$Z \to e^+e^-$ Data Analysis Tutorial

Javier Duarte, Cristian Pena, Valere Lambert

May 26, 2015

Contents

1 Introduction

You should receive a skeleton `Zee.C` file and an input `treeArray.root` file. You will need to re-install root according to the revised directions given here: `https://docs.google.com/document/d/1ss5EbpSYidwja7AHelUwNc3VRPqAjJ6cJ3jvDSl1244/edit`

Take some time to aquaint yourself with the code. It initializes a variable `mee` which represents the invariant mass of an electron and a positron in the collision event. Then it loads data from the `treeArray.root` file. Finally it plots the data and prints the resulting graph in a `Zee.pdf` file.

Running the code,

```
root -l Zee.C
```

creates an output `Zee.pdf` file, shown in Fig. 1, which illustrates a histogram of the invariant mass spectrum for an electron positron pair in *real* CMS data taken at center-of-mass energy of 8 TeV in 2012.

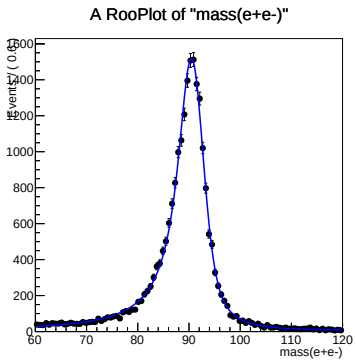

Figure 1: Invariant mass spectrum

2 Maximum Likelihood Fits

In this section, you will familiarize yourself with the practical aspects of performing a maximum likelihood fit to data using RooFit in the ROOT framework.

2.1 Fit with a Gaussian Distribution

Now, add the following code to the file

```
RooRealVar mean("mean","mean",90,60,120) ;
RooRealVar sigma("sigma","sigma",5,0,20);
RooGaussian gaus("gauss","gaus",mee,mean,sigma);
```

after the initialization of the variable `mee`.

Add the code

```
RooFitResult *fr = gaus.fitTo(*data,Save());
fr->Print("v");
```

to perform a fit to the data with a Gaussian distribution and print out the result (in the terminal). Finally, add the code

```
gaus.plotOn(frame);
```

after the line reading `data->plotOn(frame);` executing the file,

```
root -l Zee.C
```

you should now have a modified `Zee.pdf` that resembles Fig. 2. The Gaussian distribution is often a good first-approximation when fitting data that has a well defined mode (peak) and spread (width). Its functional form is parameterized by these two numbers: μ (mean, mode, median) and σ (standard deviation). Ignoring an overall normalization constant,

$$f(x) = e^{-(x-\mu)^2/2\sigma^2} \tag{1}$$

Notice, however, that in this case the fit to the data is not very good.

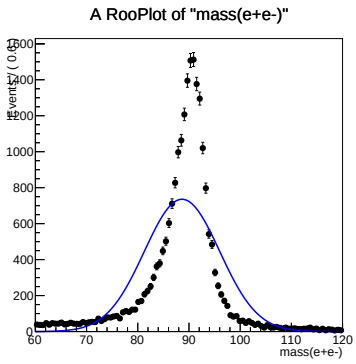

Figure 2: Invariant mass spectrum fit with a Gaussian distribution.

2.2 Fit with a Relativistic Breit-Wigner Distribution

Now, instead of adding the code from the previous section (i.e. you should comment out the code from the previous section), add the following code to the file

```
RooRealVar bwMean("mZ","BW Mean", 90, 60, 120);
RooRealVar bwWidth("GammaZee", "BW Width", 2, 0, 10);
RooBreitWigner bw("bw", "bw", mee, bwMean, bwWidth);
```

after the initialization of the variable `mee`.

Add the code

```
RooFitResult *fr = bw.fitTo(*data,Save());
fr->Print("v");
```

to perform the fit and print out the result (in the terminal).

Finally, add the code

```
bw.plotOn(frame);
```

after the line reading `data->plotOn(frame);`. After executing the file,

```
root -l Zee.C
```

you should now have a modified `Zee.pdf` that resembles Fig. 3.

This code is intended to fit the mass spectrum with a Relativistic Breit-Wigner distribution, which you can read more about on Wikipedia or in particle physics textbooks. Its functional form is (ignoring an overall normalization constant)

$$f(x) = \frac{1}{(x-\mu)^2 + \frac{1}{4}\Gamma^2} \tag{2}$$

Notice the fit is much improved with respect to the last iteration.

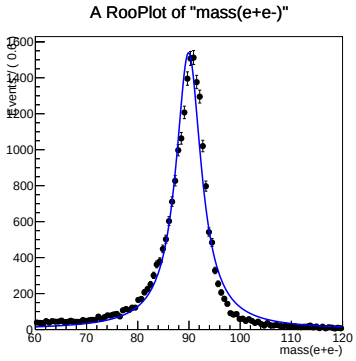

Figure 3: Invariant mass spectrum fit with a Relativistic Breit-Wigner distribution.

