Fit con Root

Stefano Lacaprara

Cos'e' l'analisi dati

- L'analisi dati consiste nell'estrarre da una (decentemente confusa) serie di dati alcune (poche) quantita' di interesse fisico

- Visto che tutte le quantita' misurabili sono affette da rumore (strumentale, stocastico, ecc), e' necessario adottare procedure per eliminare tale componente

Inoltre….

- non sempre le quantita' misurate sono pure, cioe' non affette da misurazioni di quantita' che poco c'entrano con quello che vogliamo misurare

- per esempio:
  - voglio misurare la temperature di una stanza
  - ma se c'e' gente dentro la temperatura cambia, per cui dovrei avere modo di capire se la stanza e' occupata oppure no
    - non sempre posso farlo, devo cercare di capire quale sia la temperatura media della stanza senza saperlo

Misure

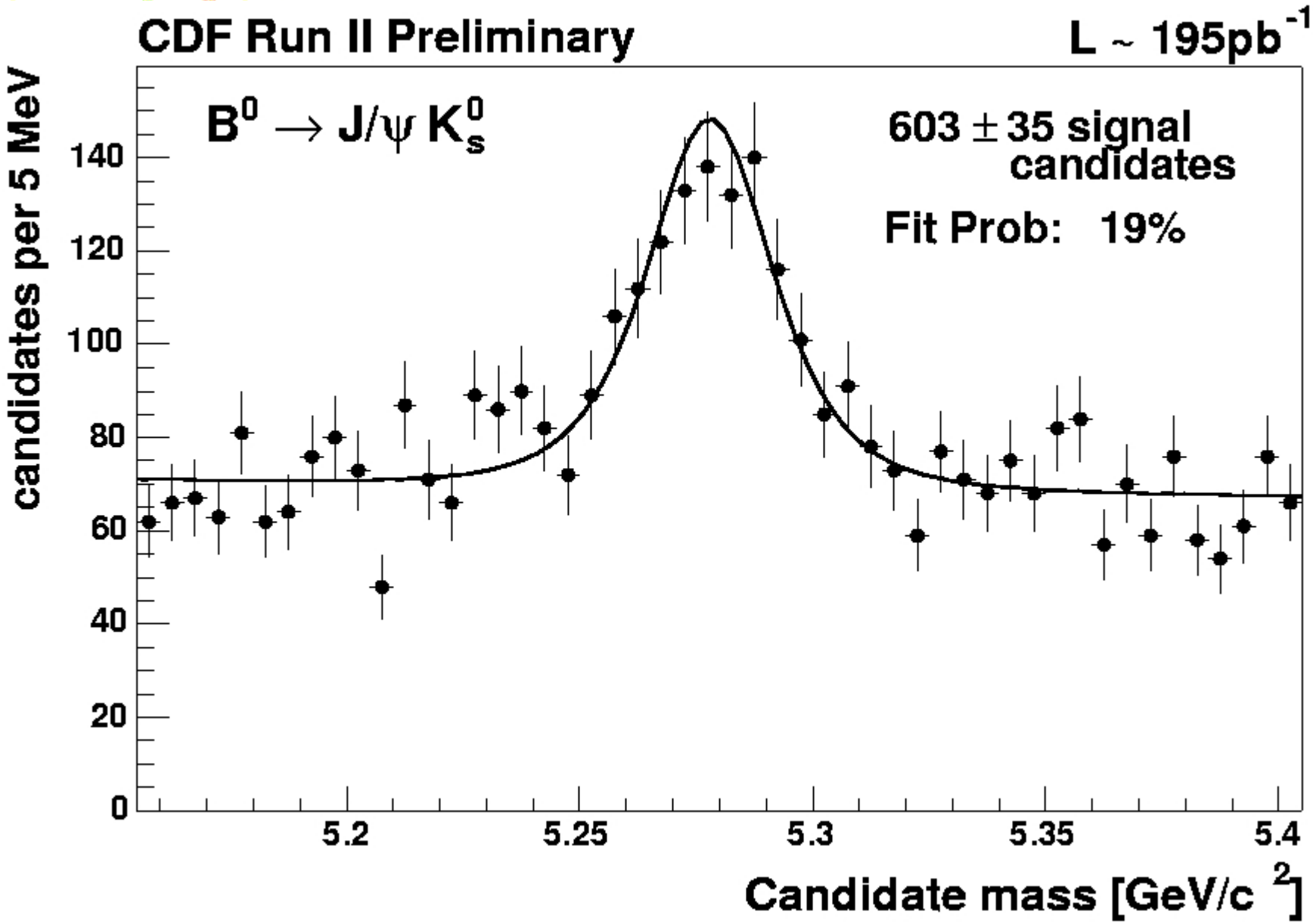
- le 3 componenti di una misura possono essere divise in
  - segnale
    - quello che vorremmo misurare
  - fondo
    - misure che vengono da fenomeni diversi da quello che vogliamo misurare
  - rumore
    - componente aggiuntiva dovuta tipicamente agli strumenti di misura (o alla meccanica quantistica…)
- Misura = segnale + fondo + rumore

