Tutto quello che avreste voluto sapere sui fit* (*ma non avete mai osato chiedere) (parte I)

Laboratorio di Metodi Computazionali e Statistici (2023/2024)

R. Cardinale, F. Parodi, S. Passaggio

November 22, 2023

Recap: fit di un grafico di punti

Fit di un grafico con errore sulle x e sulle y.

```
1 {
2    TGraphErrors *gr = new TGraphErrors("pendolo.dat");
3    TF1 *f = new TF1("f","[1]*x+[0]",0,10);
4    f->SetParameter(0,4);
5    f->SetParameter(1,0);
6    gr->Fit("f");
7 }
```

Analisi dei risultati

- Matrice di covarianza dei parametri (tramite FitResultPtr)
- Banda di "errore" della funzione (con costruzione esplicita del grafico dei punti a $\pm 1\sigma$)
- Contour nello spazio p_1 vs p_0 in corrispondenza di un certo $\Delta\chi^2$

Definizione custom del χ^2 (C++)

```
1  namespace data{
2    vector<double>  x, y, ex, ey;
3  }
4
5  double fun(const double *x,const double *par){
6    return par[1]*(*x)+par[0];
7  }
8
9  void fcn(int &npar,double *gin,double &f,double *par,int iflag){
10   f = 0.0;
11   for (int i=0;i<data::x.size();i++){
12     // Calcolo del Chi2
13   }
14 }
```

- L'utente deve sempre fornire una funzione con questo prototipo
- Significato dei parametri in input/output:
  - `npar` numero di parametri (input)
  - `gin` vettore delle derivate prime della funzione nei parametri (opzionale, output)
  - `f` valore della funzione (output)
  - `par` vettori di parametri (costanti o variabili) (input)
  - `iflag` intero che indica lo stadio di minimizzazione (input)

Per iniziare è sufficiente definire `f` sulla base di `par`.

Definizione χ^2 (C++)

```
1
2  void fitlin(){
3    ifstream file("pendolo.dat");
4    double x,y,ex,ey;
5    while (file >> x >> y >> ex >> ey){
6      data::x.push_back(x); data::y.push_back(y); data::ex.push_back(ex); data
           ::ey.push_back(ey);
7    }
8
9    // Define the minimization problem
10   // TMinuit *minuit = new TMinuit(...); // numero di parametri
11   minuit->SetFCN(fcn);
12   // minuit->DefineParameter(indice,"nome",valin,step,min,max);
13   // per ogni indice inserisco nome, valore iniziale, step, minimo, massimo
         del parametro
14
15   // Minimize
16   minuit->Command("MIGRAD"); // Comando di minimizzazione
17
18   // Get result
19   // minuit->GetParameter(indice,val,eval);
20   // per ogni indice estraggo il valore del parametro e del suo errore
21   // minuit->GetParameter(indice,val,eval);
22
23 }
```

Interfaccia Minuit in Python (iminuit)

Interfaccia Python a Minuit, indipendente da ROOT.
Funzionamento del modulo Minuit in iminuit

- La funzione da minimizzare può essere definita in due modi:
  - Parametri espliciti

    ```
    def fcn(a, b, c):
      ...
    Minuit(fcn,...)
    ```

    potete passare i parametri con assegnazione per nome (nome=...)
  - Numpy array

    ```
    def fcn(par):
      ...
    Minuit(fcn,par)
    ```

    `par` deve essere un numpy array.
- Tutti gli altri parametri di Minuit sono *default*, ma possono essere cambiati con i metodi:
  - `errordef` definisce $\Delta\chi^2$ per il calcolo degli errori (default 1)
  - `print_level` definisce il livello di output (default 0)
  - `limits` definisce limiti sui parametri

