20 Dicembre 2022

Esercitazione n. 3
Metodologie di Analisi Dati

Esercizio 1

Questo esercizio fornisce un'introduzione alla classe `TMinuit`, utilizzata in ROOT per minimizzare una funzione. Dovresti convertire qualsiasi codice in cui hai apportato modifiche e l'output del programma sotto forma di valori numerici e grafici appropriati. Non è necessario convertire il codice fornito all'utente che non è stato modificato.

Gli esercizi utilizzano `TMinuit` per eseguire un fit di Maximum Likelihood dove si cerca il minimo della quantità -2 ln *L*. Per ulteriori informazioni su `TMinuit` vedere:

`root.cern.ch/root/html/TMinuit.html`

Innanzitutto genereremo alcuni dati utilizzando un semplice programma Monte Carlo. Scarica, crea e esegui il programma `makeData` dal link:

https://cernbox.cern.ch/s/NbhmqwIPTfcryFe

`makeData` genera valori in base a una distribuzione esponenziale:

$$f(x;\xi) = \frac{1}{\xi} e^{-x/\xi} \qquad (x \geq 0)$$

e scrive i valori in un file. In una directory separata, scarica e compila il programma `expFit` che puoi trovare al link:

https://cernbox.cern.ch/s/aEweqbQoYrJUXwE

Questo programma legge nel file dei singoli valori forniti da `makeData` e fa un fit di Maximum Likelihood del parametro ξ della pdf esponenziale. Esegui `makeData` e genera un file con 200 valori. Usa questo come input per `expFit` e trova la stima $\hat{\xi}$ e la sua deviazione standard $\sigma_{\hat{\xi}}$.

Ora modifica `makeData` in modo che generi valori in base alla pdf

$$f(x;\alpha,\xi_1,\xi_2) = \alpha \frac{1}{\xi_1} e^{-x/\xi_1} + (1-\alpha)\frac{1}{\xi_2} e^{-x/\xi_2} \qquad (x \geq 0)$$

con $\alpha = 0.2$, $\xi_1 = 1.0$ e $\xi_2 = 5.0$. Per fare ciò, prima genera un numero casuale *r* uniforme in [0,1]. Se $r < \alpha$, allora genera x secondo un esponenziale con la media ξ_1, altrimenti usa ξ_2. Esegui il programma e salva 200 singoli valori in un file di testo.

Ora modifica il programma `expFit` in modo che legga i valori ed esegua un fit di ML dei parametri α, ξ_1 e ξ_2. Dovrai fornire i valori iniziali e le "dimensioni del passo" per i parametri. Scegli i valori iniziali non troppo distanti (diciamo, entro un fattore di due) dai valori veri usati in `makeData`. Per le dimensioni del passo che puoi assumere, ad es. 0.1.

Prova a eseguire il programma con i valori minimo e massimo (negli array `minVal` e `maxVal`) impostati uguali a zero; questo è equivalente a non avere limiti sui parametri. Se il fit viene eseguito in una regione dello spazio dei parametri non consentito, ad esempio $\xi_1 < 0$, allora è possibile posizionare limiti appropriati sui valori dei parametri. Alla fine è meglio vedere se è possibile eseguire il fit con ipotesi migliorate per i valori iniziali ma senza alcun limite sui parametri.

Modificare il programma in modo da ottenere un grafico ragionevole del fit (estendere il limite dell'asse orizzontale come appropriato). Trova gli stimatori di ML e la loro matrice di covarianza usando le routine `mnpout` e `mnemat`. Ciò richiede che tu aggiunga linee del tipo

```
double covmat [npar] [npar];
minuit.mnemat (& covmat [0] [0], npar);
```

dove `npar` (qui 3) è il numero di parametri adattati. Determina anche la matrice dei coefficienti di correlazione.

Eseguire un test di χ^2 per valutare la bontà del fit.

Esercizio 2

Per questo problema, si prega di trasformare i grafici e tutti i calcoli rilevanti, e in aggiunta solo quelle parti del codice in cui sono state apportate modifiche (ad es. Propagazione degli errori e il calcolo dei valori χ^2e i p-values).