Come estrarre il segnale?

- serve:
  - una modellazione del comportamento del segnale
  - una modellazione del comportamento del fondo
  - una modellazione del rumore

  - un FIT!

esempio …


CDF Run II Preliminary
L ~ 195pb^{-1}
$B^0 \rightarrow J/\psi\, K^0_s$
603 ± 35 signal candidates
Fit Prob: 19%
candidates per 5 MeV
0
20
40
60
80
100
120
140
5.2
5.25
5.3
5.35
5.4
Candidate mass [GeV/c^2]

- posso dire a occhio
  - ma la simulazione lo deve confermare
  - che il segnale e' gaussiano
  - il fondo e' circa piatto
  - il rumore c'e'…
- da quel plot voglio estrarre il valore medio e larghezza della gaussiana … come fare?

FIT

- Ci sono molti modi, ma il soldoni sapete gia'cosa sia
  - si prendono dei dati
  - si prende una funzione matematica con dei parametri liberi
    - Bisogna scegliere la funzione giusta! Nessun programma lo puo' fare per noi
  - si cambiano I parametri fino a quando I dati e la funzione sono il piu'simili possibile
    - e qui sta il casino …
- Farlo a mano e' complicato: si usano programmi che sanno farlo

Root …

- Sa fare (nativamente) fit di funzioni semplici
  - gaussiane
  - polinomiali
  - landau
  - esponenziali
  - Ecc...
- Come si fa:
  - Stupido, dato un istogramma, aprire il Fit Panel
- oppure
  - isto.Fit(“gaus”)

ma se voglio cose piu’complesse?

- Se non mi basta fare fit a gaussiane ecc ecc?
- Con root si puo' fare

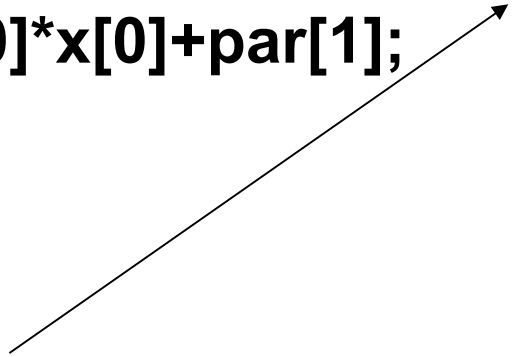
Per esempio:

- root permette di usare una funzione in C++ (con parametri liberi) per effettuare il fit
- Cioe' posso fittare una distribuzione con una funzione a mia scelta
  - Qualunque funzione!
  - Potrei anche fittare con la funzione di mandelbrot!

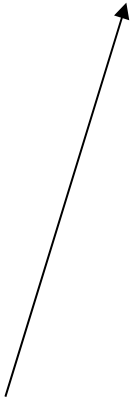
Come fare?