Definizione χ^2(iminuit): singoli parametri

```
1  from    iminuit import Minuit
2  from    numpy   import *
3  import matplotlib.pyplot as plt
4
5  def f(x,b,a):
6      return a*x+b
7
8  def chi2(a,b):
9      val = 0
10     for i in range(0,len(x)):
11         val = val + ((y[i]-f(x[i],a,b))/ey[i])**2
12     return val
13
14 # Acquisizione dati
15 x,y,ex,ey = loadtxt('pendolo.dat',usecols=(0,1,2,3),unpack=True)
16
17 ## Chiamo Minuit nella modalita' parametri passati tramite
       passaggio dei singoli parametri
18 ## Esplorare la funzione draw_mnmatrix per fare i contour
```

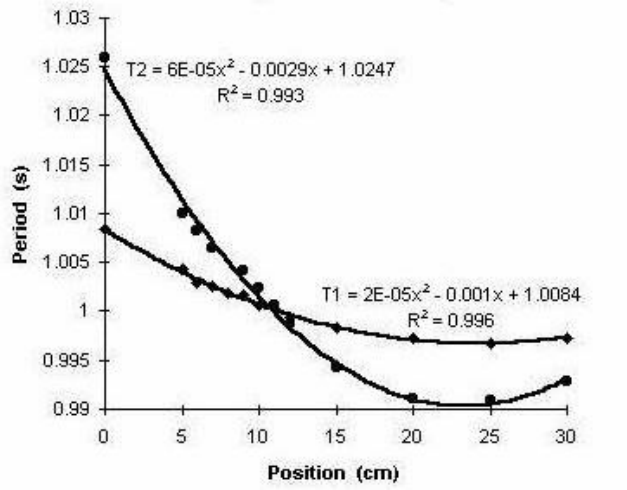
Definizione χ^2(iminuit): vettori di parametri

```
1  from iminuit import Minuit
2  from numpy   import *
3  import matplotlib.pyplot as plt
4
5  def f(x,par):
6      return par[1]*x+par[0]
7
8  def chi2(par):
9      val = 0
10     for i in range(0,len(x)):
11         val = val + ((y[i]-f(x[i],par))/ey[i])**2
12     return val
13
14 # Acquisizione dati
15 x,y,ex,ey = loadtxt('pendolo.dat',usecols=(0,1,2,3),unpack=True)
16
17 ## Chiamo Minuit nella modalita' parametri passati tramite array
```

Fit simultanei

La possibilità di costruire autonomamente la funzione di χ^2 consente di implentare strategie complesse di fit che permettano di estrarre direttamente la grandezza di interesse.

- Ad esempio una delle esperienza del primo anno consisteva nel determinare g con il pendolo di Kater:
  - si fittavano le due curve (periodo vs distanza dal perno): il punto di intersezione fornisce il periodo di isocronia.
  - problemi: propagazione complessa dell'errore sul punto di intersezione. Approssimazione (arbitraria): prendo come errore sul punto di isocronia l'errore medio sui periodi.



- Soluzione: di esegue un fit simultaneo delle due serie di dati con le funzioni:

$$f_2(x) = a_2(x - x_0)^2 + b_2(x - x_0) + T_0$$
$$f_1(x) = a_1(x - x_0)^2 + b_1(x - x_0) + T_0$$

Il valore fittato del parametro T_0 fornisce direttamente, senza approssimazioni, il periodo di isocronia con il suo errore.

Esercizio

Dall'esame del 15/01/2020:

Si vuole determinare il periodo di isocronia di un pendolo di Kater. Si sono registrati i periodi, per ciascuno dei due perni, in funzione della distanza della massa mobile.
Si vuole eseguire un file simultaneo ai due grafici usando come le funzioni:

$$f_1(x) = \alpha_1(x - x_0)^2 + \alpha_2(x - x_0) + T$$
$$f_2(x) = \beta_1(x - x_0)^2 + \beta_2(x - x_0) + T$$

Sfruttando la traccia `Kater.cpp` *si esegua il fit e si stampi il valore di* T *con il suo errore.*