



Introduzione ai tool di analisi

II. Fittare le PDF

Pietro Biassoni^{1,2}

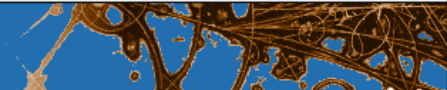
¹INFN Milano

²Dipartimento di Fisica
Università degli Studi di Milano

Metodologie di Analisi dei Dati
27 Novembre 2009



Outline



- 1 Introduzione
- 2 La nostra prima PDF
- 3 Problematiche Comuni
- 4 Esercizi



Outline

1 Introduzione

2 La nostra prima PDF

3 Problematiche Comuni

4 Esercizi



Fittare le PDFs

Dopo avere introdotto le funzioni base di RooFit, passiamo a provare a fittare delle PDFs. Il compito non è banale:

Cosa impareremo a fare:

- Determinare la forma funzionale più appropriata.
- Capire se il fit sta avendo problemi.
- Controllare che ci sia un buon accordo dati/curva.
- Evitare l' "Ufficio Complicazione Affari Semplici".

Perché è importante

I fit a Maximum Likelihood si basano sulle PDFs. Saper modellare delle "buone" PDFs è indispensabile per avere un "buon" fit.



Prerequisiti

Cosa do per scontato:

- Manipolazione TFile.
- Manipolazione TTree.
- Manipolazione RooDataSet.

Dopo queste lezioni sarete in grado di:

- Fittare un Dataset con una PDF appropriata.
- Evitare errori banali nel fit.
- Fare gli esercizi che vi assegneremo :-)



Outline

- 1 Introduzione
- 2 La nostra prima PDF
- 3 Problematiche Comuni
- 4 Esercizi



Individuare il range

Il dataset per fittare la nostra prima PDF si trova in
/home/comune/AnalisiDati/first.root: vediamo la serie di
operazioni da svolgere (che poi svolgerete tra gli esercizi).

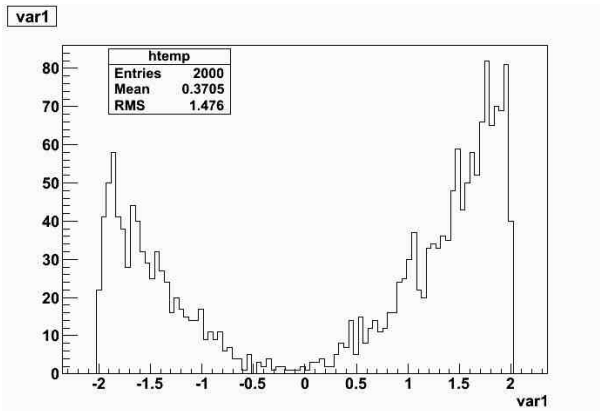
Innanzitutto dobbiamo guardare che forma ha il dataset, usiamo la
seguente macro

```
using namespace RooFit;  
  
int first(){  
    TFile *f = new TFile("first.root","READ");  
    TTree *t = (TTree*)f->Get("t1");  
  
    TCanvas *c1 = new TCanvas("c1","c1",1);  
    t->Draw("var1");  
}
```



Individuare il range

Questo dovrebbe essere il risultato:



Quindi il range della variabile è $[-2,2]$.



Dichiarare il RooDataset

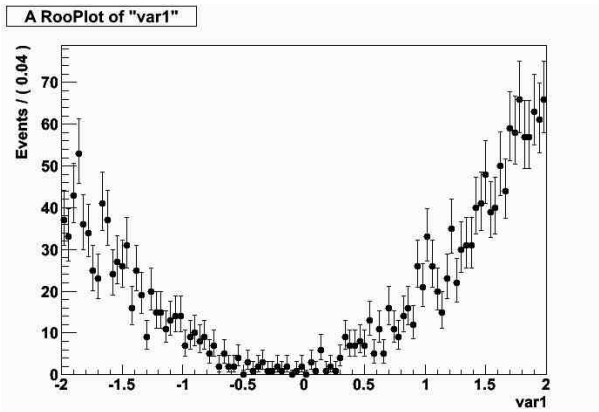
Aver disegnato `var1` ci permette di conoscere il suo range. Dichiariamo ora la `RooRealVar` e il `RooDataSet`

```
int first(){  
    ...  
    //dichiaro la variabile  
    RooRealVar var1("var1","var1",-2,2);  
    //dichiaro il dataset  
    RooDataSet data("data","data",t,RooArgSet(var1));  
  
    //dichiaro il RooPlot e disegno il RooDataset  
    RooPlot *frame = var1.frame();  
    data.plotOn(frame);  
  
    TCanvas *c1 = new TCanvas("c1","c1",1);  
    frame->Draw();  
}
```



Dichiarare il RooDataset

Questo dovrebbe essere il risultato:



I dati sembrano distribuiti come un polinomio: dichiariamo un polinomio e fittiamo.



