Probabilistic Background Suppression Method (Q-Factor)

1 December 2015 | Michael C. Kunkel | IKP-1

The Problem

Common to have a signal mixed with background

- Common method to handle background is side-band subtraction
  - Problematic if kinematics of signal is different from background
  - Problematic if signal is a function of several variables
    - Binning such as in a Dalitz plot

A Solution http://arxiv.org/abs/0809.2548

Probabilistic Weighting

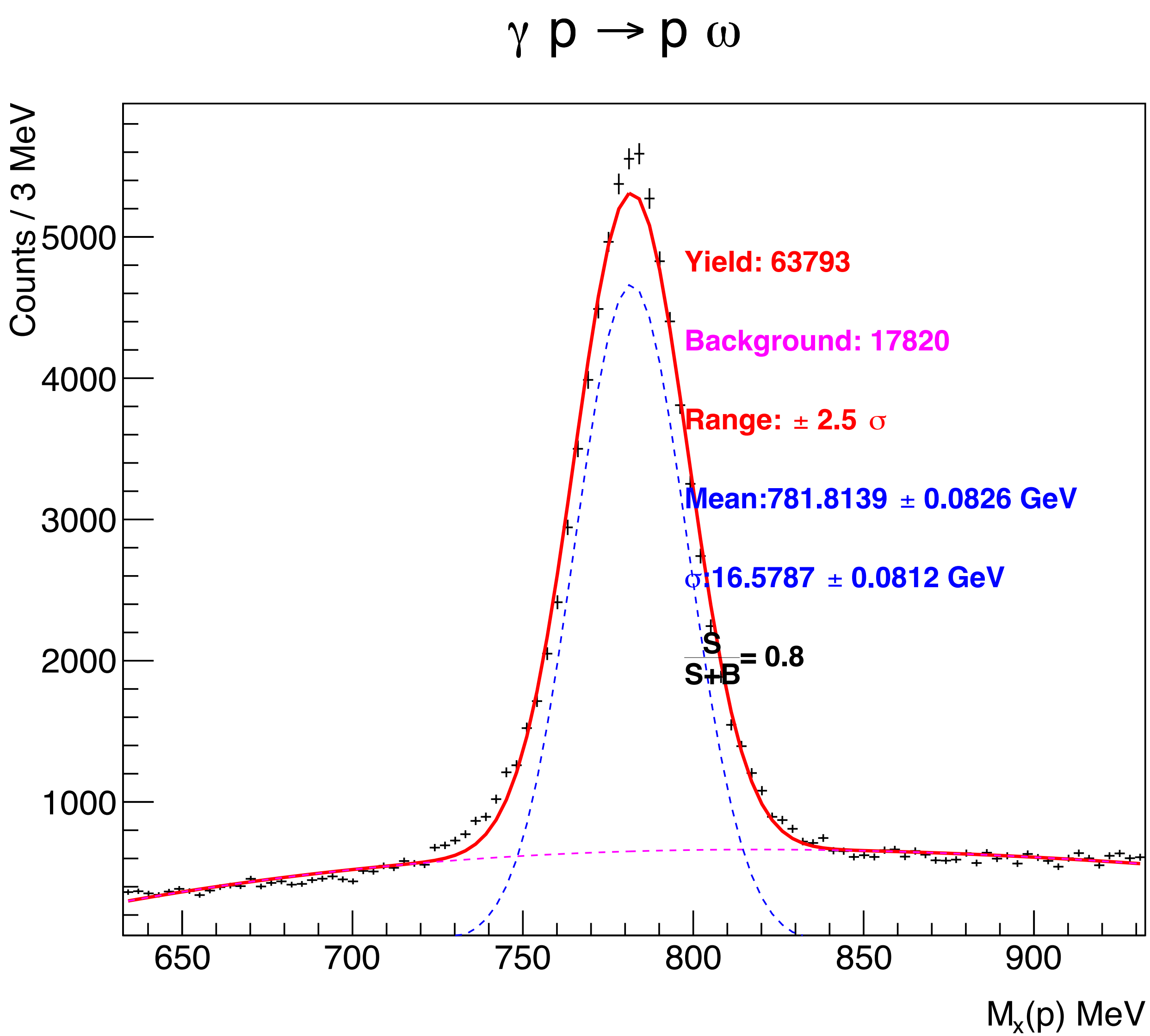
- Data set is comprised of *n* events where e_i can be described by *m* coordinates, ε_j, where *m* is at minimum 2
  - $e_i(\varepsilon_j)$ (i = 1, n), (j=1,m)
    - coordinates can be angles, massed, energies...etc.
- Data is made of *signal* $S(\varepsilon_j)$ and *background* $B(\varepsilon_j)$
  - apriori knowledge of signal + background shape for one coordinate ε_r, reference coordinate
- Goal of procedure is to identify the chance, for a given event e_i, that the event is a signal event Q, or a background event (1-Q).
  - Use of distance measure (normalized Euclidean distance)
    - $$d_{ij} = \sum_{k \neq 2}^{m} \left[\frac{(\xi_k)_i - (\xi_k)_j}{r_k} \right]^2$$
    - r_k is the maximum distance in any pair of events

