

Table of Contents

CMS.RooFit Examples in CMSSW.....	1
Introduction.....	1
Building RooFit applications in CMSSW.....	1
Simple Maximum Likelihood fit with Toy Monte Carlo generation.....	1
Code fragments.....	1
Result.....	2
Complete example.....	2
Simple Chi-squared fit.....	2
Results.....	3
Complete example.....	3
Multi-dimensional fits.....	3
Complete example.....	4
More examples.....	4

CMS.RooFit Examples in CMSSW

Complete:

Introduction

This page is intended to provide examples with CMS.RooFit that are usable within CMSSW. The presented examples can be installed and compiled under CMSSW using the following recipe:

```
cmsrel CMSSW_3_1_0_pre7
cd CMSSW_3_1_0_pre7/src
cmsenv
addpkg PhysicsTools/CandExamples V02-00-00
cd PhysicsTools/CandExamples
scram b
$CMSSW_BASE/test/slc4_ia32_gcc345/<name_of_the_application>
```

Building RooFit applications in CMSSW

Before using the following examples, you should first understand how to build a RooFit application under CMSSW. This part is described in the following page:

- Building RooFit applications in CMSSW

Simple Maximum Likelihood fit with Toy Monte Carlo generation

Generate a Toy Monte Carlo sample according to a PDF modeled as Exponential plus a Gaussian, and perform an Extended Maximum Likelihood fit according to the same model.

Code fragments

Define the variable (x) and the parameters of interest (μ , σ , λ) with their validity range:

```
RooRealVar x("x", "x", -10, 10);

RooRealVar mu("mu", "average", 0, -1, 1);
RooRealVar sigma("sigma", "sigma", 1, 0, 5);
RooGaussian gauss("gauss", "gaussian PDF", x, mu, sigma);
RooRealVar lambda("lambda", "slope", -0.1, -5., 0.);
```

Define a Gaussian and an Exponential PDF according to the above parameters:

```
RooGaussian gauss("gauss", "gaussian PDF", x, mu, sigma);
RooExponential expo("expo", "exponential PDF", x, lambda);
```

Define a linear combination of the two PDFs with yields s (signal) and b (background):

```
RooRealVar s("s", "signal yield", 100, 0, 10000);
RooRealVar b("b", "background yield", 100, 0, 10000);

RooAddPdf sum("sum", "gaussian plus exponential PDF",
              RooArgList(gauss, expo), RooArgList(s, b));
```

Generate a mixed signal plus background sample with 2000 entries:

```
RootDataSet * data = sum.generate(x, 2000);
```

Fit the sample according to the PDF model using Extended Unbinned Maximum Likelihood method:

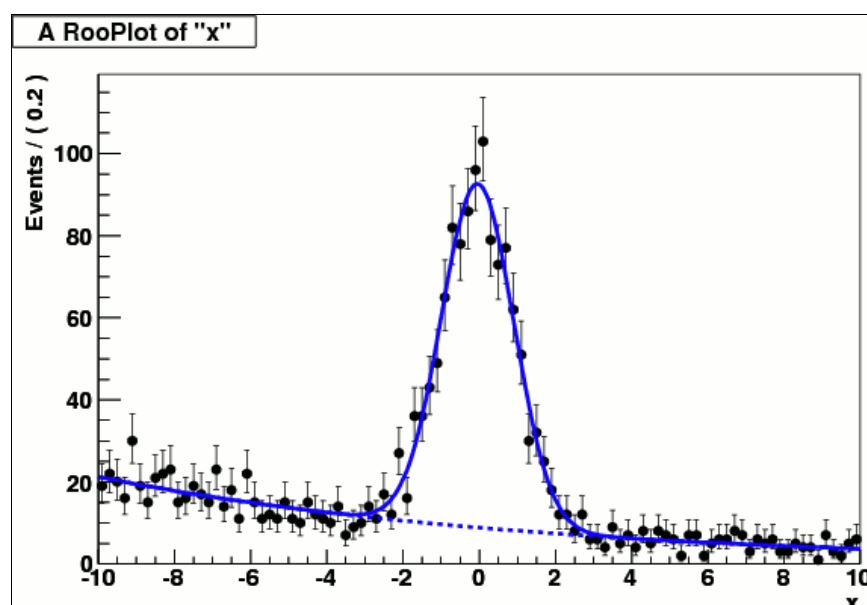
```
sum.fitTo(*data, CMS.RooFit::Extended());
```

Plot the fit result and save the output as EPS file:

```
RooPlot * xFrame = x.frame() ;  
data->plotOn(xFrame) ;  
sum.plotOn(xFrame) ;  
sum.plotOn(xFrame, CMS.RooFit::Components(expo), CMS.RooFit::LineStyle(kDashed)) ;  
TCanvas c;  
xFrame->Draw();  
c.SaveAs("extendedLikelihoodFit.eps");
```

Result

Below the plot resulting from running the above example



Complete example

• [PhysicsTools/CandExamples/test/extendedLikelihoodFit.cpp](#)

Simple Chi-squared fit

Using the same PDF defined in the previous example, we can also adopt a binned chi-squared fit.

