UNIVERSITÀ DEGLI STUDI DI MILANO

FACOLTÀ DI SCIENZE E TECNOLOGIE

Analisi Dati

Autore

Riccardo Loddo

Anno accademico 2022/23

Indice

Introduzione **3**

Problema 1 **4**

Problema 2 **8**

Problema 3 **12**

Problema 4 **19**

Problema 5 **22**

Problema 6 **27**

Introduzione

Nel presente lavoro vengono esposti sei esercizi di natura diversa, tramite l'utilizzo di tecniche statistiche. Per raggiungere tali obiettivi, sono stati scritti programmi in linguaggio C++ e Python, opportunamente progettati per eseguire le diverse elaborazioni richieste. In particolare, il linguaggio C++ è stato utilizzato per i primi quattro esercizi, mentre Python è stato utilizzato per gli ultimi due.
Le analisi effettuate hanno riguardato la determinazione di parametri statistici, come la media, la deviazione standard, la varianza, e la stima di distribuzioni di probabilità. Inoltre, l'analisi è stata arricchita dall'utilizzo di tecniche di machine learning, in particolare reti neurali, che hanno permesso di analizzare dati complessi e di apprendere le relazioni tra le variabili in esame. Infine sono state applicate tecniche come il metodo dei minimi quadrati e la Maximum Likelihood.
Tutti i programmi sviluppati sono disponibili sul (mio sito), nella sezione "*Analisi Dati*" per garantire la massima trasparenza e riproducibilità dei risultati ottenuti.

Problema 1

Esercizio 1

Supponiamo che le variabili casuali indipendenti r_i siano uniformemente distribuite tra [0;1]. Come punto di partenza, è stato scritto un programma per creare istogrammi di:

(a) $x = r_1 + r_2 - 1$

(b) $x = r_1 + r_2 + r_3 + r_4 - 2$

(c) $x = \sum_{i=1}^{12} r_i - 6.$

Gli istogrammi sono stati creati prendendo 12 numeri casuali e aggiornando di conseguenza le variabili somma. Ad esempio, la variabile somma per (c) verrebbe aggiornata con i valori di tutti i numeri divisibili per 3 e la variabile somma per (b) verrebbe aggiornata con i valori di tutti i numeri divisibili per 6. Questo processo è stato fatto 10000 volte. Successivamente, è stato sottratto l'intero richiesto ed sono stati raffigurati gli istogrammi. In Figura (1) sono mostrati i tre istogrammi richiesti.

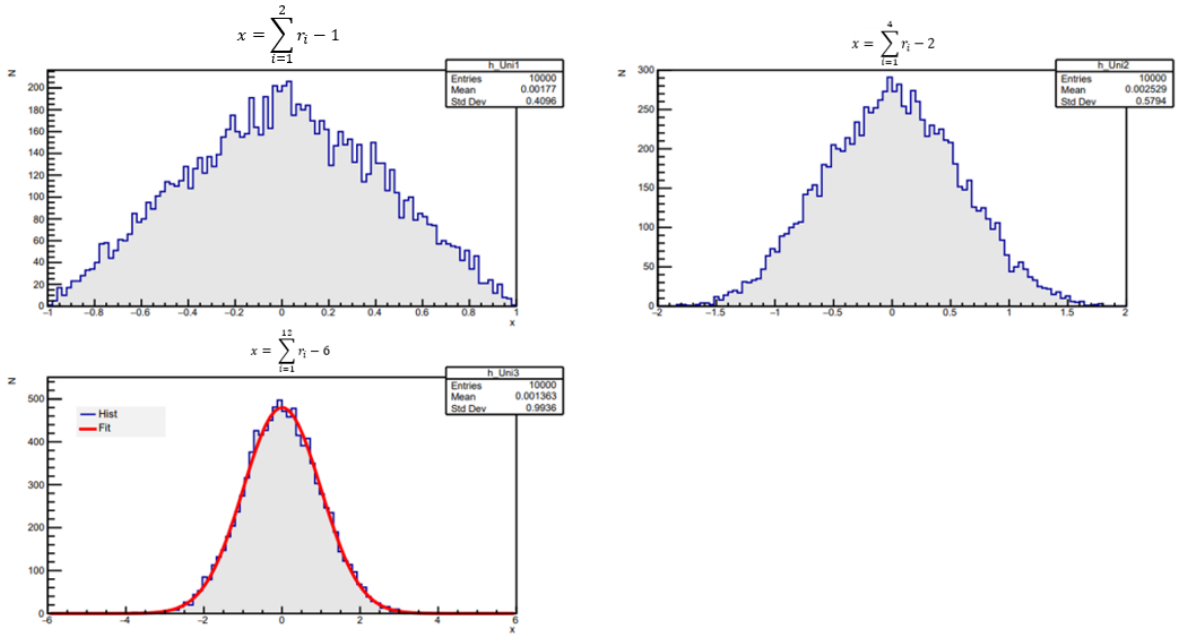

Figura 1: Istogrammi della variabile x relativa ai tre diversi casi.

L'esercizio prosegue con la richiesta di calcolare esattamente le medie e le varianze delle variabili definite in (a) - (c) e confrontale con i valori ottenuti dagli istogrammi dei numeri generati. Sapendo che $\mathrm{E}[r_i] = \frac{1}{2}$, è chiaro che per ognuno dei tre casi la media di x è $\mathrm{E}[x] = 0$. Analogamente, sapendo che la varianza della generica variabile random r_i è:

$$\sigma_{r_i}^2 = E[r_i^2] - (E[r_i])^2 = \int_{-\infty}^{+\infty} r_i^2 f(r_i) dr_i - \left(\int_{-\infty}^{+\infty} r_i f(r_i) dr_i \right)^2 = \frac{1}{12}, \tag{1}$$

e che le r_i sono variabili casuali ed indipendenti, è chiaro che la varianza di x si ricava come somma delle varianze di r_i, ovvero:

$$\sigma_x^2 = \sum_i \sigma_{r_i}^2. \tag{2}$$

Nel caso in questione si ha che:

(a) $\sigma_x^2 = \frac{1}{6}$

(b) $\sigma_x^2 = \frac{1}{3}$

(c) $\sigma_x^2 = 1$.

In Tabella (1) sono confrontati i valori esatti con quelli ottenuti dagli istogrammi.

Confronto dati				
Caso	$E_{vero}[x]$	$V_{vero}[x]$	$E_{num}[x]$	$V_{num}[x]$
(a)	0	0.166	0.0017	0,1677
(b)	0	0.333	0.0025	0,3357
(c)	0	1	0.0013	0,9872

Tabella 1: Confronto tra valori analitici e valori numerici.

Ovviamente, più vengono creati valori di x, più la stima numerica è accurata. Ciò nonostante, anche con solo 10000 valori, la differenza tra valori analitici è numerici è minima. La questione che può saltare subito all'occhio è chiedersi perché gli istogrammi hanno quella forma. Ciò che ci sta dietro è il Teorema del Limite Centrale. Esso afferma che, per una grande quantità di campioni casuali indipendenti prelevati da una popolazione con una distribuzione finita della media e della varianza (quale è il nostro caso), la distribuzione campionaria della media si avvicina alla distribuzione normale al crescere della dimensione del campione.
In altre parole, indipendentemente dalla forma della distribuzione della popolazione, la media campionaria si avvicina sempre di più alla distribuzione normale al crescere del numero di campioni estratti. Ciò viene confermato dal Fit gaussiano presente nell'istogramma (c).

Esercizio 2

Considerando una PDF a "*dente di sega*",

$$f(x) = \begin{cases} \frac{2x}{x_{max}^2}, & 0 < x < x_{max} \\ 0, & \text{altrimenti} \end{cases} \tag{3}$$

L'esercizio propone di generare numeri casuali secondo la $f(x)$, dapprima tramite il metodo della trasformata inversa e poi tramite il metodo "*accept-reject*". Il metodo della trasformata inversa è un'operazione matematica che permette di generare una distribuzione di probabilità a partire dalla sua funzione di densità di probabilità (PDF). Per trovare la sua funzione di distribuzione cumulativa (CDF), si deve integrare la PDF:

$$F(x) = \int_{-\infty}^{x} f(x')dx' = \frac{x^2}{x_{max}}, \quad \text{se} \quad 0 < x < x_{max}. \tag{4}$$

Per trovare la trasformata inversa, si deve risolvere l'equazione F(x) = u, dove u è una variabile casuale uniforme tra 0 e 1, da cui si ricava che:

$$x = \sqrt{u} \cdot x_{max}. \tag{5}$$

In Figura (2) è mostrato l'istogramma con il metodo della trasformata inversa.

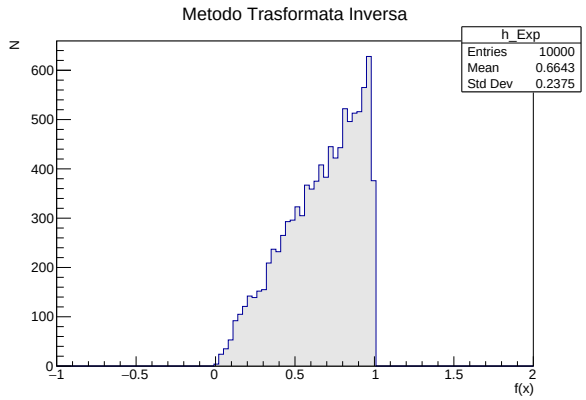

Figura 2: Trasformata inversa.

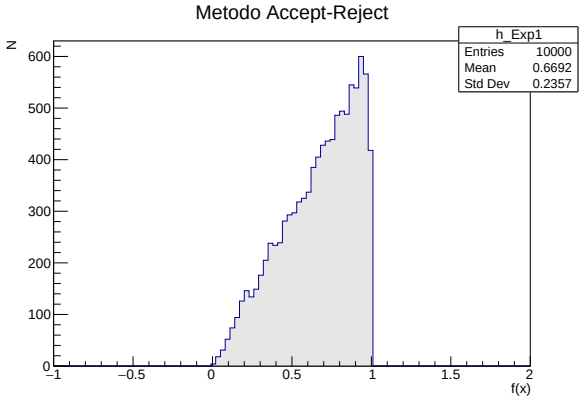

Figura 3: "*Accept-Reject*".

Con lo stesso obbiettivo di prima è stato fatto un programma per generare numeri casuali secondo la PDF a dente di sega usando il metodo "*accept-reject*". Per generare numeri casuali secondo la PDF a dente di sega $f(x)$, il metodo "*accept-reject*" può essere utilizzato ed impostato come segue:

- Trovare una costante M tale che la PDF sia sempre inferiore a M. In questo caso, la PDF ha un massimo di $\frac{2}{x_{max}}$, quindi si è scelto M = $\frac{2}{x_{max}}$.
- Generare un valore casuale s compreso tra 0 ed x_{max} dalla distribuzione uniforme.

- Generare un valore casuale t tra 0 e M dalla distribuzione uniforme.
- Sia $y = M \cdot t$, se $y \leq f(s)$ allora lo si accetta, se no lo si scarta.
- Ripetere la procedura n volte.