A Solution

Probabilistic Weighting

- each event, e_i, compute d_{ij} for all events in the data set
    - retain N_d closest to e_i
- Using N_d, fit the distribution composed of $(\varepsilon_r)_j$, reference variables to the known signal and background
    - Use unbinned likelihood method to avoid binning issues
        - ROOSTATS and ROOFIT
    - signal $s_i = f_s((\varepsilon_r)_i,\eta)$. Where $f_s(\varepsilon_r)$ is the function describing the signal with η fit paramters
    - background $b_i = f_b((\varepsilon_r)_i,\eta)$. $f_b(\varepsilon_r)$ is the function describing the background with η fit paramters
    - $$Q_i = \frac{f_s((\xi_r),\eta)_i}{f_s((\xi_r),\eta)_i + f_b((\xi_r),\eta)_i} = \frac{s_i}{s_i + b_i}$$

Example



γ p → p ω
Counts / 3 MeV
5000
4000
3000
2000
1000
650
700
750
800
850
900
Mx(p) MeV
Yield: 63793
Background: 17820
Range: ± 2.5 σ
Mean:781.8139 ± 0.0826 GeV
σ:16.5787 ± 0.0812 GeV
S / S+B = 0.8

Example

Input

- $S(\varepsilon_j)$ = Gaussian signal
- $B(\varepsilon_j)$ = 1st order Chebyshev polynomial

Coordinates with physics information

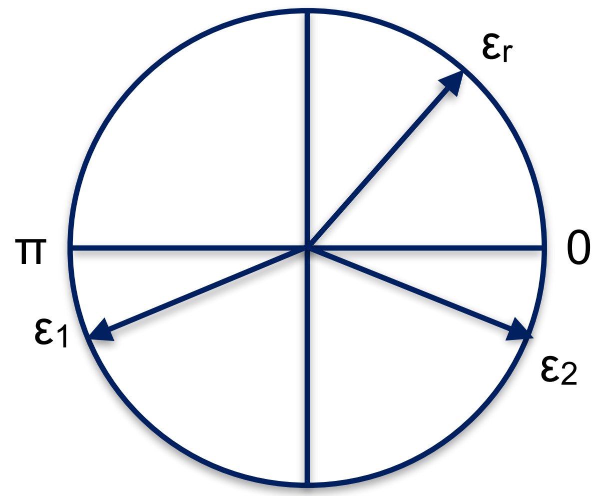
- $e_i(\varepsilon_1)$ = C.M. θ production frame
- $e_i(\varepsilon_2)$ = Beam Energy

- $$d_{ij} = \left[\frac{\cos\theta_i - \cos\theta_j}{2}\right]^2 + \left[\frac{E(\gamma)_i - E(\gamma)_j}{4.5}\right]^2$$

Note: Lab φ has no physics relation, however helicity frame φ does in some physics. Choose coordinate according to actual separable physics quantities

Circular Coordinates

Special concern needs to be addressed if using circular variables i.e. φ



ε_r is closer to ε_2 than ε_1

Must account for circular coordinate metric

Fit Errors

$$\sigma_Q^2 = \sum_{i,j} \frac{\partial Q}{\partial \eta_i} (C_\eta)_{ij}^{-1} \frac{\partial Q}{\partial \eta_i} \, .$$

$$\frac{\partial Q}{\partial \eta_i} = \frac{\partial Q}{\partial s_i} \sum_i \frac{\partial s_i}{\partial \eta_i} + \frac{\partial Q}{\partial b_i} \sum_i \frac{\partial b_i}{\partial \eta_i}$$

$$\sigma_{total}^2 = \sum_{i=1}^{n} (\sigma_{Q_i}^2 + \sigma_{stat}^2)$$

Example Code

```
using namespace RooFit ;


void Fitted_Pi0_QFactor()
{

  TChain *chain1 = new TChain("LepTree");
  chain1->Add("g12.root");

  Double_t E_g, mm2_P, CM_Theta;
  chain1->SetBranchAddress("mm2_P", &mm2_P);
  chain1->SetBranchAddress("CM_Theta", &CM_Theta);
  chain1->SetBranchAddress("E_g", &E_g);


  //---------------------------------------Masses of Particles in Mev/
      c^2-----------------------------------------------
  Double_t M_Omega = 782.65;    //pi0 fitted
  Double_t omegaWidth = 8.49;
```

