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

Laboratorio di Metodi Computazionali e Statistici (2023/2024)

R. Cardinale, F. Parodi,S. Passaggio

November 30, 2023

Fit di Likelihood

Per fare esercizio con i fit di likelihood proviamo a fittare dati distribuiti secondo un esponenziale negativo (file `exp.dat`)

$$p(t) = \frac{1}{\tau} exp(-t/\tau)$$

eseguiremo i seguenti fit:

- Likelihood binned
- Likelihood binned (extended)
- Likelihood unbinned

Fit di Likelihood

Unbinned Maximum Likelihood (n eventi)

$$\mathcal{L} = \prod_{i=1}^{N} ln\ f(x_i;\theta) \qquad -ln(\mathcal{L}) = -\sum_{i=1}^{N} ln\ f(x_i;\theta)$$

Binned Maximum Likelihood (N bins)

$$\mathcal{L}(n_1, n_2, ..., n_N; p_1, p_2, ..., p_N) = \frac{n!}{n_1!n_2!...n_M!} p_1^{n_1} p_2^{n_2} ... p_N^{n_N}$$

$$-ln(\mathcal{L}) = -\sum_{i=1}^{M} n_i ln(p_i) + const$$

Binned Extended Maximum Likelihood (N bins)

$$\mathcal{L}(n_1, n_2, ..., n_N; p_1, p_2, ..., p_N) = \prod_{i=1}^{N} \frac{e^{-\nu_i} \nu_i^{n_i}}{n_i!} \qquad \nu_i = np_i$$

$$-ln(\mathcal{L}) = -\sum_{i=1}^{M} (n_i ln(\nu_i) - \nu_i)$$

Fit di Likelihood Unbinned

```
1  namespace data{
2    vector<double>  x;
3  }
4
5  void logl(int &npar, double *gin, double &f, double *par, int iflag){
6    f = 0.0;
7    for (int i=0;i<data::x.size();i++){
8      f += log(par[0]) + data::x[i]/par[0];
9    }
10 }
11
12 void fitexp(){
13
14   ifstream file("exp.dat");
15   double x;
16   TH1D *h = new TH1D("h","",40,0,10);
17   while (file >> x){
18     data::x.push_back(x);
19     h->Fill(x);
20   }
21   TMinuit minuit(1);
22   minuit.SetFCN(logl);
23   minuit.SetErrorDef(0.5);
24   minuit.DefineParameter(0,"tau",1.5,0.01,0.,0.);
25
26   minuit.Command("MIGRAD");
27   double tau, etau;
28   minuit.GetParameter(0,tau,etau);
29
30   h->SetMarkerStyle(20);
31   h->Draw("E");
32   TF1 *fe = new TF1("fe","[0]*1/[1]*exp(-x/[1])",0,10);
33   fe->SetParameter(1,tau);
34   fe->SetParameter(0,h->GetEntries()*h->GetBinWidth(1));
35   fe->Draw("SAME");
36 }
```

Fit di Likelihood Binned

```
1  from   ROOT    import *
2  from   iminuit import Minuit
3  import numpy   as     np
4  from   math    import *
5
6  def flogl(tau):
7      val = 0
8      for i in range(1,h.GetNbinsX()+1):
9          tmin = h.GetBinCenter(i)-h.GetBinWidth(1)/2
10         tmax = h.GetBinCenter(i)+h.GetBinWidth(1)/2
11         p    = (exp(-tmin/tau)-exp(-tmax/tau))
12         val = val - h.GetBinContent(i)*log(p)
13     return val
14
15 #Main
16 h  = TH1D("h","",20,0,10)
17 for line in open("exp.dat"):
18     h.Fill(float(line))
19
20 m = Minuit(flogl,tau=2,error_tau=0.01,errordef=0.5,print_level=2)
21 m.migrad()       # run minimiser
22
23 tau = m.values[0]
24 print(tau)
25 h.Draw("E")
26 hfit = TH1D(h)
27 for i in range(1,h.GetNbinsX()+1):
28     tmin = h.GetBinCenter(i)-h.GetBinWidth(i)/2
29     tmax = h.GetBinCenter(i)+h.GetBinWidth(i)/2
30     p    = (exp(-tmin/tau)-exp(-tmax/tau))
31     hfit.SetBinContent(i,p*h.GetEntries())
32 hfit.Draw("SAME")
33 hfit.SetLineColor(2)
34
35 gApplication.Run(True)
```