Per conformità rispetto al caso precedente, il valore di n è 10000. Per questioni di generalità sul codice è stata creata una classe "*RandomGen*" implementando una funzione (AR) che descrive il metodo "*accept-reject*".
In Figura (3) è presente l'istogramma con il metodo "*accept-reject*". Si noti l'estrema somiglianza con il metodo della trasformata inversa. Lo svantaggio del metodo "*accept-reject*" è che dipende fortemente dalla PDF che si ha. Anzitutto non è assolutamente detto di poter trovare il massimo di una funzione in modo semplice. Dobbiamo inoltre assicurarci che sia un massimo assoluto e non locale, ecc. Inoltre con tale metodo si scartano molti dei punti generati e questo comporta (in generale) ad una scarsa efficienza del metodo. Ad ogni modo, il risultato è mostrato in Figura (3). Come possiamo notare i metodi coincidono, anche se, per ovvie ragioni, la media e la deviazione standard sono tra di loro leggermente diverse.
L'efficienza del metodo accept-reject in generale dipende da vari fattori come la distribuzione di probabilità target, la distribuzione di probabilità proposta e il numero di iterazioni necessarie per generare un campione accettabile. Per valutarne l'efficienza, è possibile utilizzare diverse metriche come il tasso di accettazione, il tempo di esecuzione e la qualità dei campioni generati. In questo caso è stato utilizzata la metrica del tasso di accettazione. Il tasso di accettazione ("*acceptance rate*") è la percentuale di campioni generati che soddisfano i criteri di accettazione stabiliti dall'algoritmo accept-reject. In altre parole, rappresenta la frazione di campioni generati che sono stati accettati e utilizzati per costruire l'istogramma o la stima della distribuzione di probabilità target.
Da un punto di vista computazionale è molto semplice in quanto basta contare il numero di volte che l'algoritmo itera il metodo. Nel caso in questione si sono trovati $n_{tot} = 19724$ iterazioni totali su $n_{hit} = 10000$ punti richiesti. Tale metodo presenta un'efficienza di circa:

$$\eta = \frac{n_{hit}}{n_{tot}} \approx 0.507 \tag{6}$$

ovvero solo la metà dei punti che vengono generati rispetta i criteri stabiliti.

Problema 2

Esercizio 1

L'esercizio prevede l'esecuzione di una semplice analisi multivariata utilizzando il pacchetto TMVA. L'idea è quella di svolgere un' analisi di classificazione segnale-fondo con dei dati forniti dal problema. L'analisi multivariata consiste nell'analizzare insieme più variabili per identificare relazioni e tendenze tra di esse. Anzitutto sono state generate due n-tuple di dati con 10.000 eventi ciascuno i cui valori seguono una determinata distribuzione tridimensionale per l'ipotesi del segnale e un'altra per l'ipotesi di fondo. Viene indicata con n_s il numero di eventi di segnali totali ed n_b il numero di eventi di fondo. In questo caso corrispondono entrambi a 10000. Inizialmente, è necessario allenare e verificare le prestazioni di un classificatore lineare chiamato discriminante di Fisher. Dopo l'allenamento, il software genera un file denominato "*weights*" nella directory, che contiene i pesi ideali per massimizzare la separazione tra le distribuzioni del segnale e del rumore di fondo. Il discriminante di Fisher è definito in tal modo perché mira a massimizzare la distanza tra queste due distribuzioni. Il discriminante lineare di Fisher può essere utilizzato come classificatore di apprendimento supervisionato. Avendo i dati etichettati, il classificatore può trovare un insieme di pesi per tracciare un limite decisionale, classificando i dati. Il discriminante lineare di Fisher tenta di trovare il vettore che massimizza la separazione tra le classi (segnale-fondo) dei dati proiettati. Massimizzare la " separazione" può essere ambiguo. I criteri seguiti dal discriminante lineare di Fisher per fare ciò sono massimizzare la distanza delle medie proiettate e minimizzare la varianza prevista all'interno della classe. Il problema richiede di selezionare eventi di segnale con $t_{Fisher} > 0$ e di trovare l'efficienza del segnale e del fondo. Inoltre richiede di calcolare la purezza del segnale supponendo che le probabilità a priori di ottenere fondo e segnale siano identiche. L'Efficienza del segnale è definita come:

$$\epsilon_s^F = P(t_{Fisher} > 0|s) = \frac{N_s}{n_s} = 0.8245, \tag{7}$$

dove N_s sono gli eventi di segnale che sono maggiori di $t_{Fisher} > 0$. Chiaramente il pedice F indica il tipo di metodo utilizzato. In questo caso Fisher. Analogamente, gli eventi di fondo hanno efficienza:

$$\epsilon_b^F = P(t_{Fisher} > 0|b) = \frac{N_b}{n_b} = 0.1912, \tag{8}$$

dove N_b sono gli eventi di fondo che sono maggiori di $t_{Fisher} > 0$. La purezza del segnale è definita come:

$$p_s^F = P(s|t_{Fisher} > 0) = \frac{N_s}{N_s + N_b} = 0.8117 \tag{9}$$

Successivamente sono stati creati degli istogrammi di t_{Fisher} per entrambi gli eventi di segnale e di fondo. In Figura (2) è mostrato il grafico relativo al metodo di Fisher.

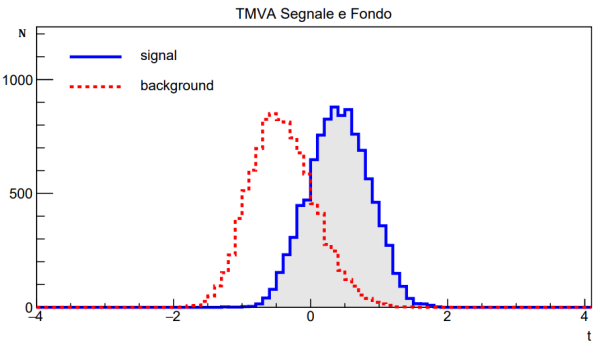

Figura 2: Fisher.

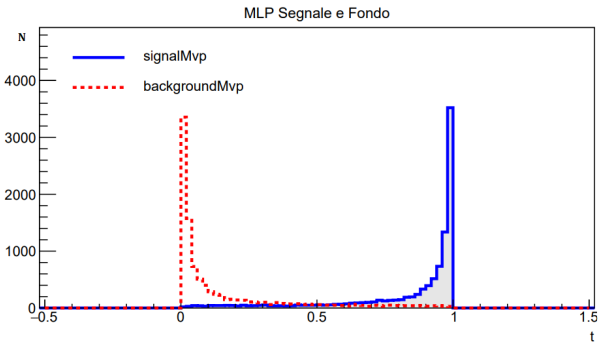

Figura 3: MLP.

Notiamo come il discriminante di Fisher è un buon classificatore perché può distinguere tra diversi gruppi di segnale. In particolare è stato isolato l'82% degli eventi di segnale per $t > 0$ ed all'interno di tale zona sono presenti il 19% degli eventi di beckground.
Nella seconda parte dell'esercizio si richiede di implementare un classificatore non lineare ovvero il Multi Layer Perceptron (MPL). Il Multi Layer Perceptron è una rete neurale artificiale che consiste in una serie di strati di neuroni, dove ogni neurone è collegato a tutti i neuroni dello strato successivo. Questi strati intermedi sono chiamati strati nascosti, e il numero di neuroni in ogni strato può variare a seconda del problema da risolvere. Nel nostro caso si hanno 3 nodi. Questo classificatore va a minimizzare una *loss function*[1] opportunamente definita e produce come nel caso precedente, un insieme di pesi salvati in un file. Quindi, una volta implementata la MLP nella fase di allenamento, il procedimento è praticamente analogo a quello di Fisher. Tuttavia, l'unica differenza rispetto al caso di prima è che ora si richiede un $t_{MLP} > 0.5$. Sempre in maniera analoga a prima (e con la stessa notazione di prima) sono state calcolate le efficienze e la purezza ottenendo quindi:

$$\epsilon_s^{MLP} = P(t_{MLP} > 0.5|s) = \frac{N_s}{n_s} = 0.8971, \tag{10}$$

$$\epsilon_b^{MLP} = P(t_{MLP} > 0|b) = \frac{N_b}{n_b} = 0.1069, \tag{11}$$

$$p_s^{MLP} = P(s|t_{MLP} > 0) = \frac{N_s}{n_s + (N_s - N_b)} = 0.8935. \tag{12}$$

[1]La *loss function* rappresenta una misura dell'errore tra l'output prodotto dalla rete neurale e l'output desiderato, ovvero la differenza tra la previsione della rete e la risposta corretta. L'obiettivo dell'addestramento del MLP è quello di minimizzare la funzione di perdita, ovvero di ridurre l'errore tra l'output previsto e quello desiderato, attraverso l'aggiustamento dei pesi dei collegamenti tra i neuroni.

Come si può notarle si ha una netta differenza rispetto al caso di Fisher. Infatti, dalla Figura (3), gli eventi di segnale e di fondo sono nettamente separati con il metodo MLP. Concludendo, l'analisi di Fisher e il Multi Layer Perceptron (MLP) sono due tecniche molto diverse e non possono essere direttamente confrontate in termini di "meglio" o "peggio". Dipende dal tipo di problema che si sta affrontando e dalle informazioni disponibili sui dati. In generale, se si dispone di un grande numero di dati di training e si vuole risolvere un problema di classificazione o di regressione complesso, il MLP potrebbe essere una scelta migliore rispetto all'analisi di Fisher. Tuttavia, se si hanno pochi dati di training e si vuole eseguire una classificazione o una riduzione della dimensionalità, l'analisi di Fisher potrebbe essere più adatta.

Esercizio 2

Questo esercizio è una continuazione di esercizio precedente e prevede di apportare un'ulteriore modifica (sia in fase di allenamento che in fase di analisi) implementando il classificatore Boosted Decision Tree (BDT). Il Boosted Decision Tree (BDT) è un algoritmo di apprendimento automatico utilizzato per la classificazione e la regressione. Si basa su una combinazione di alberi decisionali (decision tree) deboli, che sono migliorati (boosted) per aumentare la precisione della previsione finale. In questo caso sono stati utilizzati 200 alberi (Tree). Ancora una volta il procedimento è praticamente analogo a quello di prima e viene richiesto di creare degli istogrammi (mostrando le distribuzioni di segnale e di fondo sullo stesso grafico) sia per il campione di training che per il campione di test, statisticamente indipendente. Tali istogrammi vengono riportati in Figura (2) ed (3). In

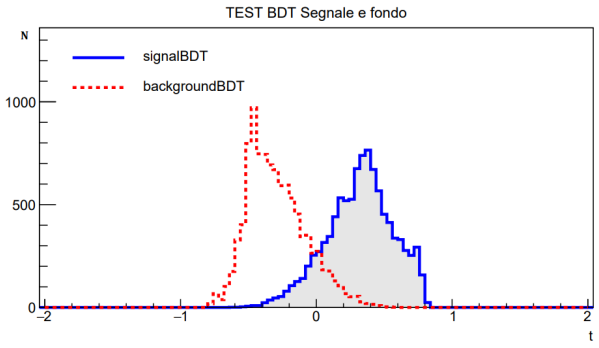

Figura 2: Test BDT.