Example Code

```
Int_t nEvent = chain1->GetEntries();

TFile outFile("Omega_QTree.root","recreate");
TTree *t4 = new TTree("QLepTree","QLepTree");

double Qweight_sig, Qweight_bck, QweightError, Egam, cm_Theta, MM2_P;

t4->Branch("Qweight_sig",&Qweight_sig,"Qweight_sig/D");
t4->Branch("Qweight_bck",&Qweight_bck,"Qweight_bck/D");
t4->Branch("QweightError",&QweightError,"QweightError/D");
t4->Branch("Egam",&Egam,"Egam/D");
t4->Branch("cm_Theta",&cm_Theta,"cm_Theta/D");
t4->Branch("MM2_P",&MM2_P,"MM2_P/D");


int bckgrnd_PolOrder = 1;
int N_NearNeighbor = 200;
```

Example Code

```
for(int i=0; i < nEvent; i++)
{
  chain1->GetEntry(i);

  cm_Theta = CM_Theta;
  MM2_P = mm2_P;
  Egam = E_g;

  double dis[nEvent], MM2_P_j[nEvent];

  for (Int_t j=0;j<nEvent;j++) {
    chain1->GetEntry(j);
    Double_t CM_Theta_j = CM_Theta;
    Double_t E_g_j = E_g;

    MM2_P_j[j] = mm2_P;
    dis[j] = pow(0.5*(cm_Theta - CM_Theta_j),2) + pow((Egam - E_g_j)/4.5,2);
  }
```

Example Code

```
Int_t *index = new Int_t[nEvent];
TMath::Sort(nEvent,dis,index,0);// Sorting
Double_t Min_Mass = M_Omega - 200; // Minimum mass in data from Copy_Tree.C
Double_t Max_Mass = M_Omega + 200; // Maximum mass in data from Copy_Tree.C
RooRealVar mass("mass","mass", Min_Mass, Max_Mass);
RooDataSet data("data","data",RooArgSet(mass));

for (Int_t i_NearNeighbor = 0; i_NearNeighbor<N_NearNeighbor; i_NearNeighbor++){
  mass = MM2_P_j[index[i_NearNeighbor]];
  data.add(RooArgSet(mass)) ;
}
delete [] index;

double sumOfWeights = data.sumEntries();

// Define signal function variables
RooRealVar mean("mean","mean", M_Omega);
// start width, minimum width, maximum width
RooRealVar sigma("sigma","sigma", 5, 0.0, 20);
RooGaussian gaussFunction("gauss", "signal", mass, mean, sigma);
```

Example Code

```
//RooRealVar for polynominals
RooRealVar a1("a1", "a1", 0.1, -100.0, 100.0);
RooRealVar a2("a3", "a3", 0.1, -100.0, 100.0);
RooRealVar a3("a3", "a3", 0.1, -100.0, 100.0);

//Checking to see which background order set is define
RooArgSet bkgArgSet = (bckgrnd_PolOrder == 3) ? RooArgSet(a1,a2,a3) :
(bckgrnd_PolOrder == 2) ? RooArgSet(a1,a2) :
((bckgrnd_PolOrder == 1) ? RooArgSet(a1) : RooArgSet());

//Using Chebychev polynomial for better stability
RooChebychev pol("pol","pol",mass,bkgArgSet);
```

Example Code

```
//This is the "f" needed for the pdf
//i.e. f*s/(f*s + (1-f)*b)
RooRealVar signal_back("signal_back","signal/(signal + background)", 0.5, 0, 1);
//Create model
RooAddPdf model("model","pol + gaus", RooArgList(gaussFunction, pol),
                RooArgList(signal_back));
//Perform fit
RooFitResult* r = model.fitTo(data, Save(true), Verbose(false));
//Some options to use in RooFitResult:
//Optimize(1), PrintLevel(-1), PrintEvalErrors(-1)
```

Individual Coding

```
mass.setVal(MM2_P);// At the seed mass
RooArgSet Mass_ArgSet(mass);


double f = signal_back.getVal();
double S_func = gaussFunction.getVal(&Mass_ArgSet);
double s = S_func * f;
double B_func = pol.getVal(&Mass_ArgSet);
double b = B_func * (1. - f);

double qweight = s / (s + b);

Qweight_sig = qweight;
Qweight_bck = (1. - qweight);
```