Fittare la PDF

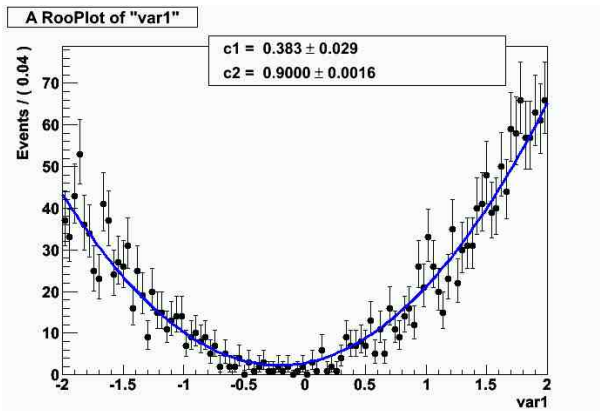
Dichiariamo ora la RooAbsPdf

```
int first(){  
    ...  
    //dichiaro i parametri  
    RooRealVar c1("c1","c1",0.1,-0.9,0.9);  
    RooRealVar c2("c2","c2",0.1,-0.9,0.9);  
    //dichiaro la PDF  
    RooChebychev pdf("pdf","pdf",var1,RooArgList(c1,c2));  
    //fitto la PDF  
    pdf.fitTo(data,Minos(kFALSE));  
    //disegno il dataset e la PDF sul RooPlot  
    data.plotOn(frame);  
    pdf.plotOn(frame);  
    pdf.paramOn(frame);  
    ...  
}
```



Fittare la PDF

Questo dovrebbe essere il risultato:



Le cose stanno andando bene?



Parametri al limite

Attenzione!!!
Il parametro "c2" è al limite!

$$c1 = 0.383 \pm 0.029$$

$$c2 = 0.9000 \pm 0.0016$$

Questo vuol dire che nella dichiarazione

```
RooRealVar c2("c2","c2",0.1,-0.9,0.9);
```

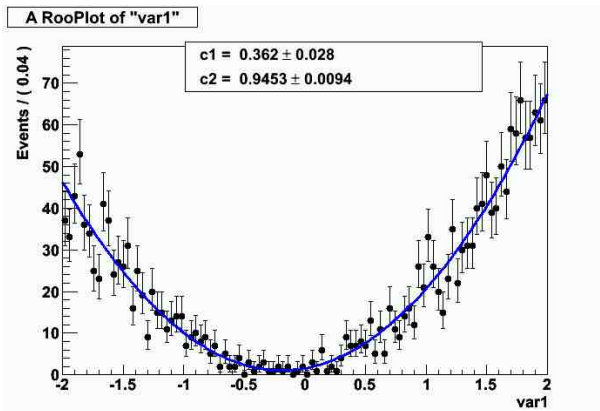
dobbiamo allargare il range del parametro

```
RooRealVar c2("c2","c2",0.1,-1.,1.);
```



Parametri al limite

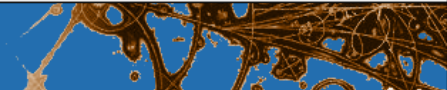
Questo dovrebbe essere il risultato:



Le cose stanno andando bene? SI!



Outline

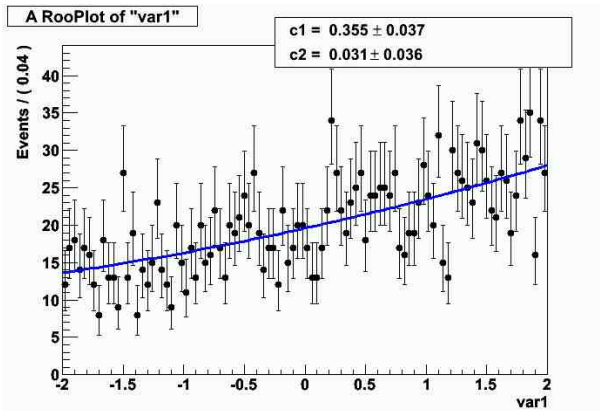


- 1 Introduzione
- 2 La nostra prima PDF
- 3 Problematiche Comuni
- 4 Esercizi



Evitare complessità inutile

Ho fittato un dataset usando un polinomio di secondo grado



Le cose stanno andando bene?



Evitare complessità inutile

ATTENZIONE!!! Il parametro "c2" è consistente con zero!