In order to define the binned histogram, one has to use the `RooDataHist` class, setting the proper binning for the `x` variable:

```
x.setBins(50);  
RooDataHist hist("hist", "hist", RooArgSet(x), *data);
```

Then, the chi-squared fit should be used instead of the likelihood fit. Notice, that if the yields should be fitted, an extra parameter (similar to the "extended" option in the likelihood fit) should be set to true in `RooChi2Var`:

```
RooChi2Var chi2("chi2", "chi2", sum, hist, true);
```

```

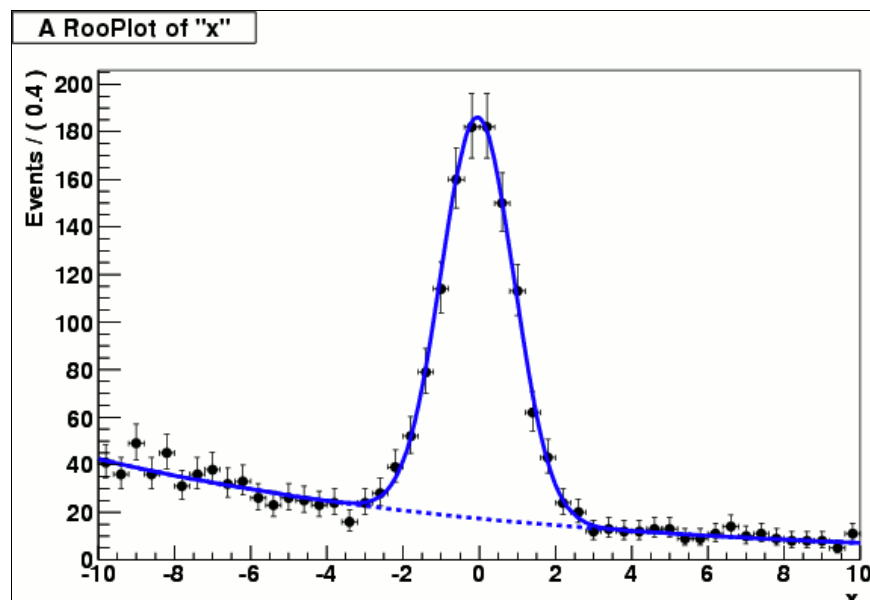
RooMinuit minuit(chi2);
minuit.migrad();
minuit.hesse();

```

All the rest of the code remains unchanged.

Results

Below the plot resulting from running the above example



Complete example

- [PhysicsTools/CandExamples/test/binnedChi2Fit.cpp](#)

Multi-dimensional fits

RootFit does not distinguish, conceptually, multi-dimensional PDFs from one-dimensional ones. The only difference is that the support for plotting via `RooPlot` is limited to a single dimension, and in the case of multi-dimensional PDFs works with one-dimensional projections.

The following example shows how to define a two-dimensional PDF using `RooProdPdf` as the product of independent one-dimensional PDFs, and how to project PDFs and data sets to a single dimension using `RooPlot`, or as two-dimensional standard ROOT contour plots:

```

RooRealVar x("x", "x", -10, 10);
RooRealVar y("y", "y", -10, 10);
RooRealVar mux("mux", "average-x", 0, -1, 1);
RooRealVar sigmax("sigmax", "sigma-x", 0.5, 0, 5);
RooGaussian gaussx("gaussx", "gaussian PDF x", x, mux, sigmax);
RooRealVar muy("muy", "average-y", 0, -1, 1);
RooRealVar sigmay("sigmay", "sigma-y", 1.5, 0, 5);
RooGaussian gaussy("gaussy", "gaussian PDF y", y, muy, sigmay);
RooProdPdf gaussyx("gaussyx", "gaussyx", RooArgSet(gaussx, gaussy));

RooRealVar lambdax("lambdax", "log-slope x", -0.1, -5., 0.);
RooExponential expox("expox", "exponential PDF x", x, lambdax);
RooRealVar lambday("lambday", "log-slope y", -0.1, -5., 0.);
RooExponential expoy("expoy", "exponential PDF y", y, lambday);
RooProdPdf expoxy("expoxy", "expoxy", RooArgSet(expox, expoy));

```

```

RooRealVar s("s", "signal yield", 50, 0, 10000);
RooRealVar b("b", "background yield", 200, 0, 10000);

RooAddPdf sum("sum", "gaussian plus exponential PDF",
              RooArgList(gaussxy, expoxy), RooArgList(s, b));
RooDataSet * data = sum.generate(RooArgSet(x,y), 250);
RooFitResult* result = sum.fitTo(*data, RooFit::Extended(), RooFit::Minos(), RooFit::Save());

result->Print();

RooPlot * xFrame = x.frame() ;
data->plotOn(xFrame) ;
sum.plotOn(xFrame) ;
sum.plotOn(xFrame, CMS.RooFit::Components(expoxy), CMS.RooFit::LineStyle(kDashed)) ;

RooPlot * yFrame = y.frame() ;
data->plotOn(yFrame) ;
sum.plotOn(yFrame) ;
sum.plotOn(yFrame, CMS.RooFit::Components(expoxy), CMS.RooFit::LineStyle(kDashed)) ;

TCanvas c;
xFrame->Draw();
c.SaveAs("twoDimensionalFit-yProjection.eps");
yFrame->Draw();
c.SaveAs("twoDimensionalFit-xProjection.eps");
TH2 * surf = dynamic_cast<TH2*>(sum.createHistogram("surf", x,
                                                CMS.RooFit::Binning(100),
                                                CMS.RooFit::YVar(y, CMS.RooFit::Binning(100)))

surf->Draw("contz");
c.SaveAs("twoDimensionalFit-xyContour.eps");
TH2 * surfData = data->createHistogram(x, y, 100, 100, "", "surfData");
surfData->Draw("contz");
c.SaveAs("twoDimensionalFit-xyDataContour.eps");

```

Complete example

- [PhysicsTools/CandExamples/test/twoDimensionalFit.cpp](#)

More examples

You can find more examples in the following pages:

- [CMS.RooFit tutorial](#)