Calculating Error

Using RooFit composite model framework

$$Q_i = \frac{s_i}{s_i + b_i} = \frac{f s_i}{f s_i + (1 - f) b_i}$$

Error of Q using composite model framework

$$\frac{\partial Q}{\partial \eta_i} = \frac{\partial Q}{\partial s_i} \sum_i \frac{\partial s_i}{\partial \eta_i} + \frac{\partial Q}{\partial b_i} \sum_i \frac{\partial b_i}{\partial \eta_i} + \frac{\partial Q}{\partial f}$$

Example Code

```
//########### Let do error ##################
//Lets Set up the needed derivatives involved in the chain rule

RooAbsReal* inttotal = model.createIntegral(mass);
double S = gaussFunction.getVal(&Mass_ArgSet)*inttotal->getVal(&Mass_ArgSet);
double B = pol.getVal(&Mass_ArgSet)*inttotal->getVal(&Mass_ArgSet);

//will be used to calculate dQ/d(sigma)and dQ/d(s)
double dQ_dS = (1. - f)* B * qweight * qweight/ ( f * S * S );
//will be used to calculate dQ/d(a1 or a2 or a3)
double dQ_dB = -1*(1. - f) * qweight  * qweight / (f * S );
//this goes the the Matrix element (dim+1,dim+1)
double dQ_dR = B*qweight * qweight /(f*f*S);
```

Example Code

```
//Lets setup the necessary Matrices
const TMatrixDSym& cov = r->covarianceMatrix();
TMatrixDSym mderivs(bckgrnd_PolOrder+2);

//Iterate over the background
TIterator *it = bkgArgSet.createIterator();
RooRealVar *tmp_param=NULL;
int deriv_dim = 0;

while( (tmp_param=(RooRealVar*)it->Next()) ){
  RooDerivative roodervar("roodervar", "roodervar", pol, (*tmp_param));
  mderivs(deriv_dim,deriv_dim) = roodervar.getVal()*dQ_dB;
  deriv_dim++;
}
```

Example Code

```
RooDerivative roodervar("roodervar", "roodervar", gaussFunction, sigma);

mderivs(deriv_dim,deriv_dim) = dQ_dR;
mderivs(deriv_dim+1,deriv_dim+1) = roodervar.getVal()*dQ_dS;

TMatrixD multmatrix( bckgrnd_PolOrder+2,bckgrnd_PolOrder+2 );
multmatrix.Mult( cov, mderivs);
TMatrixD fullmatrix( bckgrnd_PolOrder+2,bckgrnd_PolOrder+2 );
fullmatrix.Mult(mderivs, multmatrix);
```

Member of the Helmholtz Association

Example Code

```
double q2_err = 0.0;

for(int iMat1 = 0 ; iMat1 < deriv_dim+2; iMat1++) {
  for(int iMat2 = 0 ; iMat2 < deriv_dim+2; iMat2++) {
    double qerr_placer = fullmatrix(iMat1,iMat2);
    q2_err += qerr_placer;

  }
}
QweightError = sqrt(q2_err);
//########## End error #################
```

Example Code

```
    t4->Fill();

  }
  t4->Write();
  outFile.Write(); // write to the output file
  outFile.Close(); // close the output file

}
```