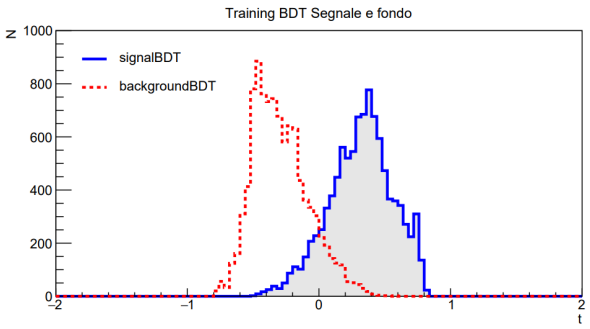

Figura 3: Training BDT.

questo caso viene preso $t_{BDT} > 0$. Infine, l'esercizio propone di ripetere questa procedura per diversi numeri di iterazioni di boosting, in particolare 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 10000, 50000. Chiaramente è facile generalizzare il processo a n iterazioni di boosting creando opportunamente un ciclo. Per ogni classificatore, è stato calcolato il tasso di errore totale utilizzando un valore limite di $t_{cut} = 0$. In altre parole, è stata calcolata la frazione di eventi (segnale e fondo) che si trovano sul "*lato sbagliato*" del limite. Successivamente è stata riportata in un plot tale frazione in funzione del numero di

iterazioni di boosting, sia per il campione di training che per il campione statisticamente indipendente di test. In Figura (2) è rappresentato l'andamento di tale grafico.

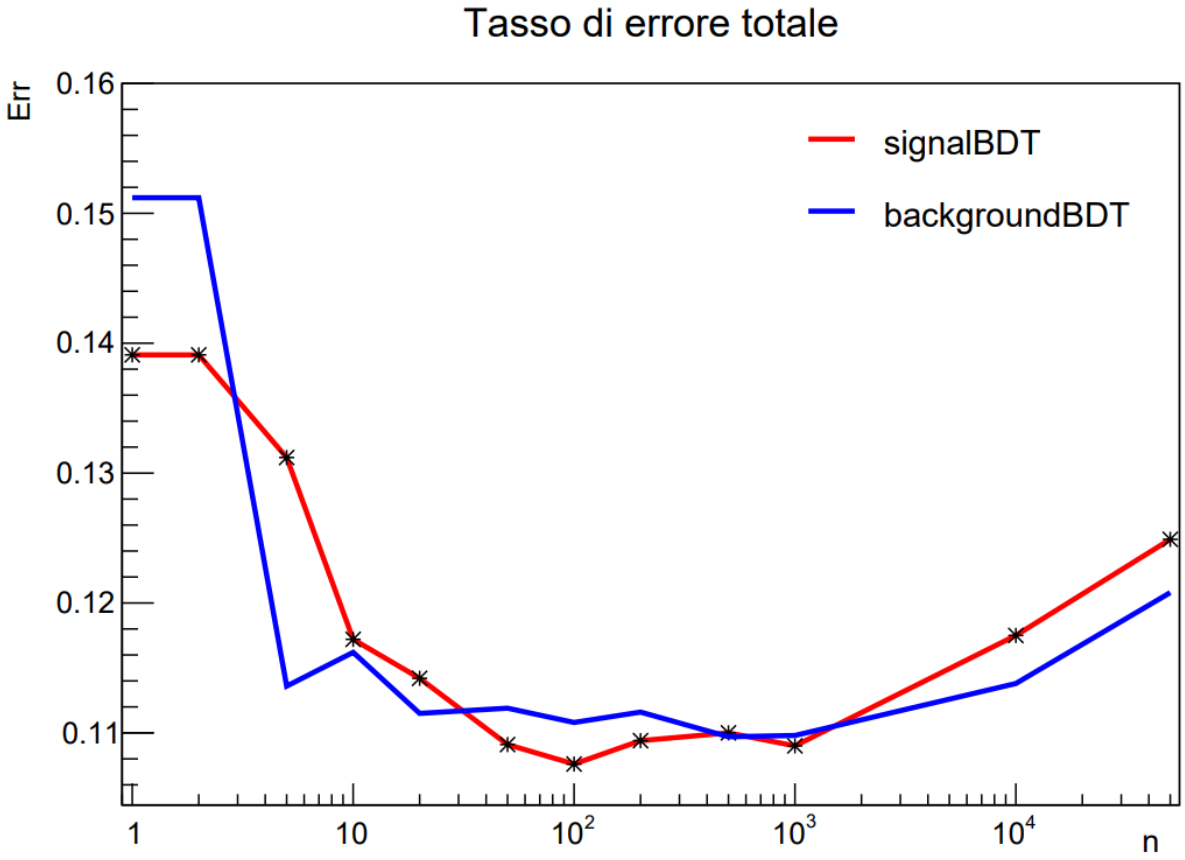

Figura 2: Tasso di errore in funzione del numero di iterazioni n.

Da ciò è stato possibile determinare approssimativamente il numero ottimale di iterazioni di boosting semplicemente osservando il minimo della curva. Il numero di iterazioni di boosting è approssimativamente:

$$N_{int} \approx 100. \tag{13}$$

Problema 3

Esercizio 1

Questo esercizio propone un'introduzione alla classe TMinuit, utilizzata in ROOT per minimizzare una funzione. La classe TMinuit è un'implementazione del metodo di minimizzazione della funzione di likelihood (o dei minimi quadrati) che permette di trovare i valori dei parametri di una funzione che minimizzano il risultato di una determinata funzione di costo.
Anzitutto viene fornito un programma (chiamato "*makeData*") in grado di generare valori in base a una distribuzione esponenziale:

$$f(x;\xi) = \frac{1}{\xi} e^{-\frac{x}{\xi}}. \tag{14}$$

Viene poi fornito un secondo programma (chiamato "*expFit*") in grado di legge nel file dei singoli valori forniti dal *makeData* creando un Fit di Maximum Likelihood del parametro ξ della PDF esponenziale. Eseguendo il programma viene generato un file con 200 valori. Usando tale file come input per *expFit* si è trovata la stima e la deviazione standard di ξ. Il risultato è riportato in Tabella (2).
L'esercizio prosegue modificando la PDF come:

$$f(x;\alpha,\xi_1,\xi_2) = \alpha \frac{1}{\xi_1} e^{-\frac{x}{\xi_1}} + (1-\alpha)\frac{1}{\xi_2} e^{-\frac{x}{\xi_2}} \tag{15}$$

con $\alpha = 0.2$, $\xi_1 = 1.0$ e $\xi_2 = 5.0$. L'idea è di creare ancora 200 valori secondo quest'ultima PDF. Per farlo, basta generare un numero casuale t distribuito in modo uniforme tra 0 ed 1. Successivamente se $t < \alpha$ allora genero i dati secondo un esponenziale con media ξ_1 altrimenti un esponenziale con media ξ_2. Computazionalmente è molto semplice in quanto basta ricondursi ad una sequenza del costrutto "*if-else*" inserito all'interno di un ciclo.
Una volta generati i dati secondo la PDF (15) è stato eseguito un Fit di Maximum Likelihood. Le modifiche rispetto al codice precedente sono evidenti. Ora siamo interessati ad estrarre una stima dei tre parametri α, ξ_1 e ξ_2, quindi l'implementazione del Fit è stata fatta con 3 parametri. Questo passaggio è cruciale poiché non conosciamo (in linea di principio) i valori veri dei parametri da trovare. Tuttavia, se il nostro parametro è ad esempio la massa, sappiamo che il più grande intervallo di ricerca è quello che parte da 0 fino ad infinito. Non ha senso cercare una massa negativa. L'idea è quindi di rimanere

il più generici possibili nella scelta dei parametri iniziali. Inoltre, bisogna fornire una condizione iniziale per i tre parametri (tendenzialmente il più realistica possibile ai valori veri) ed un intervallo di ricerca di tali parametri.
I valori ottenuti, sono riportati in Tabella (2):

Parametro	**Valore vero**	**Stimatore** $\hat{\xi}$	**Dev std Stimatore** $\sigma_{\hat{\xi}}$
ξ	1.0	0.904	0.063
ξ_1	1.0	0.925	0.309
ξ_2	5.0	5.175	0.599
α	0.2	0.2686	0.090

Tabella 2: Valori ottenuti tramite il FitML.

Inoltre è stato eseguito un Fit tramite la classe TMinuit che rappresenta la somma di esponenziali, ed è stato riportato in Figura (4).

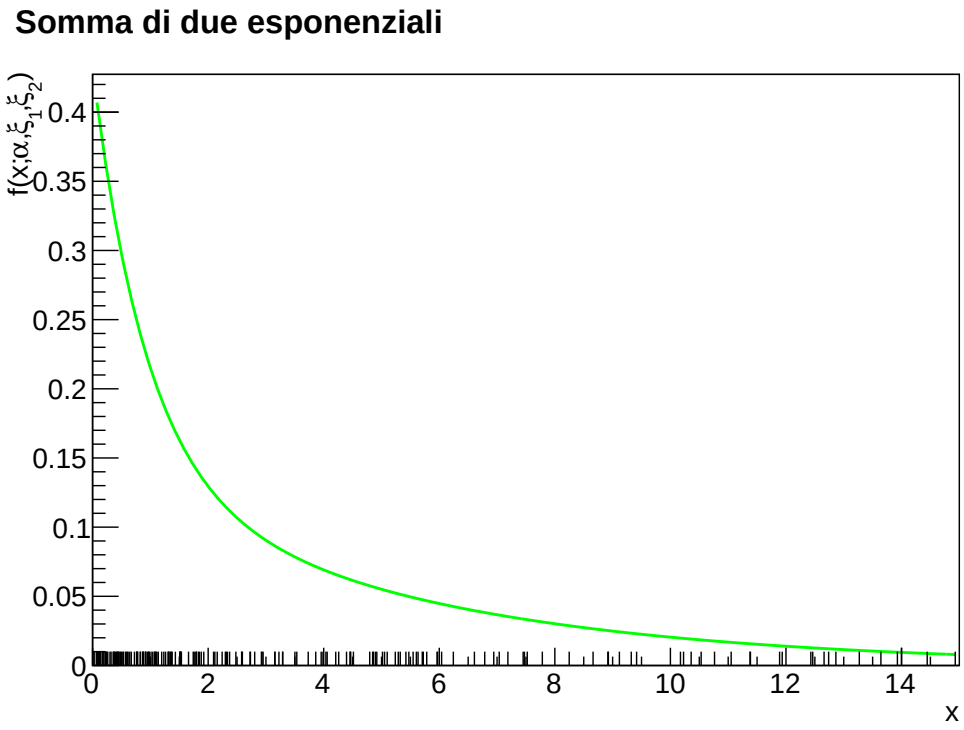

Figura 3: Fit della funzione (15) tramite la classe TMinuit.

