

Incertezze e best-fit non lineare con Python

francesco.fuso@unipi.it; <http://www.df.unipi.it/~fuso/dida>

(Dated: version 1 - FF 31 dicembre 2015)

La breve discussione riportata in questa nota serve per mettere in luce un aspetto oscuro, ostile, e potenzialmente pericoloso nella procedura di best-fit di Python. Questo argomento è stato stimolato dalle osservazioni di FLD e AC, che sono ampiamente lodati per la loro attenzione e ringraziati per la segnalazione.

I. INTRODUZIONE

Come abbiamo avuto ripetutamente occasione di affermare, l'esecuzione di un best-fit non è mai un'operazione routinaria e tutti i dettagli coinvolti devono sempre essere debitamente considerati. In questa nota si fa riferimento a una particolarità della procedura di best-fit di Python: tra le opzioni del comando `curve_fit` ce ne è una che suona piuttosto misteriosa, e rimane tale anche dopo aver consultato la scarna e confusa documentazione disponibile in rete.

L'opzione in questione è `absolute_sigma`, che può avere come valore `False` (di default, cioè attiva anche senza porre alcuna specifica nella linea di comando di `curve_fit`) oppure `True`. Potete verificare facilmente come la scelta di questa opzione non modifichi il valore dei parametri risultanti dal best-fit, ma come essa agisca, spesso in maniera non trascurabile, sulla determinazione dell'incertezza da attribuire al valore del parametro.

II. ERRORI STATISTICI E NON

Normalmente le incertezze di misura possono essere catalogate come di origine statistica o non statistica, spesso definita sistematica. Nella pratica può essere difficile separare i due contributi, che spesso sono intrecciati tra di loro, ma, almeno concettualmente, una categorizzazione così fatta è ragionevole. Essa, infatti, permette di distinguere fra quelle cause di incertezza che sono "intrinseche" alla misura (l'errore statistico) e quelle che, invece, almeno idealmente, possono essere sanate (ad esempio, attraverso una calibrazione più accurata dello strumento di misura).

Come ben sapete, per moltissime misure (non tutte le misure) si può assumere che la distribuzione di probabilità associata (*parent distribution*) sia normale, cioè che possa essere descritta da una funzione gaussiana. La larghezza σ^2 , o varianza, di questa distribuzione dà un'informazione convenzionale sull'entità dell'incertezza stocastica. Più precisamente, si assume che l'incertezza sia ben rappresentata dalla *deviazione standard* σ della distribuzione. Inoltre si assume che una buona approssimazione della parent distribution sia data dalla distribuzione sperimentale delle misure su un campione "sufficientemente" grande, che tutte le misure siano indipendenti, o scorre-

late tra loro, e che σ possa essere approssimata in queste condizioni dalla deviazione standard sperimentale.

La maggior parte dei metodi di data reduction and analysis discendono da queste assunzioni. Per esempio la propagazione dell'errore e il metodo del minimo χ^2 presuppongono di trattare grandezze affette (principalmente) da errore statistico. Per il best-fit, questo significa poter pesare i residui quadrati con l'inverso della varianza sperimentale, e di poter utilizzare il valore del χ^2 ottenuto dal fit come indicatore della sua affidabilità, o significatività (test del χ^2).

La realtà sperimentale tipica delle misure di segnali elettrici è, purtroppo, ben diversa. Infatti molto spesso la sensibilità degli strumenti è minore rispetto alla deviazione standard sperimentale delle misure stesse, che quindi non può essere determinata in modo immediato. Ad esempio, se si ripete tante volte la misura della stessa grandezza con il multimetro digitale si ottengono generalmente fluttuazioni che, al massimo, sono sull'ultima cifra (digit) del display. Analogamente, se si eseguono tante misure di d.d.p. con Arduino si determina generalmente una deviazione standard sperimentale inferiore al singolo bit (digit) che può essere letto. La situazione potrebbe essere un po' meno sfavorevole usando l'oscilloscopio, specialmente nel caso di segnali rapidamente variabili, oppure per altre misure, per esempio quella della frequenza del segnale prodotto dal generatore di forme d'onda, in cui le fluttuazioni possono essere più evidenti. Tuttavia, anche per ragioni di mera praticità (non sempre c'è tempo a sufficienza per collezionare il campione di misure), siamo normalmente portati a considerare come incertezza di origine statistica quella che, in realtà, sarebbe più propriamente la sensibilità della misura (per esempio, il singolo digit, ovvero l'ultima cifra presentata dal display del multimetro). Abbiamo dimostrato, con una specifica esperienza con Arduino, che questa scelta costituisce in realtà una sovrastima (ragionevole, ma non nulla) dell'incertezza statistica.