Individual Coding

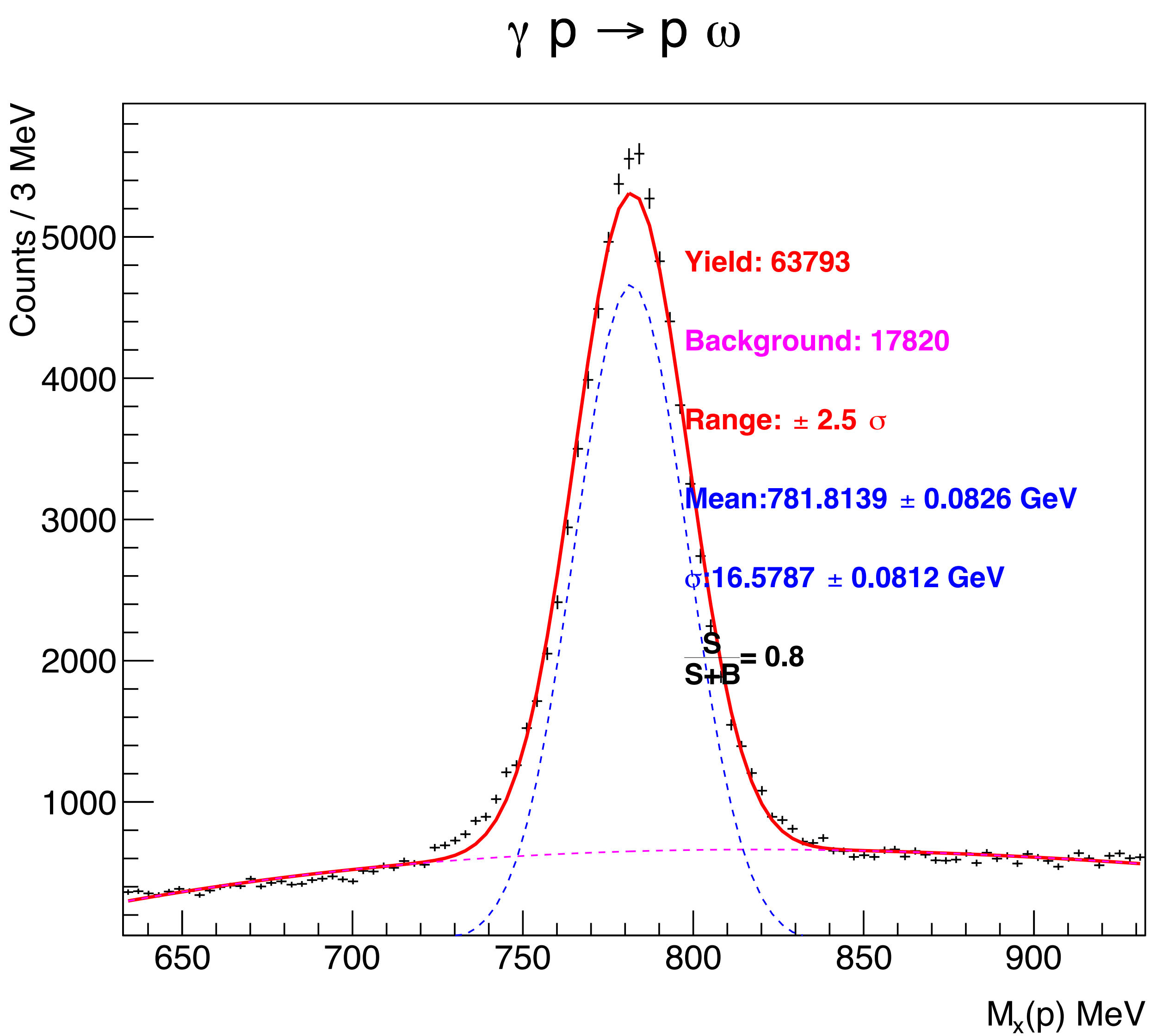
Pros:

- Not a blackbox. Not including whitespace and comments, methodology shown is 80 lines of code