Sono state stampate inoltre, le matrici degli errori (ovvero la matrice di covarianza e la matrice di correlazione), che vengono riportate di seguito. Ci si aspetta che esse non solo siano simmetriche (per definizione) ma che sulla diagonale vengano riportati gli stessi errori (al quadrato poiché sono varianze) trovati in Tabella (2). Effettivamente le matrici sono:

$$V_{ij} = \begin{pmatrix} 0.095 & 0.078 & 0.018 \\ 0.078 & 0.359 & 0.035 \\ 0.018 & 0.035 & 0.008 \end{pmatrix} \qquad \rho_{ij} = \begin{pmatrix} 1.000 & 0.423 & 0.660 \\ 0.423 & 1.000 & 0.649 \\ 0.660 & 0.649 & 1.000 \end{pmatrix} \tag{16}$$

Infine viene eseguito un test del chi^2 di Pearson: il test si basa sul calcolo del valore di una statistica del chi-quadro, che misura la differenza tra l'osservazione effettiva e quella attesa. Il valore della statistica del chi-quadro viene confrontato con una distribuzione di probabilità nota come distribuzione del chi-quadro, che dipende dal numero di gradi di libertà del test. Se il valore della statistica del chi-quadro è sufficientemente grande rispetto alla distribuzione del chi-quadro, si può rifiutare l'ipotesi nulla che la distribuzione di probabilità ipotizzata sia corretta. In altre parole, si può concludere che l'osservazione effettiva della variabile casuale non segue la distribuzione di probabilità ipotizzata. Il χ^2 è stato calcolato come:

$$\chi^2 = \sum_{i=0}^{M} \frac{(n_i - Np_i)^2}{\sigma_i^2}, \tag{17}$$

con M il numero di bin, n_i il numero di dati nell'i-esimo bin e σ_i l'incertezza associata al bin in questione. In questo caso si sono presi 10 bins, quindi un numero di gradi di libertà pari a 6. Tuttavia, come ho già sottolineato in precedenza, la scelta del numero di bins dipende anche dalla distribuzione dei dati. Se i dati sono fortemente asimmetrici o hanno una distribuzione non uniforme, potrebbe essere necessario utilizzare un numero maggiore o minore di bins per catturare le variazioni nella distribuzione dei dati. In ogni caso, la scelta del numero di bins è una decisione soggettiva e può variare a seconda del tipo di analisi che stai effettuando. Sono stati provati diversi numeri di bins ed è stato valutato l'impatto sulla forma e sulla significatività del risultato del chi quadrato. In particolare il χ^2 viene di:

$$\begin{gathered} \chi^2 = 4.357 \\ p-\text{valore} = 0.628 \end{gathered} \tag{18}$$

ovvero possiamo confermare che il modello rappresenta i dati, al 62.8%.

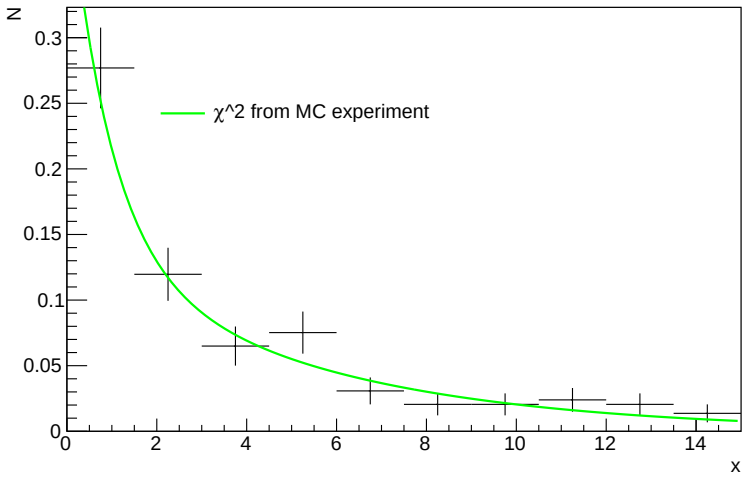

Figura 4: Il grafico rappresenta i 200 dati, spaziati in bin di 10, con associata la pdf ottenuta dal Fit.

Esercizio 2

Il codice seguente permette di trovare l'equazione dei minimi quadrati per un polinomio con grado regolabile. Il polinomio ha la forma:

$$f(x;\theta) = \sum_{k=0}^{n} \theta_k x^k. \tag{19}$$

In particolare è possibile inserire un set di dati nel programma il quale troverà l'equazione dei minimi quadrati che meglio si adatta ai dati. I dati forniti dall'esercizio sono 8 terne di numeri (x,y,σ_y). 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 $V_{i,j} = \text{cov}[\hat{\theta}_i, \hat{\theta}_j]$ e la sua matrice inversa.
Come passo preliminare l'esercizio richiede di avviare il codice cambiando l'ordine del polinomio e trovare qual'è l'ordine più piccolo che da un p-valore maggiore di 0.1. In particolare l'ordine del polinomio è n ed assume i seguenti valori: 1,2,3,4.
Per calcolare il p-valore serve il χ^2 ed il numero di gradi di libertà. I parametri ottenuti per i vari n vengono riportati in Tabella (3).

n	n_{par}	**ndof**	χ^2	p**-valore**
1	2	6	86.85	$1.36 \cdot 10^{-16}$
2	3	5	19.69	0.0014
3	**4**	**4**	**9.96**	**0.13**
4	5	3	5.05	0.16

Tabella 3: Valori del p-valore ottenuti tramite il Fit, per diversi ordini del polinomio.

Ormai è chiaro che il polinomio di ordine più piccolo affinché il p-valore sia maggiore di 0.1 è un polinomio di terzo grado. In Figura (5) vengono riportati i quattro grafici relativi ai diversi gradi del polinomio.
È stato deciso di inserire anche il polinomio con $n = 1$ anche se non richiesto dall'esercizio, per questioni di simmetria dell'immagine. Come possiamo notare, tutti gli altri polinomi seguono bene la terna dei dati forniti.

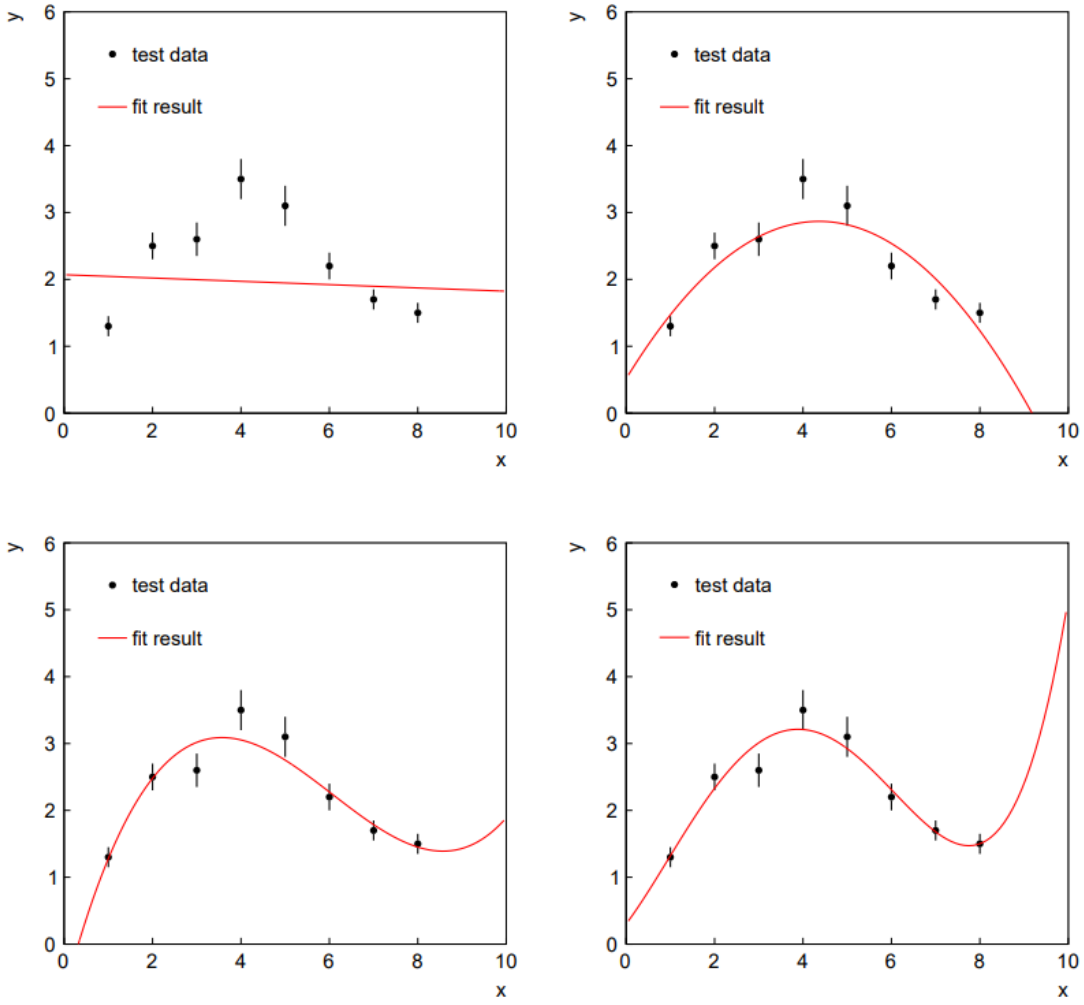

Figura 5: Partendo da in alto a sinistra: polinomio di grado n=1,2,3,4.

L'esercizio continua chiedendo di valutare il polinomio in $x = 5$, $x = 6$ e $x = 10$ nei tre casi di ordine n = 2, 3 e 4 (cioè con 3, 4 e 5 parametri). In questo caso, una volta settati i parametri tramite la funzione *"SetParameter()"* è stato creato un secondo ciclo per valutare la funzione nei diversi punti richiesti. Tuttavia, per come era già stato implementato il codice, non si è generalizzato il caso per i diversi gradi del polinomio, e quindi bisogna cambiare "manualmente" gli ordini del polinomio e rieseguire il programma. I valori ottenuti sono presentati in Tabella (4):

n	f(x = 5)	f(x = 6)	f(x = 10)
2	2.818	2.536	-1.060
3	2.752	2.275	1.888
4	2.924	2.310	5.161

Tabella 4: Valori del polinomio fittato in $x = 5$, $x = 6$ e $x = 10$.

Successivamente viene richiesto di usare la propagazione degli errori, per trova la deviazione standard della differenza, ovvero:

$$d_{ab} = f(x_a; \hat{\theta}) - f(x_b; \hat{\theta}), \tag{20}$$

valutata nelle due coppie di valori ($x_a = 6$, $x_b = 6$) e ($x_a = 10$, $x_b = 5$). Siccome le variabili sono tra di loro correlate, per propagare gli errori è stata utilizzata la seguente formula:

$$\sigma_d^2 = \sum_{i,j}^{n} \left(\frac{\partial d}{\partial \theta_i}\right)\left(\frac{\partial d}{\partial \theta_j}\right) \mathrm{cov}(\theta_i, \theta_j) = \sum_{i=0}^{n} \left(\frac{\partial d}{\partial \theta_i}\right)^2 \sigma_i^2 + \sum_{i \neq j}^{n} 2\left(\frac{\partial d}{\partial \theta_i}\right)\left(\frac{\partial d}{\partial \theta_j}\right) \mathrm{cov}(\theta_i, \theta_j) \tag{21}$$

Avendo come funzione un polinomio, allora essa è lineare nelle θ_i e quindi tale formula risulta corretta. La derivata di (19) rispetto a θ_i è:

$$\frac{\partial f_i(x;\theta)}{\partial \theta_i} = x^i \tag{22}$$

da cui:

$$\frac{\partial d}{\partial \theta_i} = x_a^i - x_b^i. \tag{23}$$

Ora basta implementare il tutto nel programma attraverso due cicli. Nel primo ciclo viene creato il vettore definito in (23). Nel secondo ciclo (doppio ciclo poiché devo usare la matrice ci covarianza che necessita di due indici) applico la (22). I risultati ottenuti sono riportato in Tabella (5):

n	x_a	x_b	f(x_a)	f(x_b)	d_{ab}	σ_d
2	10	5	-1.060	2.818	-3.878	0.671
	6	6	2.536	2.536	0	0
3	10	5	1.888	2.752	0.864	1.37
	6	6	2.275	2.275	0	0
4	10	5	5.161	2.924	2.237	1.087
	6	6	2.309	2.309	0	0

Tabella 5: Deviazione standard della differenza d_{ab} nel caso in cui n= 3.