- definire una funzione C++ del tipo

```
double funzione(double* x, double* par){
    return par[0]*x[0]+par[1];
}
```



puntatore, consideratelo un vettore delle coordinate se e'una funzione ad una sola variabile, x[0] e'x

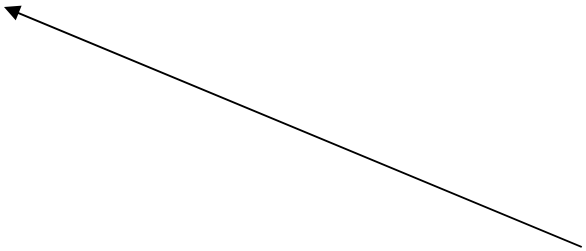


puntatore, consideratelo un vettore dei parametri liberi

- ho quindi definito una funzione ad una variabile (x[0]) che ha due parametri liberi (par[0] e par[1])

- in pratica

  - funzione = P_0*x+P_1

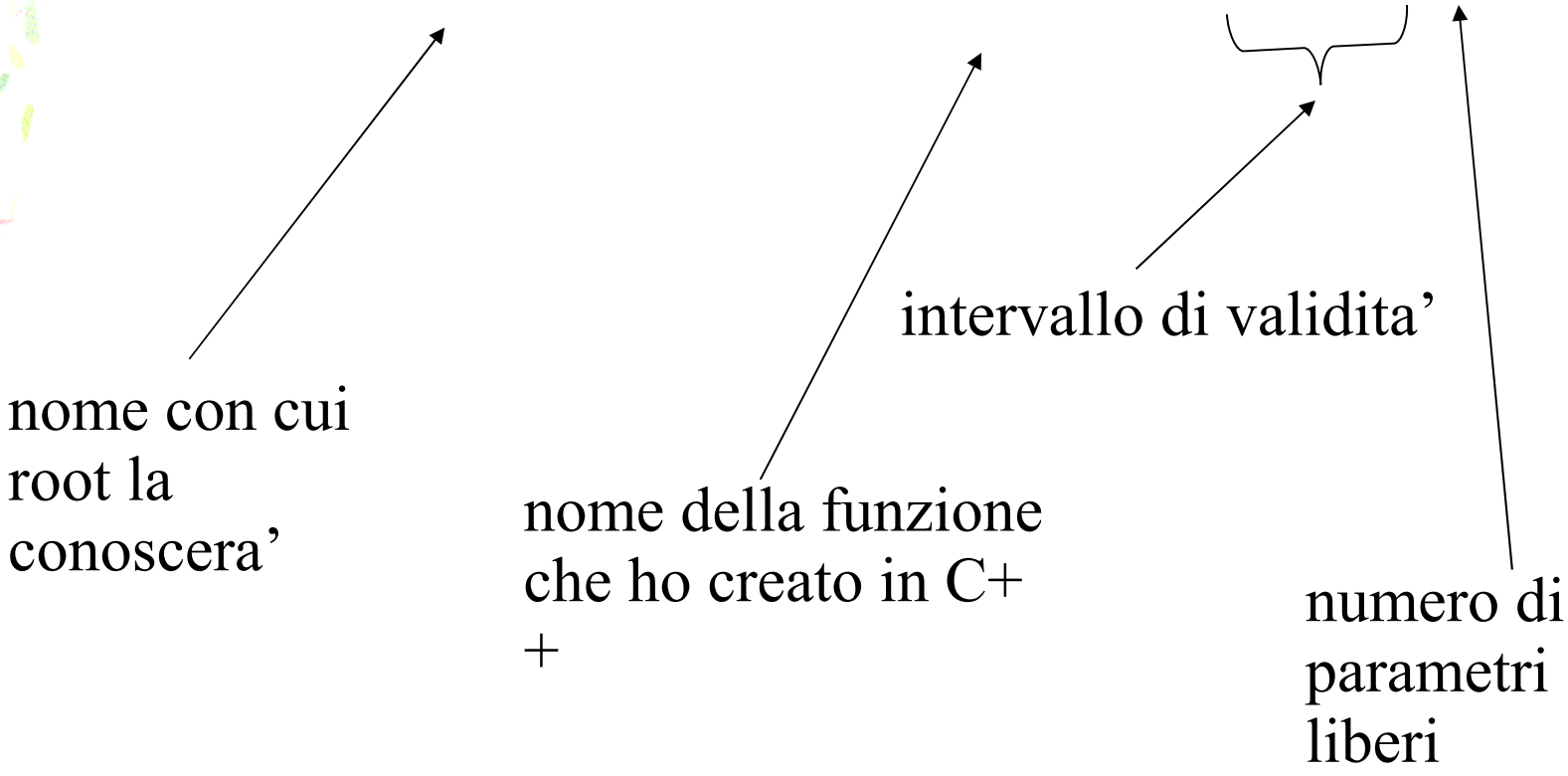


sto facendo il fit ad una retta!

a questo punto …

- devo insegnare a root questa funzione

- TF1 funz(“funzioneMia”, funzione,0,100,2);



A questo punto

- isto.Fit(“funzioneMia”);
- esegue il fit…

- facciamo un tentativo completo!

Fisica delle alte energie

- ho eseguito una misura in fisica delle alte energie, devo estrarre il valore della massa di una particella, e la sua vita media
  - posso modellizzare il segnale dalla particella come una gaussiana, il cui valore medio e' la massa e la cui sigma e' legata alla vita media
- la misura ha del fondo dovuto a fenomeni che al momento non ci interessano, ma il fondo posso supporlo constante (a meno del rumore)

Ok, complichiamoci la vita

- Supponiamo che i miei dati come sovrapposizione di una gaussiana, il segnale e di una distrtibuzione piatta (fondo)

- Creiamo un istrogramma e lo riempiamo con gaussiana e uniforme
- Poi facciamo un fit e cerchiamo di recuperare i parametri della gaussiana

ora…