    - Individual learns method

Downfalls:

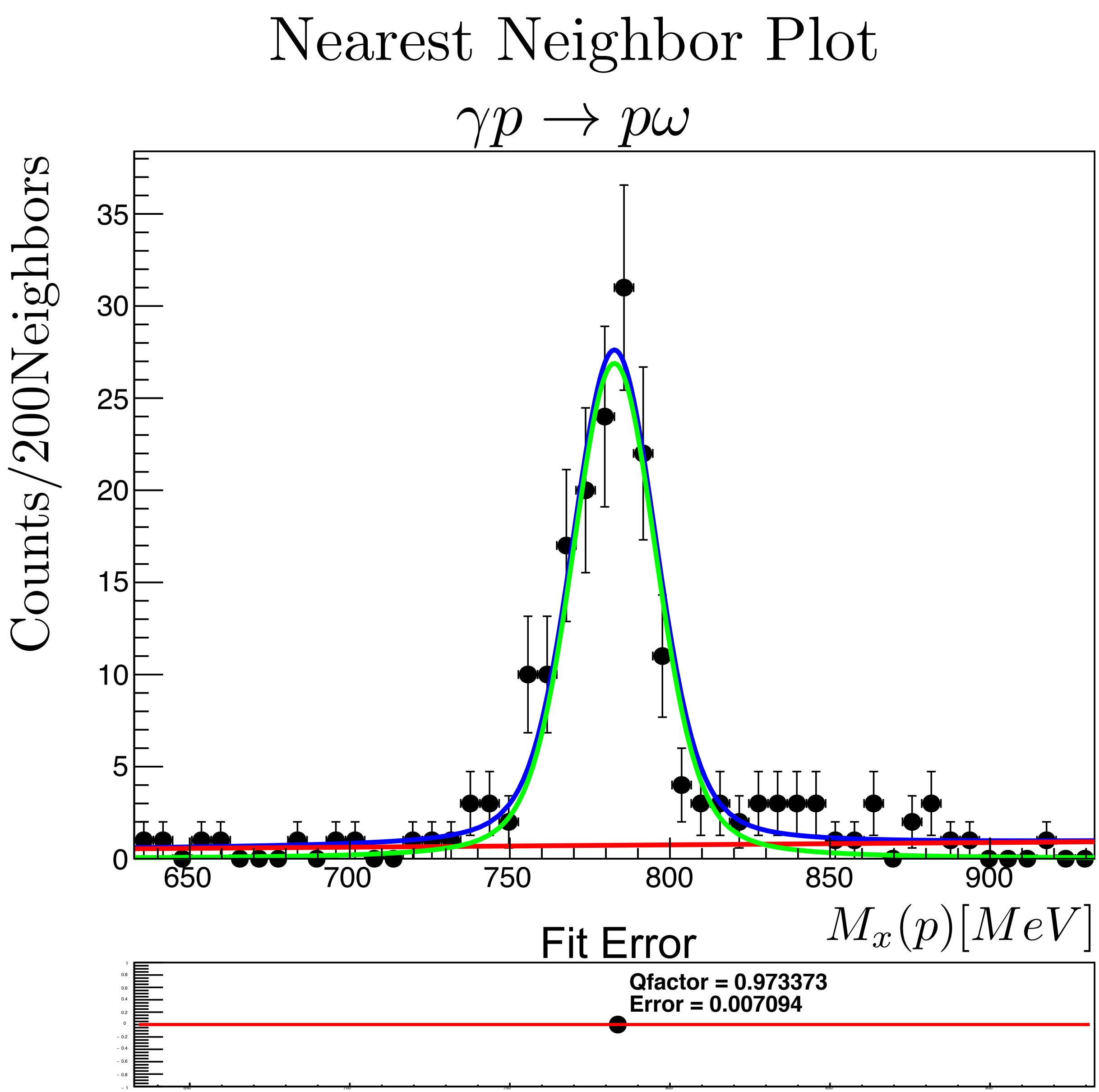
- Coding is per person and per reaction based
- Not distributable