Chiaramente ci si aspetta che più la differenza d_{ab} è valutata in punti x vicini tra loro, più la deviazione standard diminuisce. Intuitivamente è chiaro poiché se $x_a = 10$ ed $x_b = 10$ chiaramente $d_{ab} = 0$ e la deviazione standard (che segue dalla (21)) è anch'essa zero. Da un punto di vista matematico, basta osservare la (23) e capire immediatamente che se x_a ed x_b sono nettamente diverse tra loro, allora anche la loro differenza lo sarà.
Nell'ultima parte dell'esercizio vengono assegnati dei parametri previsti da un modello: $\theta_0 = -0.75$, $\theta_1 = 2.5$, $\theta_2 = -0.5$ ed $\theta_3 = 0.026$. Usando l'inverso della matrice di covarianza è richiesto di calcolare il χ^2 e di confrontarlo con i valori del nostro modello.

In questo caso è richiesto solo il caso di n = 3. Ricordo che il χ^2 dipende dall'inverso della matrice di covarianza secondo:

$$\chi^2 = \sum_{i,j=0}^{n} (\theta_{mod,i} - \hat{\theta}_i)(V^{-1})_{ij}(\theta_{mod,j} - \hat{\theta}_j). \tag{24}$$

Per prima cosa è stato implementato il codice per calcolare il χ^2 ed il p-valore. I risultati sono riportati in Tabella (6):

n	χ^2	p-valore
3	57.375	$1.033 \cdot 10^{-11}$

Tabella 6: Valori di χ^2 e p-valore.

Infine è stato creato un grafico (Figura (6)) per mettere a confronto se il modello è in accordo con con i valori dei parametri stimati.

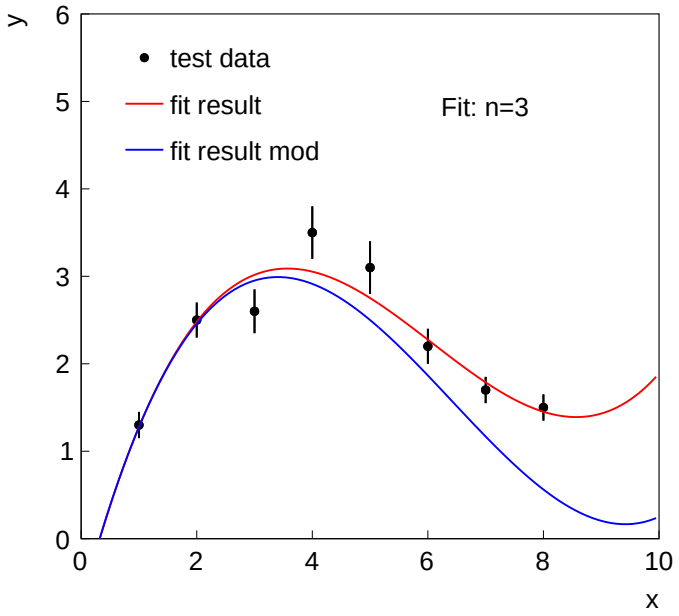

Figura 6: Fit di Maximum Likelihood in rosso ed il Fit del modello con i parametri fissati in blu.

Come si può osservare, all'inizio sembra che il modello sia in buon accordo con il Fit di Maximum Likelihood ma soprattutto verso la fine si osserva una grande discrepanza. Ciò era già stato confermato dai valori del χ^2 e dal p-valore.

Problema 4

Esercizio 1

L'esercizio genera un campione di dati di n = 200 valori da una pdf che è data dalla somma di una funzione esponenziale e di una gaussiana:

$$f(x;\theta,\xi) = \theta\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma}} + (1-\theta)\frac{1}{\xi}e^{-\frac{x}{\xi}} \quad (x \geq 0) \tag{25}$$

Per impostazione predefinita, il programma fissa i parametri μ e σ e tratta solo i parametri θ e ξ come liberi. L'esercizio richiede di produrre la funzione fittata, un grafico dello scan di -log(L) verso θ ed un "contour" plot di:

$$\log(L) = -\log(L_{max}) + \frac{1}{2} \tag{26}$$

nello spazio dei parametri (θ, ξ).

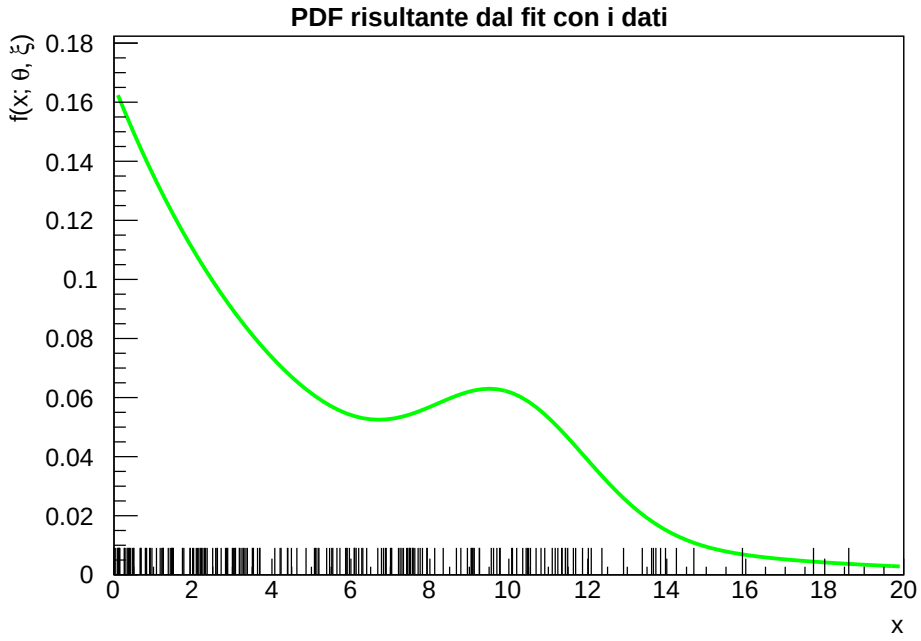

Figura 7: Risultati del fit di Maximum Likelihood.

L'esercizio esegue un fit di Maximum Likelihood (ML) fittando un set di dati generati in modo casuale tramite la (25).

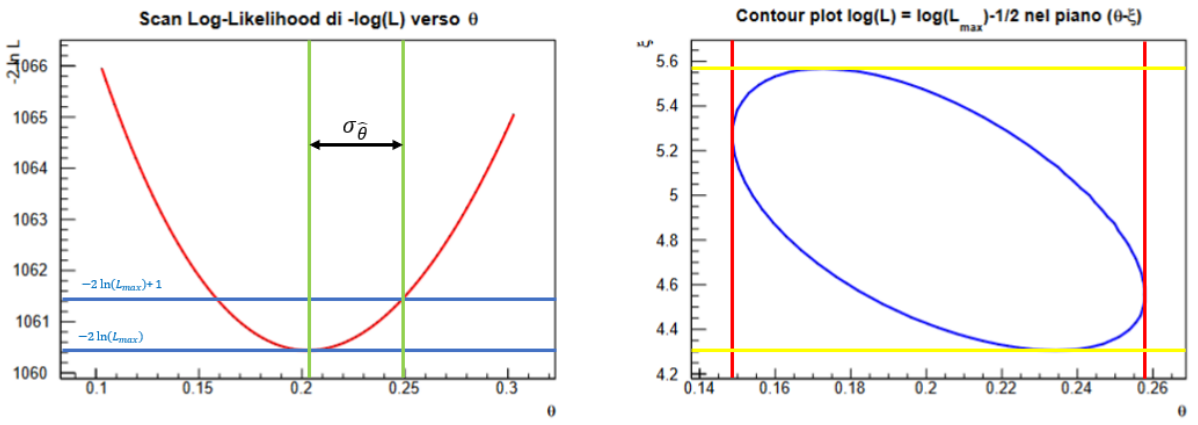

Figura 8: A sinistra lo scan verso θ della Likelihood. A destra il contour plot in funzione di (θ, ξ).

In seguito, si richiede l'utilizzo di un metodo grafico per ottenere le deviazioni standard dei parametri e confrontarle con i valori stampati dal programma. Nello specifico, tramite la funzione (26) è possibile ottenere la stima di $\sigma_{\hat{\theta}}$. Dallo scan (Figura (8)) verso θ, valutando la funzione nel minimo ed utilizzando (26), si è ottenuto che:

$$\sigma_{\hat{\theta}} = 0.047 \quad \text{Scan.} \tag{27}$$

Mentre dal contour in Figura (8), tracciando le tangenti orizzontali e verticali al grafico, si è ottenuto che:

$$\begin{aligned} \sigma_{\hat{\theta}} &= 0.048 \quad \text{Contour.} \\ \sigma_{\hat{\xi}} &= 0.635 \quad \text{Contour.} \end{aligned} \tag{28}$$

I risultati ottenuti graficamente rispecchiano quelli stampati dal programma, ovvero:

$$\begin{aligned} \sigma_{\hat{\theta}} &= 0.0544 \quad \text{Programma.} \\ \sigma_{\hat{\xi}} &= 0.626 \quad \text{Programma.} \end{aligned} \tag{29}$$

Il secondo punto dell'esercizio richiede di dimostrare che l'inverso della matrice di covarianza, nel limite di grande campione, è proporzionale alla dimensione del campione stesso. Supponiamo che la dimensione del campione sia n. Anzitutto è necessario scrivere la forma generale della probabilità per un campione indipendente e identicamente distribuito:

$$L(\theta) = \prod_{i=0}^{n} P(x_i|\theta) = \prod_{i=0}^{n} f(x_i;\theta) \tag{30}$$

Assumendo un grande campione, allora la covarianza inversa (dall'informazione di Fisher) è:

$$V_{ij}^{-1} = -E\left[\frac{\partial^2 \ln(L)}{\partial\theta_i \partial\theta_j}\right] = -\int \frac{\partial^2 \ln(L)}{\partial\theta_i \partial\theta_j} P(\mathbf{x}|\theta) d\mathbf{x} \tag{31}$$

Poiché

$$\ln(L(\theta)) = \sum_{i=1}^{n} \ln(P(x_i|\theta)), \tag{32}$$

si trova che:

$$V_{ij}^{-1} = -\sum_{i=1}^{n} \int \frac{\partial^2 \ln(P(x_i|\theta))}{\partial\theta_i \partial\theta_j} P(x_i|\theta) dx_i = -n \int \frac{\partial^2 \ln(P(x_i|\theta))}{\partial\theta_i \partial\theta_j} P(x|\theta) dx. \qquad (33)$$

Da cui è chiaro che $V \propto \frac{1}{n}$ e quindi $\sigma_{\hat{\theta}_i} \propto \frac{1}{\sqrt{n}}$.
Come punto successivo l'esercizio richiede di rieseguire il programma per diverse dimensioni del campione di dati con $n = 100, 200, 400, 800$ eventi e trovare in ogni caso la deviazione standard di $\hat{\theta}$. Chiaramente, ci si aspetta che l'andamento di tali deviazioni standard rispecchia il risultato trovato prima, ovvero $\sigma_{\hat{\theta}_i} \propto \frac{1}{\sqrt{n}}$. Per verificarlo, in Figura (9) è riportato il grafico delle deviazioni standard.