Questo vuol dire che un polinomio di primo ordine basta a fittare i dati

$$c1 = 0.355 \pm 0.037$$

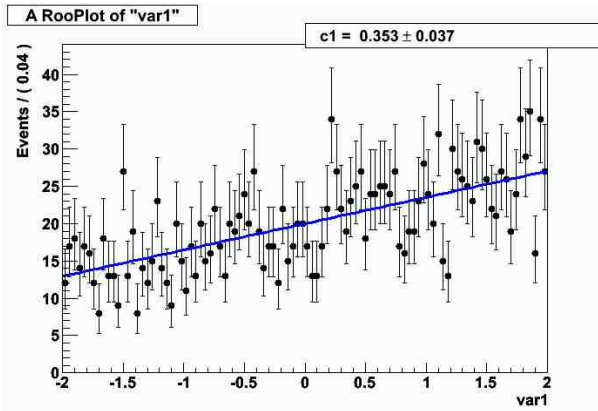
$$c2 = 0.031 \pm 0.036$$

Nelle misure di un parametro fisico si assegna un errore *sistematico* legato all'incertezza sui parametri delle PDFs

PDFs più semplici = errore *sistematico* più piccolo.



Evitare complessità inutile

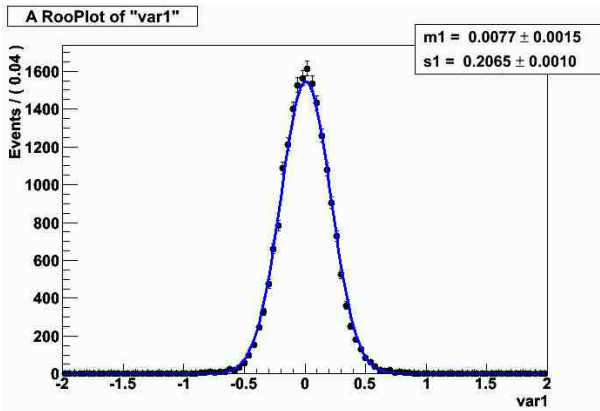


Le cose stanno andando bene? SI!



Code

Ho fittato un dataset usando una Gaussiana



Le cose stanno andando bene? Sembrerebbe di si...

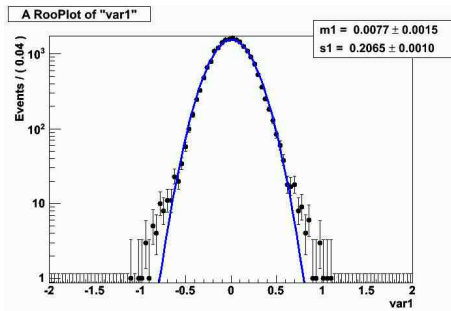


Code

Plot logaritmici

Ora guardiamo la stessa figura usando un plot logaritmico

```
c1->SetLogy();
```



La distribuzione ha delle code che nel plot non logaritmico non erano visibili: necessario fittare con una doppia gaussiana.



Code

PDF composte

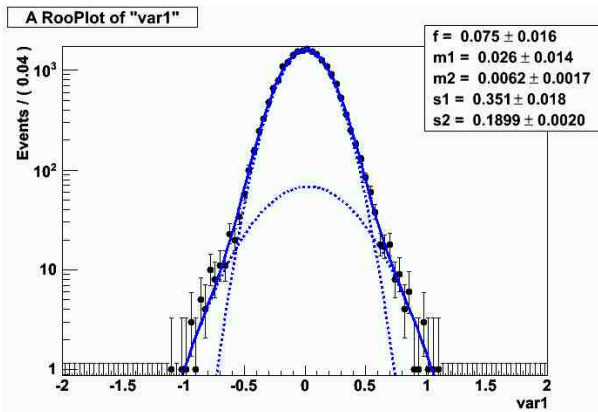
```
RooGaussian g1PDF("gauss","gauss",var1,m1,s1);
RooGaussian g2PDF("gauss2","gauss2",var1,m2,s2);
RooAddPdf *var1PDF = new RooAddPdf("var1PDF","var1PDF",
                                     RooArgList(g1PDF,g2PDF),RooArgList(f));

var1PDF->fitTo(data,Minos(kFALSE));
var1PDF->plotOn(var1Frame);
var1PDF->plotOn(var1Frame,Components("gauss"),
               LineStyle(kDashed));
var1PDF->plotOn(var1Frame,Components("gauss2"),
               LineStyle(kDashed));
```



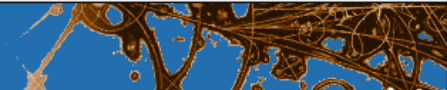
Code

PDF composte





Outline



- 1 Introduzione
- 2 La nostra prima PDF
- 3 Problematiche Comuni
- 4 **Esercizi**



Ora tocca a voi! (Volume 1)



Esercizio: Fittare il dataset presente in `/home/comune/first.root` seguendo quanto spiegato in queste slides (giusto per prenderci la mano).



Ora tocca a voi! (Volume 2)



Esercizio: Nel file `/home/comune/pdfs.root` è presente un albero con 8 branches (`var1`, `var2`, ...). Fittare le PDFs per ogni variabile. Mandateci via mail tutti i plot dei risultati in un unico file chiamato `pdfs_nome_cognome.tar.gz`