Example



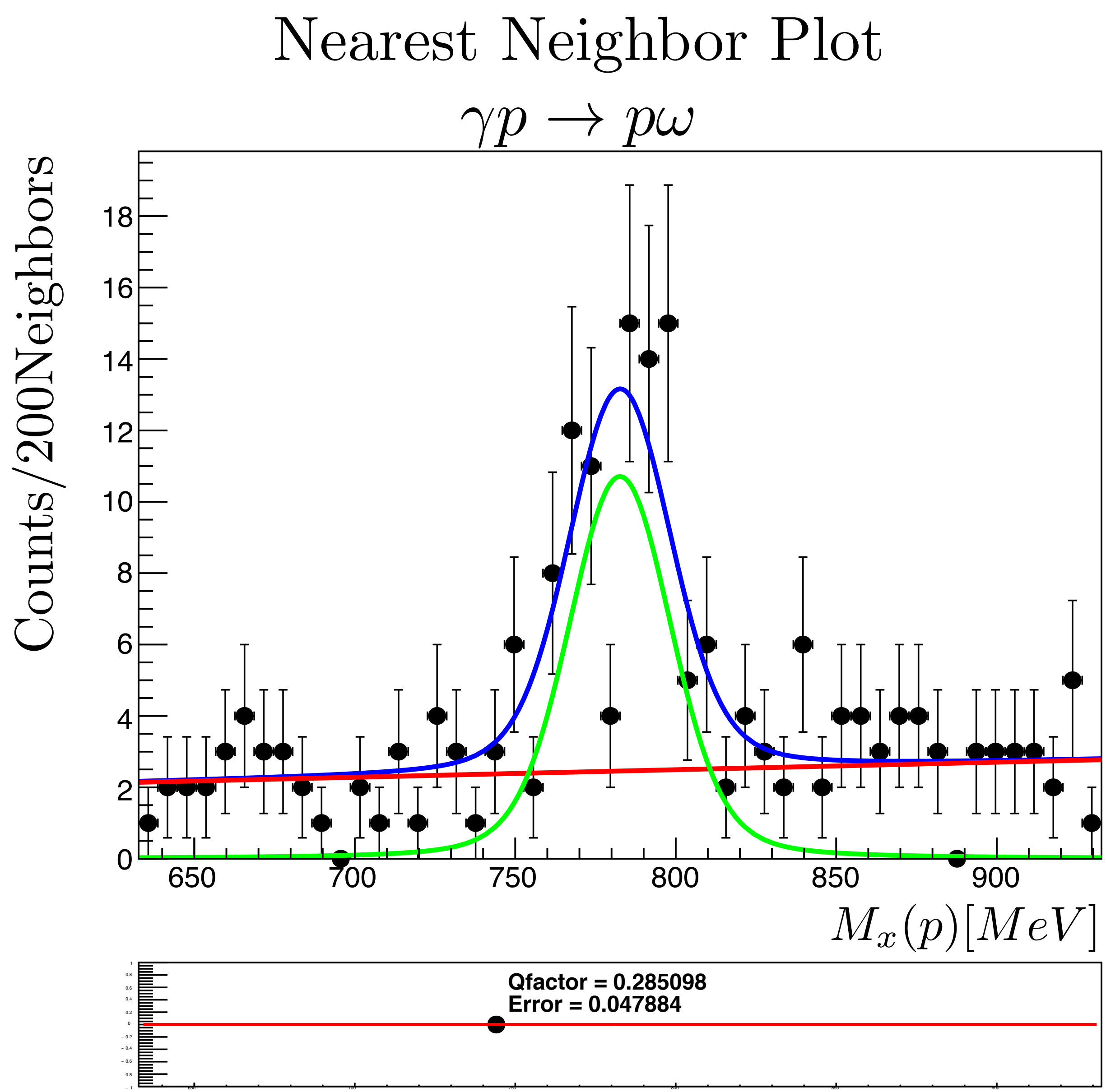
γ p → p ω
Counts / 3 MeV
5000
4000
3000
2000
1000
650
700
750
800
850
900
Yield: 63793
Background: 17820
Range: ± 2.5 σ
Mean:781.8139 ± 0.0826 GeV
σ:16.5787 ± 0.0812 GeV
S/S+B = 0.8
Mx(p) MeV