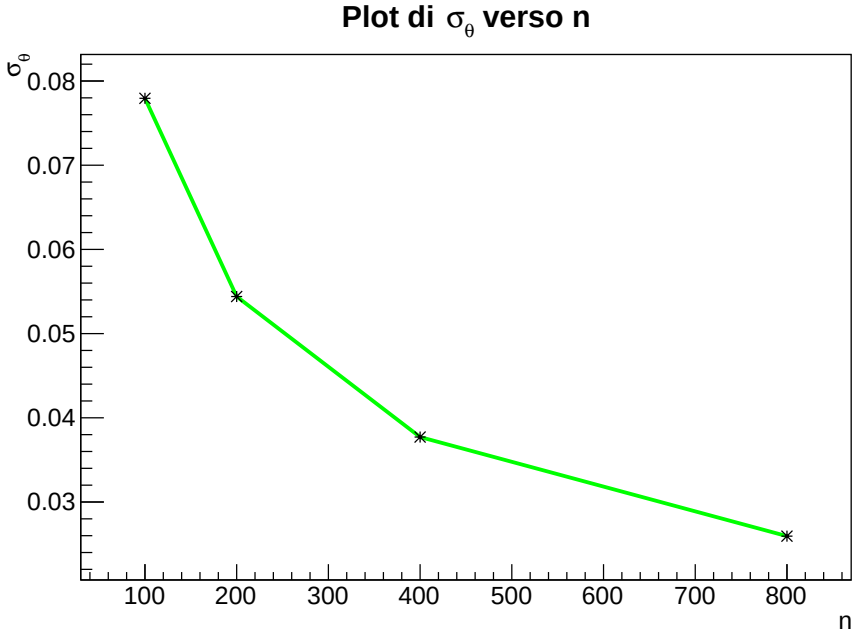

Figura 9: Risultati delle deviazioni standard per diverse dimensioni del campione.

Infine si è trovata $\hat{\theta}$ e la sua deviazione standard per i diversi casi di parametri liberi. Infatti nel programma è possibile variare il numero di paramenti liberi, modificando opportunamente le righe di codice in cui venivano inizializzati i parametri iniziali. La Tabella (7) riporta i valori di $\hat{\theta}$ e $\sigma_{\hat{\theta}}$. Si può osservare come al crescere del numero

Parametri liberi	**Parametri fissi**	$\hat{\theta}$	$\sigma_{\hat{\theta}}$
θ	$\mu,\ \sigma,\ \xi$	0.1976	0.0451
$\theta,\ \xi$	$\mu,\ \sigma$	0.2029	0.0544
$\theta,\ \mu,\ \xi$	σ	0.2252	0.0583
$\theta,\ \mu,\ \sigma,\ \xi$	-	0.3315	0.1181

Tabella 7: Valori di $\hat{\theta}$ e $\sigma_{\hat{\theta}}$ in base al numero di parametri liberi.

di parametri liberi, la stima su $\hat{\theta}$ e $\sigma_{\hat{\theta}}$ è sempre meno precisa. Ciò è dovuto proprio all'efficienza del programma, il quale più parametri liberi ha, più è *"complicato"* stimarli.

Problema 5

Esercizio 1

In questo problema viene fornita una likelihood per un i-esimo eseprimento definita come:

$$\mathcal{L}_i(s, b|n_i, \bar{b}_i) = \mathcal{P}(n_i|s+b)\mathcal{G}(\bar{b}_i|b, \sigma_b) \tag{34}$$

dove $\mathcal{P}(n_i|s+b)$ è una distribuzione Poissoniana mentre $\mathcal{G}(\bar{b}_i|b, \sigma_b)$ è una distribuzione Gaussiana. I parametri s e b sono rispettivamente il segnale ed il fondo del nostro esperimento ed n_i sono il numero di osservazioni totali. La variabile $\bar{b}_i$ è il valore di aspettazione di fondo, ottenuto da una misurazione esterna con con incertezza σ_b.
Anzitutto viene richiesto di dimostrare (analiticamente) quali siano gli stimatori per s e b. Per farlo basta esplicitare la likelihood, calcolarne il logaritmo e porre la sua derivata (rispetto ad s e b) uguale a zero. Per non appesantire troppo la notazione, non riporto le dipendenze in $\mathcal{L}_i(s, b|n_i, \bar{b}_i)$ ma scriverò solamente $\mathcal{L}_i$:

$$\begin{aligned}
\mathcal{L}_i &= \frac{(s+b)_i^n}{n_i!} e^{-(s+b)} \cdot \frac{1}{\sqrt{2\pi\sigma_b^2}} e^{-\frac{(\bar{b}_i - b)^2}{2\sigma_b^2}} \\
\log(\mathcal{L}_i) &= n_i \log(s+b) - (s+b) + \frac{1}{2}\log\left(\frac{1}{2\pi\sigma_b^2}\right) - \frac{(\bar{b}_i - b)^2}{2\sigma_b^2} \\
\log(\mathcal{L}_i) &= n_i \log(s+b) - (s+b) - \frac{1}{2}\log(2\pi\sigma_b^2) - \frac{(\bar{b}_i - b)^2}{2\sigma_b^2} \qquad (35) \\
\frac{\partial \log(\mathcal{L}_i)}{\partial s}\Big|_{s=\hat{s}, b=\hat{b}} &= \frac{n_i}{\hat{s}+\hat{b}} - 1 = 0 \\
\frac{\partial \log(\mathcal{L}_i)}{\partial b}\Big|_{s=\hat{s}, b=\hat{b}} &= \frac{n_i}{\hat{s}+\hat{b}} - 1 + \frac{(\bar{b}_i - \hat{b})}{\sigma_b^2} = 0.
\end{aligned}$$

Concentriamoci sulla derivata rispetto ad s. Essa può assumere diversi valori a seconda della variabile $\bar{b}_i$. Infatti:

$$\hat{s} = \begin{cases} n_i \quad \text{se} \quad \bar{b}_i \leq 0 \\ n_i - \hat{b} \quad \text{se} \quad 0 < \bar{b}_i \leq n_i \\ 0 \quad \text{se} \quad \bar{b}_i \geq n_i. \end{cases} \tag{36}$$

Ovvero, se il valore di aspettazione di background è minore di zero, allora non ci si aspetta fondo e quindi $b = 0$. Contrariamente, se il valore di aspettazione di background

è maggiore del numero totale di osservazioni allora si ha solo fondo e quindi non c'è segnale. Chiaramente, se siamo in una via di mezzo, allora sia b che s sono diversi da zero.
L'equazione per b invece è di secondo grado e si ottiene facilmente che:

$$\hat{b}^2 + \hat{b}(s - \bar{b}_i + \sigma_b^2) - s\bar{b}_i + \sigma_b^2 s - n_i\sigma_b^2 = 0. \tag{37}$$

Chiaramente, se $\bar{b}_i \leq 0$ allora ne segue immediatamente che $b = 0$, come già detto. Se invece $\bar{b}_i \geq n_i$ l'espressione (37) si semplifica nel seguente modo:

$$\hat{b}^2 - \hat{b}(\bar{b}_i - \sigma_b^2) - n_i\sigma_b^2 = 0, \tag{38}$$

da cui e facile ottenere la soluzione per $\hat{b}$:

$$\begin{aligned} \hat{b} &= \frac{\bar{b}_i - \sigma_b^2 \pm \sqrt{(\bar{b}_i - \sigma_b^2)^2 + 4n_i\sigma_b^2}}{2} \\ \hat{b} &= \frac{\bar{b}_i - \sigma_b^2}{2} \pm \sqrt{\frac{(\bar{b}_i - \sigma_b^2)^2}{4} + n_i\sigma_b^2} \equiv \tilde{b} \end{aligned} \tag{39}$$

Se invece non siamo nel caso $\bar{b}_i \geq n_i$ ma siamo nella zona intermedia, ovvero $0 < \bar{b}_i \leq n_i$, i calcoli diventalo leggermente più lunghi. In particolare, sapendo che $\hat{s} = n_i - b$ si ottiene la seguente equazione:

$$\begin{gathered} \hat{b}^2 + bn_i - \hat{b}^2 - \hat{b}\bar{b}_i + \hat{b}\sigma_b^2 - n_i\bar{b}_i + \hat{b}\bar{b}_i + \sigma_b^2 n_i - \sigma_b^2 b - n_i\sigma_b^2 = 0. \\ \hat{b}n_i - n_i\bar{b}_i = 0. \\ \hat{b} = \bar{b}_i. \end{gathered} \tag{40}$$

Da cui si ottiene che:

$$(\hat{s}, \hat{b}) = \begin{cases} (n_i, 0) & \text{se} \quad \bar{b}_i \leq 0 \\ (n_i - \bar{b}_i, \bar{b}_i) & \text{se} \quad 0 < \bar{b}_i \leq n_i \\ (0, \tilde{b}) & \text{se} \quad \bar{b}_i \geq n_i. \end{cases} \tag{41}$$

Esercizio 2

Premessa

Il problema prosegue chiedendo di eseguire 10000 pseudo esperimenti e di calcolare il coverage dell'upper limiti al 95%, calcolato con il metodo della verosimiglianza, assumendo un limite asintotico per la statistica del test :

$$t = -2\log\left(\frac{\mathcal{L}(s = s', \tilde{b}')}{\mathcal{L}(\hat{s}, \hat{b})}\right) \tag{42}$$

dove $\tilde{b}$ è ottenuto facendo un fit, quando $s = s'$. In particolare bisogna verificare il coverage nelle seguenti condizioni:

1. $s = 50; b = 150; \sigma_b = 30$

2. $s = 5; b = 15; \sigma_b = 9.5$

L'esercizio può essere affrontato con un pseudo esperimento, per poi essere generalizzato a 10000 esperimenti.
Anzitutto è possibile affrontare l'esercizio in due approcci diversi: un approccio bayesiano ed un approccio frequentista. In questo caso ho deciso di riportare solo l'approccio frequentista. L'idea è quella di minimizzare la statistica t in quanto questo significa far avvicinare la likelihood al suo valore massimo. Per far ciò, è possibile utilizzare il teorema di Wilks per convertire la statistica del test in significatività (limite asintotico) e quindi calcolare gli intervalli di confidenza. Questo approccio è valido solo nel limite delle statistiche elevate, ma lo useremo in questo esercizio a scopo didattico. L'idea di generare 10000 pseudo esperimenti deriva proprio dall'ovviare il problema del campione ridotto.
In generale, supponiamo di fissare il numero di osservazioni n, ad esempio $n = 5$. Poiché il numero totale di eventi è fissato, si può fare un grafico della statistica t per diversi valori di s e b ed osservare in quale zona possiamo aspettarci gli intervalli di confidenza. La

