RooPlot* xframe = x.frame() ;
gauss.plotOn(xframe) ;
data->plotOn(xframe) ;
xframe->Draw() ;
```

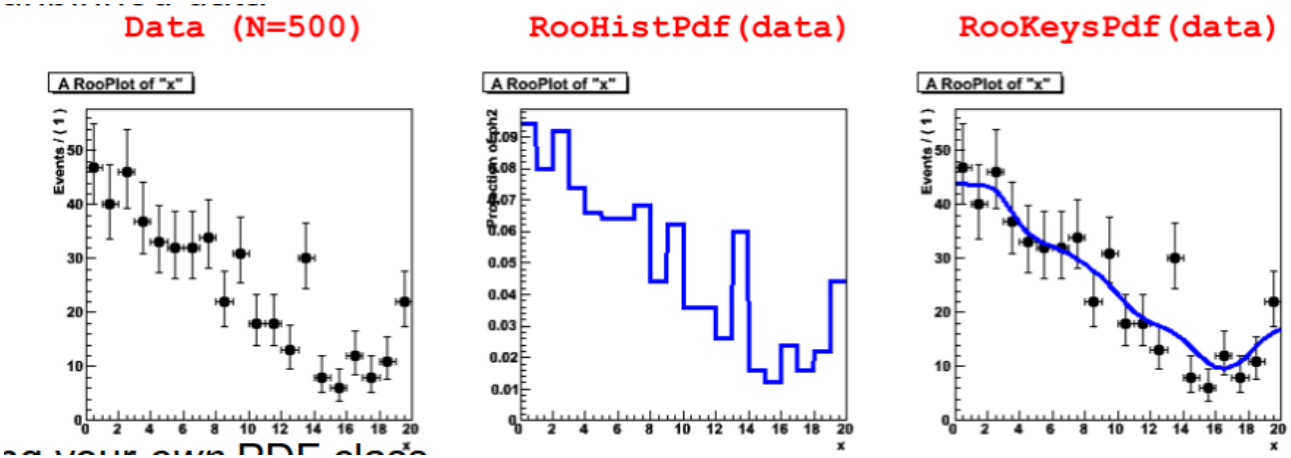
5. plot data and PDF

RooFit: PDF

- RooGaussian: Gaussian
- RooBifurGauss: Different width on left and right side of mean
- RooExponential: Exponential decay
- RooPolynomial: Standard Polynomials
- RooChebychev: Chebychev polynomials (recommended because of higher fit stability due to little correlation)
- RooPoisson: Poisson distribution

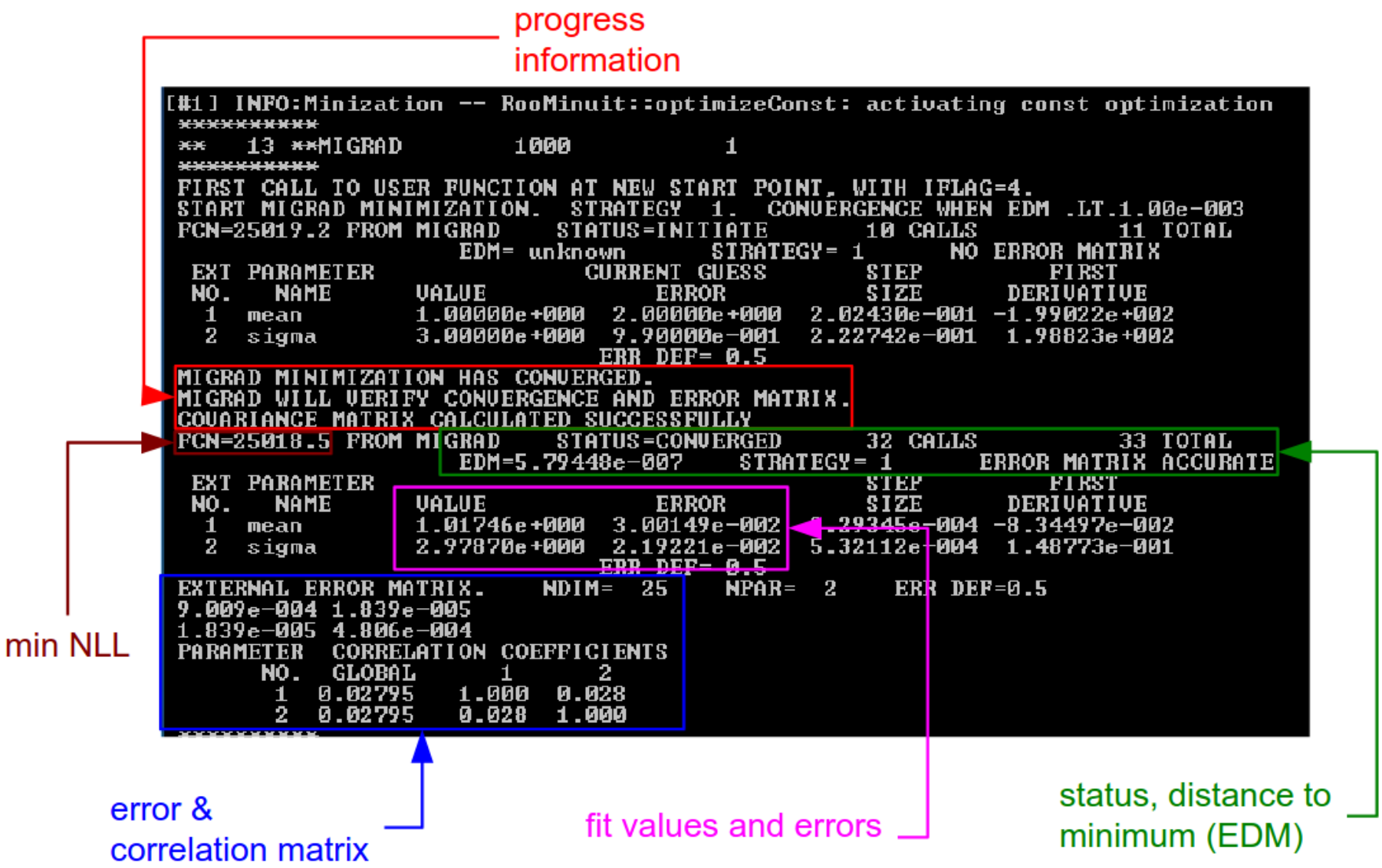
RooFit: PDF

- RooHistPdf: Created from an external ROOT histogram, optional interpolation for smoothing
- RooKeysPdf: Kernel estimation, superposition of Gaussians on external unbinned data



- Writing your own PDF class from a formula expression: RooGenericPdf gp("gp","Generic PDF","exp(x*y+a)-b*x", RooArgSet(x,y,a,b)) ;
- RooClassFactory to write and compile own C++ code for PDFs

RooFit: Output

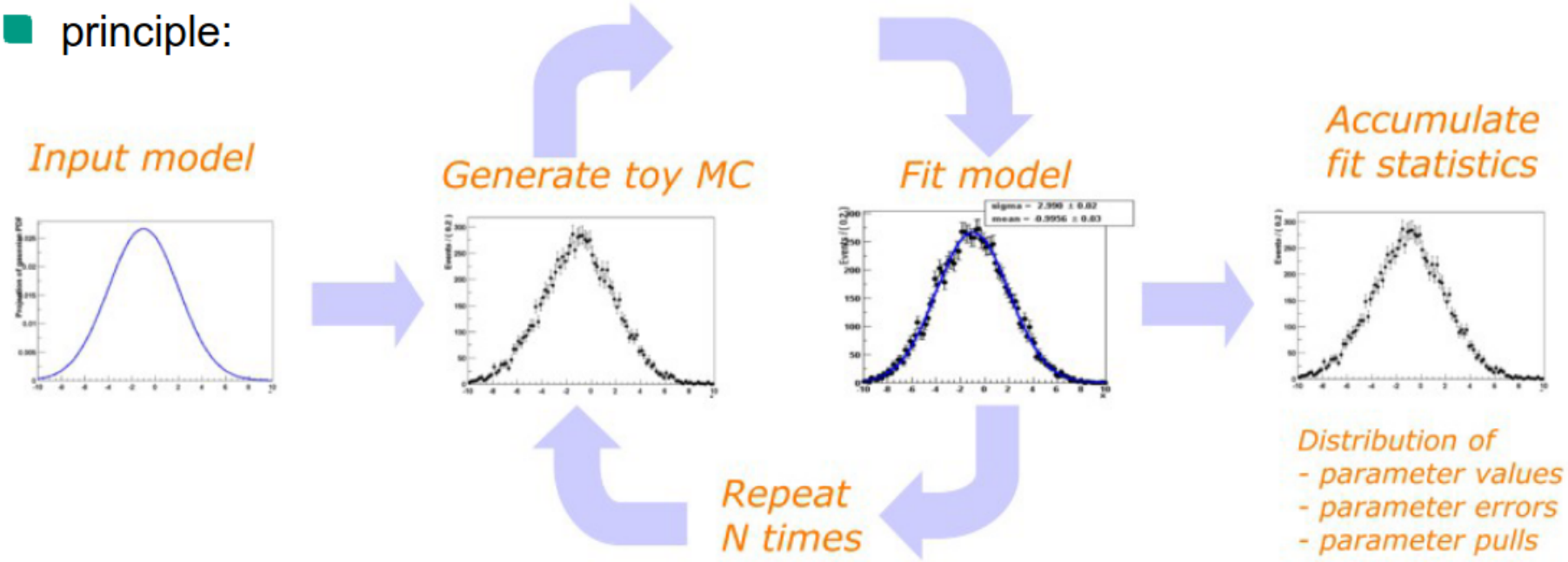