Dall'altra parte, tutti gli strumenti che misurano grandezze elettriche possono essere considerati dei "trasduttori" e come tali misurano la data grandezza elettrica con un'accuratezza che dipende dalla calibrazione. L'incertezza di calibrazione è, normalmente (e ovviamente), significativamente maggiore di quella statistica nella maggior parte delle condizioni di impiego e per la maggior parte degli strumenti di uso pratico,

Di conseguenza, le condizioni nelle quali ci troviamo

ad operare, per necessità, o, qualche volta, per praticità, sono le peggiori per l'applicazione rigorosa dei metodi di data reduction and analysis [1]. In particolare, la sovrastima delle incertezze sui dati dovuta alla considerazione dell'errore sistematico comporta, molto spesso, una sottostima del χ^2 ottenuto dal best-fit, e la conseguente impossibilità di servirsi del test del χ^2 con cognizione di causa.

III. FALSE VS TRUE

È facilissimo verificare cosa determina la scelta `absolute_sigma = True` [2] rispetto a quella, di default (dunque valida anche senza specificare l'opzione) `absolute_sigma = False`. Detta Δp_m l'incertezza sul parametro p_m del best-fit, cioè la grandezza ottenuta estraendo la radice quadrata degli elementi diagonali della *covariance matrix*, si ha

$$\Delta p_m|_{\text{False}} = \sqrt{\chi_{rid}^2} \Delta p_m|_{\text{True}}. \quad (1)$$

In altre parole, l'incertezza sui parametri di fit data dall'opzione di default (`False`) è “pesata”, o “normalizzata”, con la radice quadrata del χ_{rid}^2 .

Come illustreremo nel seguito, l'opzione `True` conduce a una valutazione dell'errore sui parametri che è in linea con le aspettative naïf. Supponiamo infatti di voler fittare dai dati a una retta: operando in modo grafico (con matita e righello), possiamo facilmente dedurre che maggiore è l'entità delle barre di errore dei nostri dati, maggiore è l'incertezza con cui possiamo determinare i parametri della retta di fit, cioè intercetta e coefficiente angolare.

Questo è in effetti quanto si verifica con la procedura *analitica* di best-fit lineare. In essa le incertezze sui parametri sono costruite lavorando con la propagazione dell'errore, dunque con un approccio valido quando le misure sono ritenute indipendenti e scorrelate tra loro, affermazione certamente corretta quando le barre di errore hanno un'origine statistica. È rilevante notare, come fatto da FLD e AC, che le incertezze sui parametri fornite dalla procedura numerica di Python coincidono con quelle determinate analiticamente solo se si usa l'opzione `True`.

Partendo dalla supposizione che chi ha scritto la procedura `curve_fit` volesse, con la scelta di default `False`, soddisfare le esigenze di best-fit più comuni, che effettivamente sono quelle in cui l'errore sistematico prevale, possiamo cercare un rationale per la normalizzazione effettuata dalla procedura stessa. È ovvio che, nel caso di sovrastima delle barre di errore, il χ_{rid}^2 tende a valere meno di uno, per cui la normalizzazione di Eq. 1 agisce in modo da ridurre l'incertezza sulla valutazione dei parametri. Viceversa, nel caso in cui le barre di errore fossero, per qualche motivo, sottostimate, il χ_{rid}^2 tenderebbe a essere ben maggiore di uno, e quindi la normalizzazione agirebbe in senso opposto. Naturalmente per un best-fit “a regola d'arte”, cioè realizzato con una appropriata

funzione modello e facendo uso di incertezze di origine rigorosamente statistica, allora $\chi_{rid}^2 \simeq 1$ e la normalizzazione non ha alcun effetto sulla valutazione dell'incertezza sui parametri, cioè l'incertezza sui parametri è la stessa indipendentemente dall'opzione.

A. Asymptotic vs standard

La definizione comunemente accettata per l'incertezza sui parametri risultanti da un best-fit è la seguente [3]: Δp_m costituisce l'intervallo di variazione di p_m che corrisponde a un aumento unitario del χ^2 . Questa definizione è in linea con le definizioni correnti di incertezza per altre grandezze, misurate o calcolate, per cui essa prende qualche volta il nome di *standard error*.