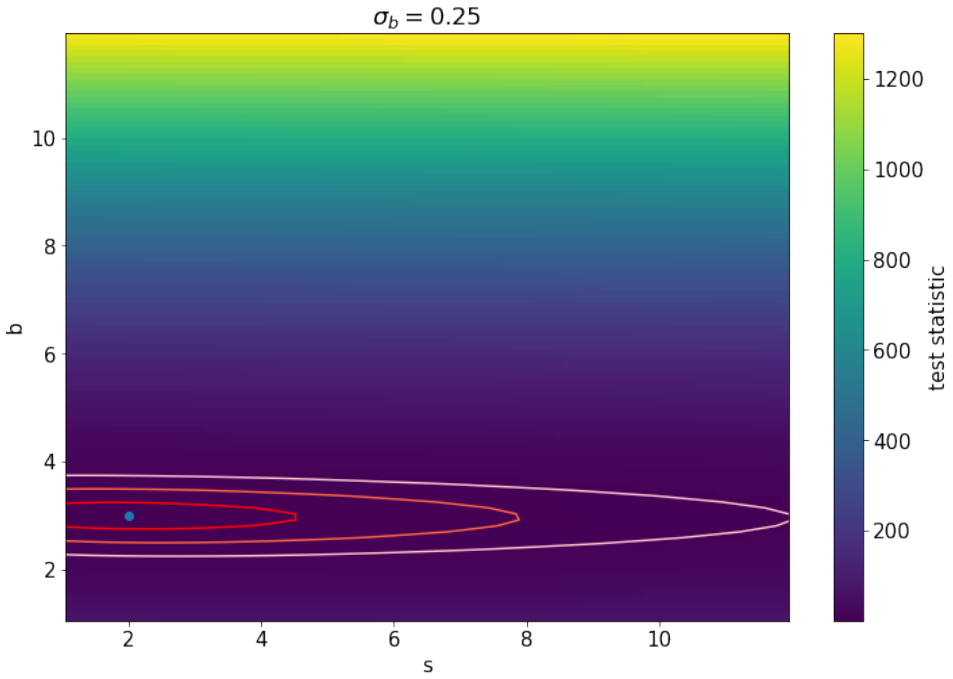

Figura 10: Colorbar bidimensionale rappresentate la statistica t. In questo caso $n = 5$ e $\bar{b} = 3$. Il punto azzurro rappresenta il valor minimo della statistica t mentre le ellissi di diverso colore rappresentano la zona di coverage per diversi livelli di confidenza. In particolare con un C.L. di 68%, 95% e 99%.

Figura (10) rappresenta un tentativo di estrarre informazioni di s (segnale) imponendo dei vincoli su b (background) noti a priori. Come si può osservare dalla Figura (10) il valore stimato per b è 3 con una sigma di 0.25 (questo era ovvio perché sono i valori di

imput messi dall'utente), ma otteniamo un valore di s di circa 2 con una sigma dettata dall'apertura dell'ellisse (coverage). Sembrerebbe che una prima stima dell'upper limit al 68% sia di circa 4.3. L'upper limit al 95% è circa 8 mentre al 99% è circa 12. Ciò si può verificare (confermare) con il metodo della *Profiled Likelihood*. Tale metodo consiste nel fissare il valore di s e stimare $\tilde{b}$ dal fit, proprio come è richiesto dall'esercizio. La Figura (11) conferma quanto detto prima.

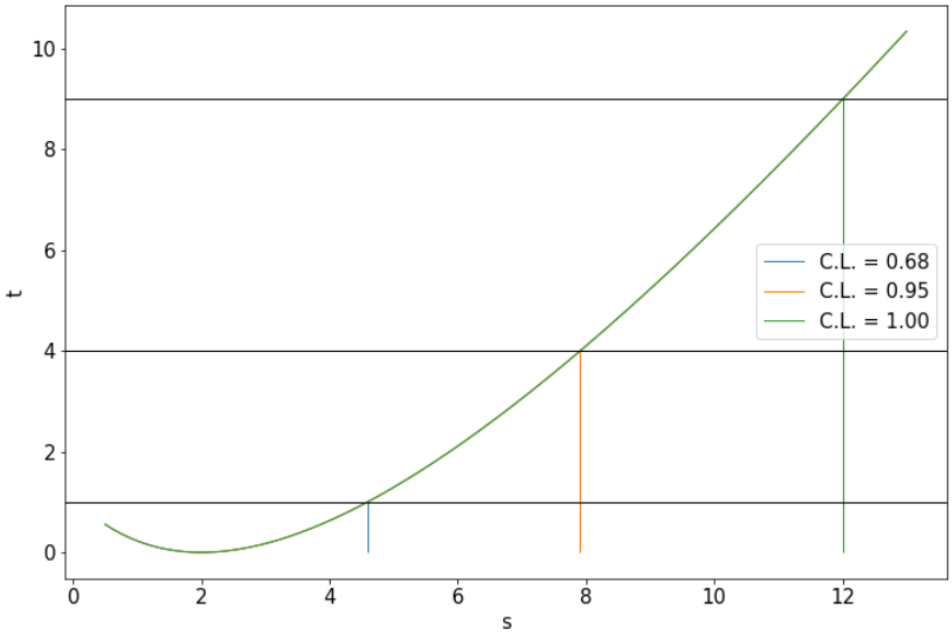

Figura 11: Upper limit per diversi livelli di confidenza.

Soluzione

Dopo questa premessa è possibile svolgere l'esercizio in maniera più semplice perché si tratta solamente di generalizzare il processo sopra spiegato. I paramenti n_i e $\bar{b}_i$ sono stati generati nel seguente modo:

- Sapendo che $n = s + b$, allora gli n_i sono stati creati tramite una distribuzione uniforme tra $[(s+b) - 2\sqrt{s+b}; (s+b) + 2\sqrt{s+b}]$.
- Lo stesso procedimento è stato applicato per $\bar{b}$, ovvero $[\langle b| - 2\sigma_b; \langle b| + 2\sigma_b]$.
- Infine è stato applicato il metodo accept-reject per generare i dati secondo il modello (34).

Una volta generati i dati, è stato creato un ciclo che ripete il procedimento spiegato in precedenza, salvando i dati in un file. A causa dell'elevato tempo computazionale non è stato possibile effettuare 10000 pseudo esperimenti ma solamente 1000.
Inoltre, i dati venivano creati con un picco centrato nel valore iniziale di s, per esempio se s parte da 20, ci sono almeno 100 dati che hanno valore 20. Questo errore potrebbe

essere causato da come vengono generati i dati, ma applicando una "*mask*" si è riusciti ad ovviare tale problema, con il prezzo che il file si è ulteriormente ridotto. In particolare, per $s = 50$ si sono ottenuti 885 dati, mentre per $s = 5$ si sono ottenuti 985 dati.
In Figura (12) e (13) sono rappresentati gli istogrammi dei vari upper limit trovati nei due casi.

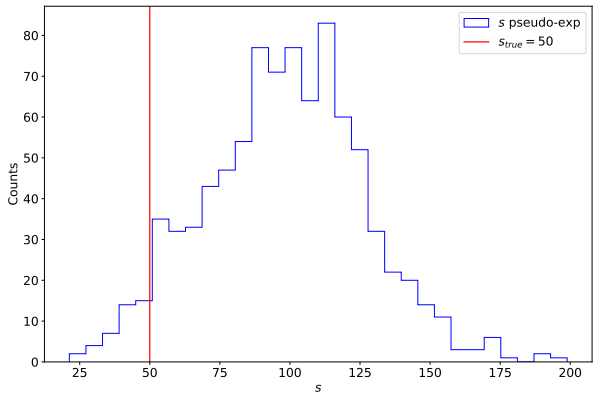

Figura 12: Upper limit con $s = 50$.

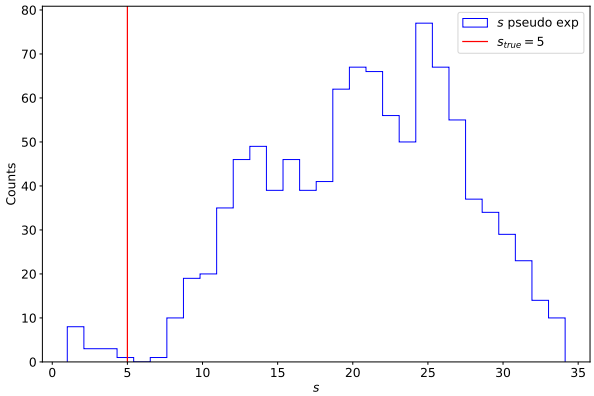

Figura 13: Upper limit con $s = 5$.

Calcolando il numero di volte in cui gli upper limit sono maggiori del valore vero s, si è ottenuto il coverage nei due seguenti casi:

- Caso $s = 50$: Coverage $= 0.9548$
- Caso $s = 5$: Coverage $= 0.9847$

I risultati sono leggermente diversi: ciò potrebbe essere dovuto a molte ragioni, come ad esempio una dimensione del campione diversa, una varianza dei dati diversa, una distribuzione dei dati diversa. In particolare, siccome $s \geq 0$ nel caso in cui $s = 50$ siamo abbastanza lontani da 0 ed, avendo generato il segnale tra 20 e 200, si ha un'alta probabilità che siamo lontani dal limite fisico. Infatti il coverage nel primo caso viene esattamente di 0.95. Nel secondo caso invece si ha $s = 5$ e quindi si è molto più vicini allo zero e questo fa si che in generale i valori dell'upper limit siano maggiori di 5. Ciò comporta quindi un "*over-coverage*" ovvero una sovrastima del coverage.

Problema 6

Esercizio 1

In pratica, quando si effettuano molte analisi su un insieme di dati, aumenta la probabilità di trovare un effetto significativo casualmente, anche se non esiste effettivamente alcuna relazione significativa. Questo può portare a risultati cosi detti "*falsi positivi*", in cui un effetto casuale viene erroneamente considerato significativo. Tale effetto va sotto il nome di *Look-Elsewhere Effect* e per evitarlo, è necessario correggere il valore di soglia di significatività per il numero di analisi effettuate o utilizzare tecniche statistiche che tengano conto del numero di test eseguiti.
In questo esercizio viene dato il seguente modello:

$$f(m_i; a_i, m_0) = \left| 1 + \frac{a_i e^{i\theta}}{(m_i^2 - m_{0_i}^2) + i m_{0_i} \Gamma} \right|^2 , \tag{43}$$

relativo alle misure di massa invarianti di un insieme di particelle. In particolare a_i ed θ rappresentano rispettivamente la magnitudine relativa e la fase dell'ampiezza risonante rispetto a quella non risonante, m_{0_i} è la massa della particella risonante ed Γ la sua larghezza. I parametri sconosciuti del modello di decadimento sono a_i e m_{0_i}, mentre $\theta = \frac{\pi}{2} - 0.3$ è noto. Ci viene fornito un dataset con 300 misure della massa m_i. La statistica è definita come:

$$t_i = -2 \log \left(\frac{\mathcal{L}_i(a_i = 0)}{\mathcal{L}_i(\tilde{a}, \bar{m}_{0_i})} \right) , \tag{44}$$

dove $\bar{m}_{0_i}$ è il valore assegnato di m_{0_i} ed $\tilde{a}_i$ è il valore del fit di a, per $m_{0_i} = \bar{m}_{0_i}$.
Il primo punto richiede di determinare il p-valore per l'ipotesi nulla $(a_i = 0)$ per $m_{0_i} = \bar{m}_{0_i}$, dove $\bar{m}_{0_i}$ varia tra [5;18] in step di 0.1.

Notazione

Con $\mathcal{L}$ si intende la likelihood. Con L si intende la "*Negative Log-likelihood*". Il logaritmo è già "incorporato" all'interno di L. Spesso si incontrerà il termine *nll* il quale rappresenta l'acronimo di "*Negative Log-likelihood*". Con il simbolo $\mathcal{F}[f(x)]$ si intende il Fit della funzione f(x).