Fit di Likelihood Binned (Extended)

```
1  from   ROOT    import *
2  from   iminuit import Minuit
3  import numpy   as     np
4  from   math    import *
5
6  def flogl(tau,norm):
7      val = 0
8      for i in range(1,h.GetNbinsX()+1):
9          tmin = h.GetBinCenter(i)-h.GetBinWidth(1)/2
10         tmax = h.GetBinCenter(i)+h.GetBinWidth(1)/2
11         mu   = (exp(-tmin/tau)-exp(-tmax/tau))*norm
12         val = val - (h.GetBinContent(i)*log(mu)-mu)
13     return val
14
15 #Main
16 h  = TH1D("h","",20,0,10)
17 for line in open("exp.dat"):
18     h.Fill(float(line))
19
20 m = Minuit(flogl,tau=2,norm=1000,errordef=0.5,print_level=2)
21 m.migrad()       # run minimiser
22
23 tau  = m.values[0]
24 norm = m.values[1]
25 print(tau,norm)
26 h.Draw("E")
27 hfit = TH1D(h)
28 for i in range(1,h.GetNbinsX()+1):
29     tmin = h.GetBinCenter(i)-h.GetBinWidth(i)/2
30     tmax = h.GetBinCenter(i)+h.GetBinWidth(i)/2
31     p    = (exp(-tmin/tau)-exp(-tmax/tau))
32     hfit.SetBinContent(i,p*norm)
33 hfit.Draw("SAME")
34 hfit.SetLineColor(2)
35
36 gApplication.Run(True)
```

Fit di Likelihood in ROOT

ROOT ha già disponibili molte delle tipologie di fit che abbiamo visto

- Binned Likelihood. Implementata tramite opzione "MULTI" nel metodo Fit per istogrammi
- Binned Extended Likelihood. Implementata tramite opzione "L" nel metodo Fit per istogrammi
- Unbinned Likelihood. Per questo ovviamente serve un contenitore diverso dall'istogramma.
  La classe TTree permette di descrivere più variabili per un singolo dato (senza alcun "impacchettamento"). Ecco alcuni metodi utili

```
TTree()
ReadFile(const char *filename, const char *var_descr="")
int TTree::GetEntries()
int TTree::GetEntry(int i)
TTree::UnbinnedFit (const char * funcname, const char * varexp)
```

Tree

- `TTree()`
  Costruttore
- `ReadFile (const char *filename, const char *var_descr="")`
  legge da file secondo l'espressione "var1:var2:var3...", è possibile specificare i formati: F(float)/D(double)/I(int)/C(string) (F default) come "var1/D:var2/D:var3/D..."
- `int TTree::GetEntries()`
  ritorna il numero di dati
- `int TTree::GetEntry(int i)`
  carica in memoria il dato i-esimo che da quel momento sara' accessibile con nomeOggettoTree.var
- `TTree::UnbinnedFit (const char * funcname, const char * var)`
  esegue un Unbinned Fit con la funzione funcname (TF1) sulla variabile var

Esempio

```
1 from   ROOT    import *
2
3 t = TTree();
4 t.ReadFile("exp.dat","t/D")
5 for i in range(0,t.GetEntries()):
6     t.GetEntry(i)
7     print(t.t)
```

Fit di Likelihood Unbinned (ROOT)

```
1  void fitexp_ROOT(){
2
3    TTree tree;
4    tree.ReadFile("exp.dat","t");
5
6    TF1 f("f","1/[0]*exp(-x/[0])",0,10);
7    f.SetParameter(0,1);
8    tree.UnbinnedFit("f","t");
9
10 }
```

Fit di Likelihood Binned (ROOT)

```
1  from   ROOT    import *
2  from   iminuit import Minuit
3  import numpy   as     np
4  from   math    import *
5
6  #Main
7
8  h  = TH1D("h","",20,0,10)
9  for line in open("exp.dat"):
10     h.Fill(float(line))
11
12 h.Draw("E")
13
14 f = TF1("f","[1]/[0]*exp(-x/[0])",0,10)
15 f.SetParameter(0,1)
16 f.FixParameter(1,1)
17
18 h.Fit("f","OMULTI")
19 f.SetParameter(1,h.GetEntries()*h.GetBinWidth(1))
20 f.Draw("SAME")
21
22 gApplication.Run(True)
```

Fit di Likelihood Binned+Extended (ROOT)

```
1  from   ROOT    import *
2  from   iminuit import Minuit
3  import numpy   as     np
4  from   math    import *
5
6  #Main
7
8  h  = TH1D("h","",20,0,10)
9  for line in open("exp.dat"):
10     h.Fill(float(line))
11
12 h.Draw("E")
13
14 f = TF1("f","[1]/[0]*exp(-x/[0])",0,10)
15 f.SetParameter(0,1)
16 f.SetParameter(1,h.GetEntries()*h.GetBinWidth(1))
17 print(h.GetEntries()*h.GetBinWidth(1))
18 h.Fit("f","L")
19
20 gApplication.Run(True)
```