Accanto a questa definizione ne esiste, anche se in forma un po' clandestina, un'altra, a cui qualche volta si fa riferimento con il nome *asymptotic error*. Nel caso di misure, il carattere asintotico significa che questa incertezza sarebbe quella determinata su un campione contenente un numero infinito di elementi. Se l'errore è dominato da fluttuazioni statistiche, allora le due definizioni di incertezza coincidono perfettamente.

Nel caso del best-fit su dati pesati, cioè dotati di barre di errore, l'errore asintotico sui parametri del best-fit può essere inteso come l'incertezza che si avrebbe se il numero di dati fosse molto grande e se la loro incertezza di misura fosse di origine puramente statistica. In queste condizioni il metodo del minimo χ^2 stabilisce $\chi_{rid}^2 \simeq 1$ (il best-fit “a regola d'arte”, secondo quanto scritto prima). Allora possiamo rifrasiare la definizione di errore asintotico sui parametri del best-fit, dicendo che esso rappresenta l'incertezza che sarebbe attribuita alla valutazione dei parametri *se fosse* $\chi_{rid}^2 \simeq 1$.

In sostanza, la procedura `curve_fit` di Python consente di scegliere le due definizioni di errore sui parametri:

```
absolute_sigma = False → asymptotic error
absolute_sigma = True  → standard error .
```

Dimostrare che la normalizzazione attraverso $\sqrt{\chi_{rid}^2}$ espressa in Eq. 1 conduca effettivamente a quanto affermato sopra è compito piuttosto complicato dal punto di vista della matematica. Qualche hint può essere trovata in [4] e, meglio ancora, in [5], dove si illustra un pacchetto di Python, chiamato `kmpfit`, che, basandosi su `scipy.optimize`, permette di eseguire best-fit con un certo numero di funzionalità aggiuntive pre-assemblate.

B. Gli altri

Ho fatto un rapido controllo per verificare quale sia la definizione di errore sui parametri di best-fit in uso in alcuni software di trattamento dati (nei limiti delle mie

possibilità). Usano l’asymptotic error, corrispondente all’opzione `False` di Python: MATLAB (R2015b), gnuplot (5.0.0), Mathematica (6.0, in forma piuttosto complicata); usa lo standard error, corrispondente all’opzione `True` di Python, IgorPro (6.37), un software orientato più a trattare dati di interesse per la fisica che non per l’ingegneria. OriginPro, che non uso, dovrebbe consentire la scelta dei due metodi attraverso opportuna definizione dell’incertezza di misura sui dati da fittare e similmente dovrebbero comportarsi, per quanto ne so, altri software pensati per analisi di dati in fisica.

Visto che si parla di altri softwares, e anche se non è del tutto appropriato, faccio qui un breve cenno ad altre particolarità di codifica dei risultati del best-fit.

- *Confidence level*: assumendo che il best-fit sia “a regola d’arte” (funzione modello appropriata, errore statistico sui dati, misure scorrelate tra loro e, in definitiva, $\chi_{rid}^2 \simeq 1$), può fare comodo esplicitare il livello di confidenza con cui il best-fit individua il valore dei parametri. In questo contesto, l’errore standard definito prima corrisponde a una deviazione standard nella distribuzione del χ^2 (che qui è ovviamente supposta valida) e quindi a un livello di confidenza del 68%; un livello di confidenza del 95% corrisponde a due deviazioni standard, del 99% a tre deviazioni standard.
- *R-squared*: nella nostra pratica di impiego del best-fit siamo abituati a valutare la “bontà” del fit attraverso l’analisi dei residui normalizzati, della covarianza normalizzata (se c’è più di un parametro) e infine del χ_{rid}^2 . Poiché normalmente abbiamo ottimi motivi per aspettarci che il χ_{rid}^2 sia viziato dalla non corretta identificazione dell’errore sui dati sperimentali, generalmente sovrastimati a causa del contributo sistematico, può in qualche caso essere utile estrarre altri parametri numerici che informino sulla qualità del best-fit, senza essere così fortemente condizionati dalla determinazione delle barre di errore. Di questi parametri ne esistono parecchi (si veda, ad esempio, l’approccio ANOVA [6], che, essendo stato inventato per analizzare le covarianze in popolazioni statistiche, può essere esteso all’esame dei risultati del best-fit). Uno particolarmente semplice da calcolare è l’R-squared (R^2), definito come

$$R^2 = 1 - \frac{\sum_i ((y_i - f(x_i))^2 / \sigma_i^2)}{\sum_i ((y_i - \bar{y})^2 / \sigma_i^2)}, \quad (2)$$