- Definiamo una funzione che (a nostro giudizio) riproduce la distribuzione
- gaussiana+costate, quindi con 3+1 parametri liberi
- facciamo il fit…

$$f(x) = P_1 * e^{\left(\frac{(x - P_2)^2}{P_3^2} \right)} + P_4$$

- Ci interessano P_2 e P_3

Una volta fatto il fit…

- possiamo avere in C++ il risultato, con

- funz.GetParameter(2); // legge par[2]
- funz.GetParError(2); // legge l'errore su par[2]

- prego … provare

ha funzionato?

- probabilmente no … il fatto e' che la procedura di fit non puo'essere piu'di tanto furba, per cui devo dare di valori di partenza "non inverosimili"

- Prima di fare il fit, posso fare
- funz.SetParameter(2,66);
  - inizializza il par[2] a 66 prima di fare il fit

se la sigma viene negativa?

- possiamo dare dei limiti ai parametri liberi
- per esempio
- funz.SetParLimits(3,0,100);
  - impongo ch par[3] possa andare da 0 a 100 solamente

Alternativa

- In root (come nella vita) ci sono molti modi di fare la stessa cosa. Vediamone altri
- Definirie una funzione: gaus+constante
  - sono due funzioni che root gia' conosce (gaus+pol0)
  - devo solo combinarle

```
TF1 gausPlusConst(“gausPlusConst”,
                  ”gaus(0)+pol0(3)”,-10.,10.)
```

gausiana: il primo parametro e' il numero 0

polinomiale di grado 0 (costante); primo parametro e' il numero 3

Se voglio disegnarla

gausPlusConst.Draw()

ma prima devo dare un valore ai parametri (altrimenti valgono tutti 0)

```
g.SetParameter(0,100);
g.SetParameter(1,0);
g.SetParameter(2,1);
g.SetParameter(3,10);
g.DrawCopy();
```