Il problema utilizza la macro root `SimpleFit.C` e il file di dati `testData.txt` dal link:

https://cernbox.cern.ch/s/NOyaiJmYWLiTzQq

Il codice è fondamentalmente scritto in C++, ma viene eseguito tramite il programma ROOT anziché essere eseguito come un programma indipendente. Prima eseguire root e al prompt digitare

```
.L simpleFit.C
simpleFit()
```

Il primo comando carica il contenuto del file `simpleFit.C` e quindi definisce le funzioni in esso contenute. Il secondo comando chiama la funzione `simpleFit()`. Questo richiede all'utente un file di dati contenente colonne di numeri che rappresentano qui i soliti ingredienti di un fit dei minimi quadrati x, y e σ.

La macro contiene una funzione di fit, che è definita come un polinomio,

$$f(x; \boldsymbol{\theta}) = \sum_{k=0}^{n} \theta_k x^k$$

Dopo aver letto i dati, i parametri del polinomio sono fittati usando il metodo dei minimi quadrati. I risultati vengono estratti e visualizzati, inclusi i valori dei parametri fittati, le loro deviazioni standard, il minimo χ^2, il p-value corrispondente, la matrice di covarianza $V_{ij} = \text{cov}[\hat{\theta}_i, \hat{\theta}_j]$ e la sua matrice inversa.

1 (a) Provare diversi ordini per il polinomio cambiando la variabile `npar` (cioè, per ottenere un ordine *n* = `npar - 1`). Qual è l'ordine più piccolo che dà un p-value maggiore di 0.1? Produrre il plot del risultato del fit con *n* = 2, 3, 4.

1 (b) Considera i casi di ordine n = 2, 3 e 4 (cioè con 3, 4 e 5 parametri). Utilizzando il polinomio insieme ai parametri corretti, trovare il valore previsto della funzione in x = 5, x = 6 e x = 10. Il programma contiene un oggetto chiamato `f` di tipo `TF1*` che è possibile utilizzare per accedere alla funzione di fit . Avrai bisogno (in un ciclo sopra i parametri)

```
f-> SetParameter (i, thetaHat [i]);
```

per impostare i valori dei parametri su quelli adattati e

```
f->eval([x [i]);
```

per valutare la funzione nel punto `x[i]`. Si prega di fare riferimento alla documentazione di root per ulteriori informazioni e pdettagli.

Usando la propagazione dell'errore, trova la deviazione standard della differenza

$$d_{ab} = f(x_a; \hat{\boldsymbol{\theta}}) - f(x_b; \hat{\boldsymbol{\theta}})$$

Implementa la formula che trovi per la deviazione standard nella macro root e trova numericamente i valori per (i) $x_a = 6, x_b = 6$ e (ii) $x_a = 10, x_b = 5$. Commenta come uno si aspetta che la deviazione standard della differenza si comporti quando la differenza tra i valori x diminuisce.

1 (c) Supponiamo per il caso di ordine *n* = 3, un determinato modello prevede i valori dei parametri: $\theta_0 = -0.75, \theta_1 = 2.5, \theta_2 = -0.5$ e $\theta_3 = 0.026$. Usando l'inverso della matrice di covarianza trovata nella macro (variabile `Vinv`), calcola il valore del χ^2 confrontando i tuoi valori fittati con le previsioni del modello,

$$\chi^2 = \sum_{i,j=0}^{n} (\theta_{\text{mod},i} - \hat{\theta}_i)(V^{-1})_{ij}(\theta_{\text{mod},j} - \hat{\theta}_j)$$

Si noti che qui gli stimatori $\hat{\boldsymbol{\theta}}$ sono trattati come un insieme di 4 quantità misurate e sono confrontati con 4 predizioni del modello. Trova il valore p-value corrispondente. Il modello è in accordo accettabile con i valori dei parametri stimati? Verifica la tua conclusione facendo un grafico della previsione del modello per $f(x)$ insieme ai dati.