dove (x_i, y_i) sono le coppie di dati sperimentali da fittare, σ_i rappresenta la deviazione standard usata come peso per il best-fit, \bar{y} è il *valore medio* dei dati acquisiti e le somme si intendono estese su tut-

to il campione di misure. Questo parametro dà un’indicazione della capacità del fit di riprodurre la variazione dei dati osservati e si può dimostrare [7] che esso, normalmente compreso tra 0 e 1, tende tanto più all’unità quanto meglio la funzione modello è in grado di interpretare l’osservazione sperimentale. Ciò può essere facilmente compreso in termini qualitativi: per un “buon” fit ci si aspetta che la somma dei residui quadri, che compare, debitamente pesata, al numeratore della frazione, sia molto minore dello scarto quadrato medio dei dati che sta al denominatore, per cui la funzione di Eq. 2 tende in queste condizioni all’unità. Viceversa essa tende a zero nel caso in cui residui e scarto siano paragonabili tra loro, circostanza che si verifica quando il best-fit non riproduce in modo corretto le osservazioni sperimentali [8].

IV. CONCLUSIONI

Questa nota raccoglie informazioni e propone argomenti di vario genere, ognuno dei quali potrebbe meritare un approfondimento specifico. Tuttavia il suo scopo principale è quello di fornire delle semplici indicazioni pratiche, per quanto possibile tenendo conto del complicato scenario nel quale ci si muove.

Il messaggio principale è che la determinazione dell’incertezza sui parametri individuati dal best-fit è “critica”, potendo dipendere dalla valutazione delle barre di errore sui dati da fittare, che è a sua volta un’operazione delicata. Infatti, la situazione “ideale”, in cui gli errori di misura sono di origine prevalentemente statistica, si incontra raramente nella pratica delle nostre esperienze. Eseguire un best-fit richiede attenzione in tutti i passaggi coinvolti e nella valutazione dei risultati: l’incertezza sui valori dei parametri ottenuti dal best-fit deve essere considerata con attenzione per evitare conclusioni erranee.

Python conosce la situazione e ci viene incontro, proponendo la scelta fra due opzioni, pur poco chiare e documentate malissimo. Dal punto di vista operativo, le osservazioni che abbiamo svolto suggeriscono che:

1. se gli errori dei dati sperimentali sono di origine prevalentemente statistica, l’opzione da usare è `True`;
2. se invece l’origine degli errori è poco conosciuta, oppure se si sa che il contributo sistematico è dominante, come si verifica molto spesso nelle nostre esperienze, è *consigliata* l’opzione `False`, quella di default, che consente di evitare marchiane sovrastime (in qualche caso sottostime) dell’incertezza sui valori di parametri di best-fit.

[1] Naturalmente è possibile compiere degli sforzi finalizzati a migliorare questa situazione. Per esempio, nelle misure

automatizzate con Arduino si può in genere prevedere la

possibilità di estrarre la deviazione standard sperimentale su un campione ragionevolmente esteso di misure ripetute in automatico da Arduino. Questo è il punto di partenza per compiere una distinzione tra contributi statistici e sistematici all'errore. Si potrebbe ad esempio eseguire gli eventuali best-fit per l'analisi dei dati usando solo l'incertezza statistica e poi tenere conto del contributo sistematico nella conversione dei risultati del best-fit a unità "fisiche".

- [2] L'opzione `False` o `True` deve essere inserita tra gli argomenti della chiamata alla procedura di best-fit. La sintassi del comando diventa, nei due casi trattati in questa nota, rispettivamente `curve_fit(..., absolute_sigma=False)` o `curve_fit(..., absolute_sigma=True)`. Se l'opzione

non viene esplicitata, vale la scelta di default (`False`).

- [3] P.R. Bevington and D.K. Robinson, *Data reduction and error analysis for the physical sciences* (McGraw-Hill, New York, 2002).
- [4] P.H. Richter, "Estimating Errors in Least-Squares Fitting", TDA Progress Report 42-122 (1995); http://ipnpr.jpl.nasa.gov/progress_report/42-122/122E.pdf
- [5] <https://www.astro.rug.nl/software/kapteyn/kmpfittutorial.html>
- [6] https://en.wikipedia.org/wiki/Analysis_of_variance
- [7] https://en.wikipedia.org/wiki/Coefficient_of_determination
- [8] Il parametro R-squared nasce per l'analisi della significatività di best-fit lineari. La sua estensione a best-fit non lineari non è scontata, specie per alcune tipologie di funzioni modello, ma tuttavia esso è spesso impiegato in maniera estensiva.