

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

Laboratorio di Metodi Computazionali e Statistici (2022/2023)

R. Cardinale, F. Parodi, S. Passaggio

November 30, 2022

Fit di Likelihood

Unbinned Maximum Likelihood (N eventi)

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

Binned Maximum Likelihood (M bins)

$$P(n_1, n_2, \dots, n_M | p_1, p_2, \dots, p_M) = \frac{N!}{n_1! n_2! \dots n_M!} p_1^{n_1} p_2^{n_2} \dots p_M^{n_M}$$
$$-\ln(\mathcal{L}(P)) = -\sum_{i=1}^M n_i \ln(p_i) + \text{const}$$

Binned Extended Maximum Likelihood (M bins)

$$P(n_1, n_2, \dots, n_M | p_1, p_2, \dots, p_M) = \prod_{i=1}^M \frac{e^{-\mu_i} \mu_i^{n_i}}{n_i!} \qquad \mu_i = N p_i$$
$$-\ln(\mathcal{L}(P)) = -\sum_{i=1}^M (n_i \ln(\mu_i) - \mu_i)$$

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 unbinned
- Likelihood binned
- Likelihood binned (extended)

sia implementando la funzione $-\ln(\mathcal{L})$ sia utilizzando metodi di ROOT

Fit di Likelihood in ROOT

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

- 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)
```

- Binned Likelihood. Implementata tramite opzione “MULTI” nel metodo Fit per istogrammi
- Binned Extended Likelihood. Implementata tramite opzione “L” nel metodo Fit per istogrammi

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 sarà 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 tree = TTree();
4 tree.ReadFile("exp.dat","t/D")
5 for i in range(0,tree.GetEntries()):
6     tree.GetEntry(i)
7     print(tree.t)
```

Riassunto metodi ROOT per fit di likelihood

	Binned	Unbinned
Likelihood	Fit + opzione MULTI	TTree + UnbinnedFit
Extended Likelihood	Fit + opzione L	