2.3 Fit with a Relativistic Breit-Wigner Distribution Convoluted with a Crystal Ball Distribution

Now add the following code instead of the code from the previous sections.

```
RooRealVar cbBias ("cbBias", "CB Bias", -.5, -10, 10);
RooRealVar cbSigma("cbSigma", "CB Width", 5.7, 0.02, 10.0);
RooRealVar cbCut  ("cbCut","CB Cut",1.05, 0.1, 3.0);
RooRealVar cbPower("cbPower","CB Order", 2.45, 0.1, 20.0);
RooRealVar bwMean("mZ","BW Mean", 91.1876, 80, 100);
RooRealVar bwWidth("GammaZee", "BW Width", 2.4952, 0, 10);
RooBreitWigner bw("bw", "bw", mee, bwMean, bwWidth);
RooCBShape  cball("cball", "Crystal Ball", mee, cbBias, cbSigma, cbCut, cbPower);
RooFFTConvPdf model("model", "bw X crystal ball", mee, bw, cball);
```

after the initialization of the variable `mee`.

Add the code

```
RooFitResult *fr = model.fitTo(*data,Save());
fr->Print("v");
```

to perform the fit and print out the result (in the terminal).

Finally, add the code

```
model.plotOn(frame);
```

after the line reading `data->plotOn(frame);`

You should now have a modified `Zee.pdf` that resembles Fig. 4.

This code is intended to fit the mass spectrum with a Relativistic Breit-Wigner distribution convoluted with a Crystal Ball distribution, which you can read more about on Wikipedia. The Crystal Ball distribution is essentially a

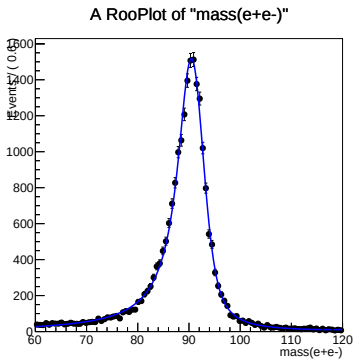

Figure 4: Invariant mass spectrum fit with a Relativistic Breit-Wigner distribution convoluted with a Crystal Ball distribution.

Gaussian distirbution with a longer tail on one side,

$$f(x) = \begin{cases} (\frac{n}{|\alpha|})^n e^{-\alpha^2/2} (\frac{n}{|\alpha|} - |\alpha| - x)^{-n}, x < -|\alpha| \\ e^{-(x-\mu)^2/2\sigma^2}, x \geq -|\alpha| \end{cases} \tag{3}$$

Notice the fit is even better with respect to the last iteration. Why is this so? Why did this homework (especially the final section with the convolution) require the installation of a fast fourier transform (FFT) library?

3 Further Discussion and Analysis

The endpoint of an analysis is usually not performing a maximum likelihood fit, but instead interpretting the output: the best-fit parameter values (sometimes called the maximum likelihood estimators) and their uncertainties.

3.1 Retreiving the ML fit result

From the text output of the code, you are given a series of numbers representing the best-fit values and uncertainties for each paramter. To complete the exercise you will need to find the best-fit values for Γ_Z, the total width of the Z boson, and m_Z, the mass of the Z boson. In the following, the best fit values for these have been removed (so you can find and use your own numbers).

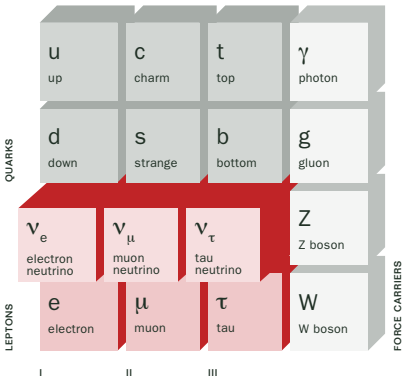

Figure 5: Particles in the standard model of particle physics.

```
RooFitResult: minimized FCN value: 71901, estimated distance to minimum: 7.7855e-07
              covariance matrix quality: Full, accurate covariance matrix
              Status : MIGRAD=0 HESSE=0

  Floating Parameter  InitialValue    FinalValue +/-  Error     GblCorr.
--------------------  ------------  --------------------------  --------
           GammaZee    2.4952e+00              +/-             <none>
             cbBias   -5.0000e-01    1.6158e+00 +/-  5.68e-01  <none>
              cbCut    1.0500e+00    1.1297e+00 +/-  6.51e-02  <none>
            cbPower    2.4500e+00    1.6536e+00 +/-  1.04e-01  <none>
            cbSigma    5.7000e+00    1.6482e+00 +/-  8.28e-02  <none>
                 mZ    9.1188e+01              +/-              <none>
```

3.2 Determining the Number of Light Neutrinos

The standard model, whose particle content is depicted in Fig. 5, contains three families of light neutrinos: ν_e, ν_μ, and ν_τ. The Z boson has many observable decay modes including

- $Z \to e^+e^-$
- $Z \to \mu^+\mu^-$
- $Z \to \tau^+\tau^-$
- $Z \to u\bar{u}$
- $Z \to d\bar{d}$

- $Z \to c\bar{c}$
- $Z \to s\bar{s}$
- $Z \to b\bar{b}$

The Z boson also has several *invisible* decay modes. In the standard model, the invisible decay modes are

- $Z \to \nu_e \bar{\nu}_e$
- $Z \to \nu_\mu \bar{\nu}_\mu$
- $Z \to \nu_\tau \bar{\nu}_\tau$

one for each flavor of neutrino.

Section under construction...