Results

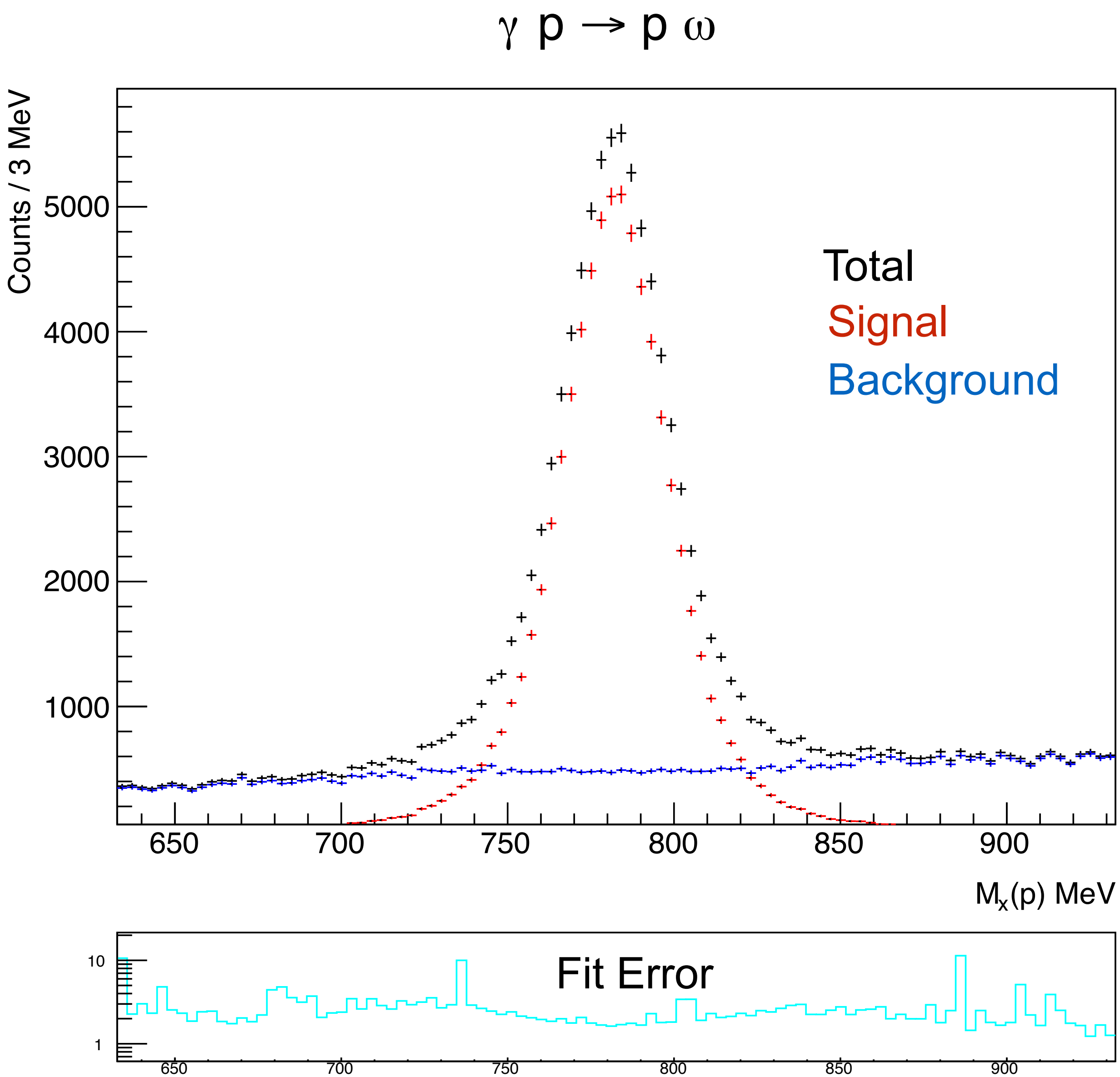


Nearest Neighbor Plot
$\gamma p \rightarrow p\omega$
Counts/200Neighbors
0
5
10
15
20
25
30
35
650
700
750
800
850
900
$M_x(p)[MeV]$
Fit Error
Qfactor = 0.973373
Error = 0.007094

Results


Nearest Neighbor Plot
$\gamma p \to p\omega$
Counts/200Neighbors
0
2
4
6
8
10
12
14
16
18
650
700
750
800
850
900
$M_x(p)[MeV]$
Qfactor = 0.285098
Error = 0.047884

Results



γ p → p ω
Counts / 3 MeV
5000
4000
3000
2000
1000
Total
Signal
Background
650
700
750
800
850
900
M_x(p) MeV
Fit Error
10
1
650
700
750
800
850
900

Caveats and Insights

Caveats:

- There must be m coordinates greater that 1
- Must know background and signal functions
    - Method is dependent on quality of this knowledge
- CPU intensive
- Time intensive

Insights:

- Q can be calculated in multiple dimensions
- Can reduce in-peak background depending on input coordinates
- Method is dependent on quality of this knowledge

Conclusion

Q-Factor

- Separates signal from background
- If done properly error on fit is minimum

Individual Method:

- Easily coded
- Time consuming

http://arxiv.org/abs/0809.2548

Plot

```
if(!(i%1000))
{

  TCanvas canvas("canvas", "My plots", 0, 0, 550, 500);
  TPad upperPad("upperPad", "upperPad", .005, .1525, .995, .995);
  TPad lowerPad("lowerPad", "lowerPad", .005, 0.005, .995, .1525);
  upperPad.Draw();
  lowerPad.Draw();

  upperPad.cd();
  RooPlot *frame = mass.frame(Title("My Roo Plot"));
  frame->SetXTitle("M_{#pi^{0}}^{2} + 0.5^{2} GeV^{2} [GeV^{2}]");
  frame->SetYTitle("Events of Nearest Neighbors");

  data.plotOn(frame, Binning((int)(N_NearNeighbor / 4)), DataError(RooAbsData::SumW2));
  model.plotOn(frame, RooFit::LineColor(kBlue),RooFit::Normalization(sumOfWeights, RooAbsReal::NumEvent));
  pol.plotOn(frame, RooFit::LineColor(kRed),
             RooFit::Normalization((1-f)*sumOfWeights, RooAbsReal::NumEvent));
  gaussFunction.plotOn(frame, RooFit::LineColor(kGreen),
                       RooFit::Normalization(f*sumOfWeights, RooAbsReal::NumEvent));

  frame->Draw();

  //Now lets draw the refererence point to show where it is on the spectrum
  lowerPad.cd();
  lowerPad.DrawFrame(Min_Mass, -1, Max_Mass, 1, "");
```

Plot

```
  //lets create a TLine
  TF1 f1("f1","[0]",Min_Mass,Max_Mass);
  f1.FixParameter(0,0);
  //lets make TGraph
  Double_t x[1], y[1];
  x[0] = MM2_P; y[0] = 0.;
  TGraph *gr = new TGraph(1,x,y);

  gr->SetMarkerStyle(20);
  gr->SetTitle("");
  gr->Draw("P");
  f1.Draw("same");

  //Lets display the Qfactor of this individual reference point
  Float_t m_size = gr->GetMarkerSize(); // For the sizing of the TLatex
  TString qvalue; qvalue.Form("\t Qfactor = %f",qweight);
  TString q_err; q_err.Form("\t Error = %f",sqrt(q2_err));

  TLatex t;
  t.SetTextSize(0.15*m_size);
  t.DrawLatex(MM2_P , .55 ,qvalue);

  TLatex t_err;
  t_err.SetTextSize(0.15*m_size);
  t_err.DrawLatex(MM2_P , .20 ,q_err);

  //End TGraph

  canvas.SaveAs(creater);
  delete frame;



}
```