Posso anche riutilizzarla

- Posso definire una “formula”

```
TFormula ff(“ff ”, “abs(sin(x)/x)”);
TF1 h("h","g+ff",-10,10);
```

- Oppure una combinazione

```
form1 = new TFormula("form1","abs(sin(x)/x)");
sqroot = new TF1("sqroot","x*gaus(0) + [3]*form1",0,10);
sqroot->SetParameters(10,4,1,20);
```

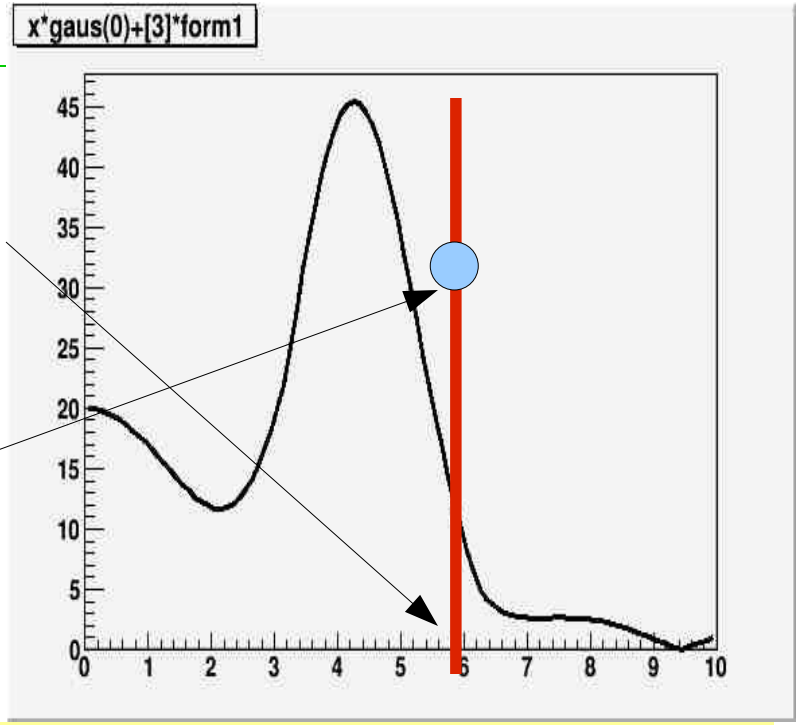
parametro #3

Riempire isto con random

- Si puo' fare a mano:
  - si generano numeri casuali con la distribuzione voluta
  - isto.Fill(x);
- Se non abbiamo un generatore per la distribuzione voluta dobbiamo farlo noi
  - Metodo dello scarto
- Oppure lo si fa fare a root

Metodo dello scarto

- Ho una funzione f(x)
- Genero un numero casuale x tra Xmin e Xmax
- Genero un secondo numero casuale y tra Ymin e Ymax
  - Se y<f(x), allora riempio l'istogramma con x
  - se y>f(x), allora scarto x e riparto
- Alla fine ho un istogramma riempito con numeri casuali distribuiti come f(x)



E' un buon esercizio da fare a casa!

Inventatevi una funzione, riempite un istogramma con numeri casuali secondo quella distribuzione e poi fate un fit con la funzione.

Lo fa root!

- Molto semplicemente

```
TF1 miaFunc("miaFunc",...); // come prima

histo.FillRandom("miaFunc",10000);
```

Riempi istogramma histo con 10000 numeri casuali distribuiti secondo “miaFunc”

Esercizio

- Create un istogramma e riempitelo con due gaussiane e uniforme (due particelle con massa vicina) eg
  - gaus1: n=1000, mean = 10.3, sigma=1.5
  - gaus2: n=500, mean = 13.1, sigma = 0.7
  - Fondo: n=2000 (uniforme 0-20)
- Fate un fit con una funzione appropriata e ritrovate le due particelle
  - NB: ci vuole qualche iterazione per scegliere bene i parametri iniziali