RooFit Exercises

Getting Started

- Assume knowledge of ROOT
  - how to write and run a ROOT macro
  - root>.x myMacro.C (interpreted)
  - root>.x myMacro.C+ (compiled with ACLIC but require providing all include files)
- some RooFit classes are in a namespace
- recommended to add at beginning of macro:
  - using namespace RooFit
    - this will also load automatically the RooFit libraries
- note that RooFit methods start with lower case letter

Getting Started

- RooFit class documentation (from code)
  - http://root.cern.ch/root/html/ROOFIT_ROOFITCORE_Index.html
  - http://root.cern.ch/root/html/ROOFIT_ROOFIT_Index.html
- RooFit home page at ROOT web site
  - http://root.cern.ch/drupal/content/roofit
- Has links to manual and tutorial macros

RooFit Exercises (1)

- Exercise 1
  - Make a simple exponential unbinned fit
    - make exponential model (possibly using the factory)
      - N.B. RooExponential is : pdf(x) = 1/a e^{ax}
    - generate random data set (e.g. N = 1000)
      - using pdf.generate(...) (→ slide 11)
    - fit the data using pdf.fitTo(..) (→ slide 13)
    - plot data set and resulting fit function
      - use also pdf->paramOn(plot) to display fit parameters
    - re-parameterize exponential using tau
      - pdf(x) = $-\tau\ e^{-x/\tau}$

Exercise (2)

- Exercise 2: create a simple gaussian model
  - generate random data set (e.g. N=100)
    - use workspace factory, if not create a workspace to import the model (→ slide 15)
    - import also the data set in workspace
  - save the model in a ROOT file using
    - workspace.writeToFile("fileName.root",true)
  - save also the ROOT macro making the model (will be needed later for RooStats exercises)
  - make another macro to read the workspace from the file (using TFile::Get("workspaceName"))
    - get pdf and data from workspace
    - fit the data and look at fit result
      - to get fit result: r = pdf->fitTo(*data, "Save()")
      - print fit result (r->Print())

RooFit Exercises (3)

- Exercise 3: (extended pdf's)
  - modify exponential model (or gaussian) to make an extended pdf
    - introduce number of expected events, Nexp
      - w.factory("Exponential::expo(x[0,10],a[-20,0])");
      - w.factory("ExtendPdf::(expo, Nexp[1000,0,10000])");
  - generate dataset
  - fit the model
  - note that now we have an extra fit parameter: the number of expected events

Solutions (1)

- Exercise 1:
  - exponentialFit.C
  - exponentialFit_tau.C
- Exercise 2:
  - GaussianModel.C
  - fitModel.C
- Exercise 3:
  - ExtendedGaussianModel.C

RooFit Exercises (4)

- Exercise 4: (composite model)
  - make an exponential + gaussian model using w.factory("SUM::pdf(fs*gaus, expo)") (→ slide 32)
    - do first non-extended model (0 < fs < 1)
  - generate data (e.g. N=1000), fit the data and plot results (use RooFit::Components() to specify components to plot) (→ slide 31)
  - make extended model using Ns and Nb (Ns = fs*N)
    - w.factory("SUM::pdf(Ns*gaus, Nb*expo)") (→ slide 30)
    - check difference in error in Ns obtained between extended and not-extended fit
  - save the macro making the extended model and also save the workspace in a file (we will be using it later in a RooStats exercise)

RooFit: Exercises (5)

- Modify generation of exponential + gaussian model
  - use generateBinned to generate a binned data set (→ slide 11)
    - N.B: result is now a RooDataHist object instead of a RooDataSet
- fit now model (a binned Poisson likelihood fit will be performed)
  - check Ns fit result (a small bias is now present, result is smaller than in the unbinned fit)

RooFit: Exercises (5)

- Option:
  - do now a fit of the model using directly ROOT
  - use pdf->asTF to make transform RooAbsPdf in a ROOT TF1 object
    - N.B. need to multiply pdf by number of events and by the bin width
- use data->createHistogram to transform in a TH1
- use h1::Fit(f1,"LI")
  - performs a fit but will integrate bin content
    - bias from binned fit is reduced

RooFit: Exercise 6

- Convolution
  - run fftdemo.C macro and look at it to show how convolution is done between a Landau pdf and a Gaussian resolution model
  - A binned likelihood fit of the numerically convoluted pdf with three floating parameters takes ~1 second

Solutions (2)

- Exercise 4:
  - SPlusBExpoFit_notExtended.C
  - SPlusBExpoFit.C
- Exercise 5:
  - SPlusBExpoFit_Binned.C
- Exercise 6:
  - fftdemo.C

n-Dim models and likelihood ratio plot

- run llrdemo.C
- This macro builds a 3-dimensional model
  - Flat background in (x,y,z)
  - Gaussian signal in (x,y,z) with correlations
- It plots three 2D projections (x,y), (x,z) and (y,z)
- Then it makes three varieties of 1D plots of model and data
  - Plain projection on x (shows lots of background)
  - Projection on x in a 'signal box' in (y,z)
  - Projection on x with a cut on the LR(y,z)>68%, where LR(y,z) is defined as

$$LR(y,z) = \frac{\int f_{sig} \times S(x,y,z)dx}{\int S(x,y,z) + B(x,y,z)dx}$$

(i.e. the signal probability according to the model using the (y,z) observables only)