RooFit: Tasks

- Simple_Fit.C : A simple macro which declares a gaussian PDF for Mbc variable, generates the data using the gaussian PDF and performs fit.
- Run MultiDim.C : Macro to fit a 2D PDF: $\mathcal{P}(M_{bc}, \Delta E) = \mathcal{G}(M_{bc}) \times \mathcal{G}(\Delta E)$
- Run Composite.C : A macro to perform a two component composite fit: $\mathcal{P}(\text{Total}) = \text{nsig} \times \mathcal{P}(\text{nsig}) + \text{nbkg} \times \mathcal{P}(\text{nbkg})$
- Run Composite_2.C: Some plot decoration

RooFit: Toy ensemble studies

- principle:



- Create pull and residual:
- Residual $= \theta^{\text{Fit}} - \theta^{\text{True/Generated}}$ (Centered at zero)
- Pull $= \frac{\theta^{\text{Fit}} - \theta^{\text{True/Generated}}}{\sigma_\theta}$ (Normal distribution)

RooFit: Toy ensemble studies

- Instantiate MC study manager:
  RooMCStudy mgr(inputModel) ;
- Generate and fit 100 samples of 1000 events:
  mgr.generateAndFit(100,1000) ;
- Plot pull for parameter (param):
  RooPlot* xframe = mcstudy→plotPull(param, -5.0, 5.0, 50, kTRUE) ;

`Task`:

- Run Fit_Stability.C : A macro to perform fit stability test for the composite model which we defined earlier.

Home work

- There's a macro H_W.c which performs a fit to an input file
- Run the macro and check the fit output
- `Task`: Change the fit function so that the fit result is good.
- `Task`: There are three files for unknown signal and background. Find out the fit model for signal and background PDF.
- `Task`: Perform a composite fit on the sig+bkg (total) file to find out number of signal events. While doing the composite fit, fix the parameters of the signal component to the one's obtained in the previous step. Keep parameters of background floated.

Summary

- We took a quick look at the statistical basics.
- A brief introduction of Roofit
- Some examples, few homework problems.

Questions