Dati

Come detto in antecedenza, sono state fornite delle misurazioni di m_i riportate in un file. In Figura (2) sono mostrate tali misurazioni. Come si può notare, i dati presentano un

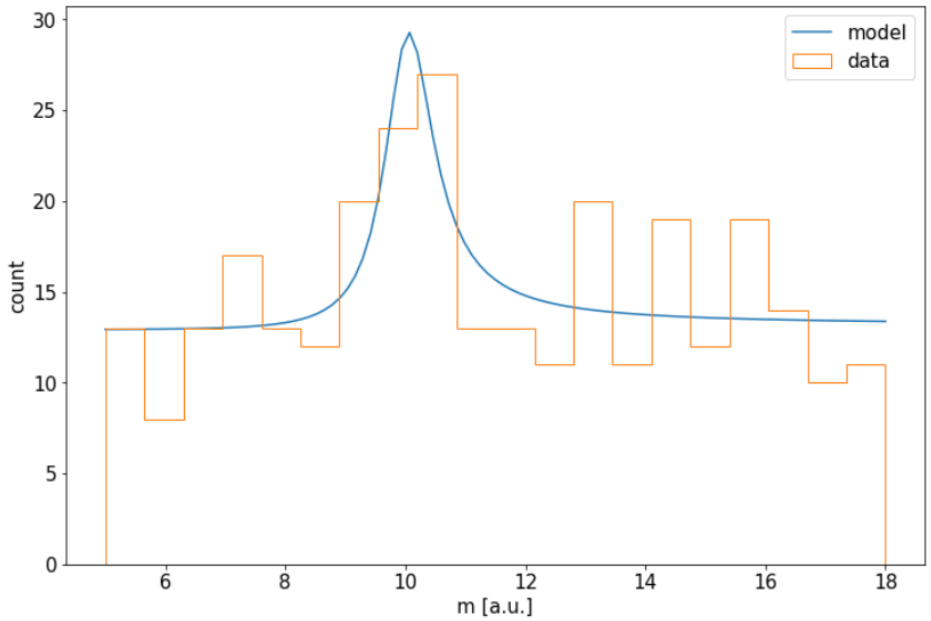

Figura 2: Rappresentazione dei dati in arancione, ed il modello (43) in blu.

picco intorno a $m \approx 10$. Ci chiediamo se questa è una fluttuazione statistica oppure è un qualcosa di "*nuovo*" che si è osservato (per esempio una nuova particella). Per farlo eseguiremo due tipi di analisi, una locale ed una globale.

- L'analisi locale è un'analisi dettagliata dei dati in una regione specifica dell'istogramma. Ad esempio, se si sta cercando di scoprire una nuova particella, si può esaminare una regione specifica dell'istogramma dove ci si aspetta che la particella appaia.

- L'analisi globale, d'altra parte, considera l'intero spettro dell'istogramma, prendendo in considerazione tutte le regioni. L'analisi globale è utile per stabilire se un segnale in una regione specifica dell'istogramma è reale o è solo una fluttuazione casuale. Nell'analisi globale, vengono creati molti pseudo-esperimenti[2] per confrontare il risultato osservato con la distribuzione dei risultati attesi in assenza di un segnale. In questo modo, è possibile calcolare la probabilità di ottenere un segnale come quello osservato per puro caso. Se questa probabilità è sufficientemente bassa (ad esempio, inferiore a un certo valore soglia, come il 5%), si può concludere che il segnale è significativo e non è solo il risultato di fluttuazioni casuali.

Nel nostro caso, se si riscontra un evidenza superiore a 3 sigma si può pensare che sia significativa e quindi bisogna fare ulteriori indagini.

[2]In un pseudo-esperimento, si utilizza un campione casuale di dati simulati, generati in base a una distribuzione di probabilità nota, che rappresenta l'ipotesi nulla (cioè, la situazione in cui non vi è alcun effetto o segnale da rilevare).

Analisi locale

Anzitutto modifico la statistica t_i nel seguente modo:

$$t_i = -2[\log(\mathcal{L}_i(a_i = 0)) - \log(\mathcal{L}_i(\tilde{a}_i, \bar{m}_{0_i}))]. \tag{45}$$

Conviene scriverla così perché computazionalmente è meno pesante. Inoltre, per come è stato scritto il programma, si tiene conto della "*Negative Log-likelihood*" quindi la statistica è:

$$t_i = -2[L_i(\tilde{a}_i, \bar{m}_{0_i})) - L_i(a_i = 0))]. \tag{46}$$

Per calcolare $L(a_i = 0)$ è facile in quanto per $a_i = 0$ il modello (43) è sempre costante, per qualsiasi valore di $\bar{m}_{0_i}$. Per calcolare $L(\tilde{a}, \bar{m}_{0_i})$ invece bisogna procedere per punti:

1. È stato creato un vettore $\bar{\mathbf{m}}_0(5.0, 5.1, ..., 17.9, 18.0)$ che parte da 5 ed, in step di 0.1, arriva a 18. È chiaro che tale vettore sarà composto da 131 valori.

2. È stato fatto un fit, minimizzando la *nnl* ed è stato estratto il valore di $\tilde{a}_i$. Il problema è capire da dove far partire il fit, ovvero i parametri da assegnare alla *nnl*. In particolare, il parametro per m_{0_i} è rappresentato dal vettore $\bar{\mathbf{m}}_0$, mentre come parametro per a_i è stato assegnato 0. Ovvero è stato fatto il fit della seguente funzione:
$$\forall i, \quad \mathcal{F}_i[L_i(a_i = 0, \bar{m}_{0_i})] \implies \tilde{a}_i \tag{47}$$

3. Si ripete il punto 2 per ogni valore di $\bar{\mathbf{m}}_0$, ovvero per 131 volte, ottenendo quindi un vettore $\tilde{\mathbf{a}}$ contenente 131 valori di $\tilde{a}_i$ ognuno di essi risultate dal Fit i-esimo.

4. Infine è stata calcolata la statistica t_i dalla formula (46) ottenendo $\mathbf{t}$.

Sapendo che abbiamo un solo grado di libertà, il p-valore è stato calcolato come $1 - F_{\chi^2}(t)$, dove $F_{\chi^2}(t)$ è la cumulativa del χ^2. Ciò va bene solo per fluttuazioni locali, ma non per quelle globali.

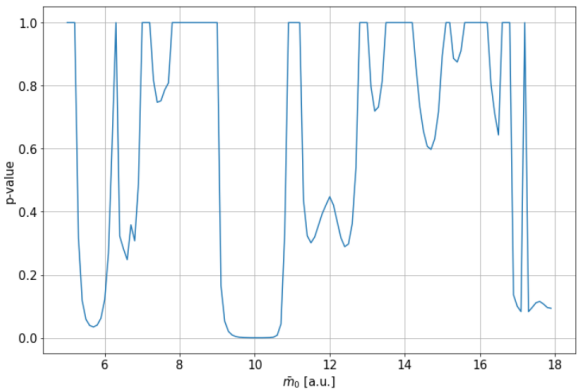

Figura 3: P-value in funzione di m_{0_i}

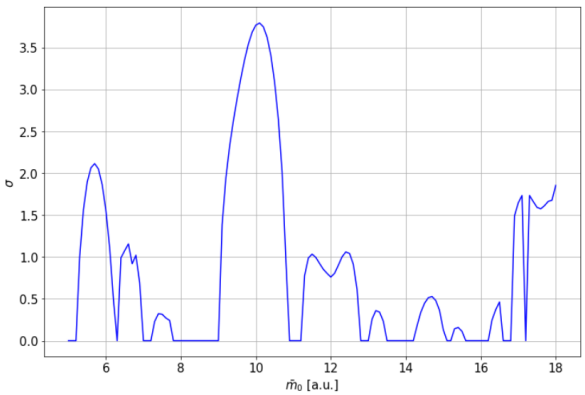

Figura 4: Dev std in funzione di m_{0_i}.

In Figura (3) sono riportati i diversi p-valori in funzione di m_{0_i}. In Figura (4) è stato convertito il p-value osservato in deviazioni standard. Dalla Figura (4) è possibile notare che si ha un picco sopra 3 sigma. Sembrerebbe quindi che ci sia qualcosa di interessante, ma per accertarci di ciò, facciamo un controllo globale simulando 10000 pseudo esperimenti.

Analisi globale

Nel caso dell'analisi globale, come detto sopra, si creano 10000 pseudo-esperimenti. In questo caso si utilizza la seguente statistica:

$$t' = -2\log\left(\frac{\mathcal{L}(a=0)}{\mathcal{L}(\hat{m},\hat{a})}\right) \tag{48}$$

dove $\hat{m}_0$ ed $\hat{a}$ rappresentano le stime di m_0 ed a ricavati dal Fit degli pseudo-esperimenti. In questo caso, per generare i dati, come parametri iniziali sono stati messi $a_i = 0$ ed $m_{0_i} = 0, \forall i$. Successivamente il procedimento è analogo a quello dell'analisi locale, con la differenza che ora si stanno utilizzando dei dati generati, e non dei dati osservativi. Il p-value relativo ai 10000 esperimenti risulta:

$$p-\text{value} = 0.0008 \tag{49}$$

Infine è stata tracciata la distribuzione di t' ottenuta dagli pseudo-esperimenti ed è stata stimata la media usando una funzione χ^2.

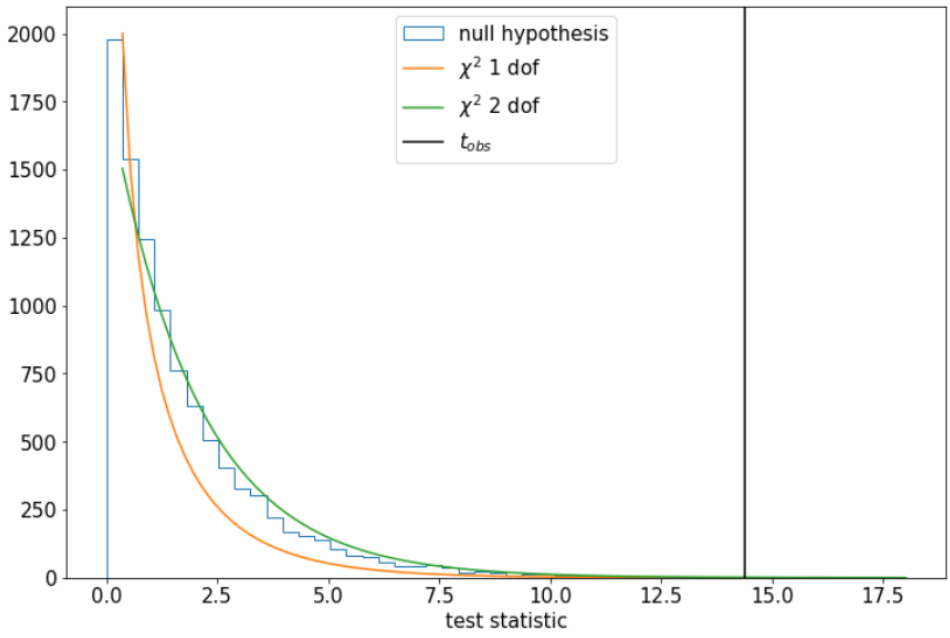

Figura 2: Distribuzione di t'.

Come si può osservare la distribuzione di χ^2 a due gradi di libertà rappresenta più fedelmente la statistica t'.