Resampling Methods - 21/10

Resampling Methods

I Metodi di Resampling consistono nel “pescare” elementi dal training set ripetutamente ed effettuare il refitting del modello di interesse su ogni sample per recuperare informazioni aggiuntive sul modello fitted, ad esempio stime del prediction error del test-set e una caratterizzazione degli stimatori dei parametri.

Questi metodi possono essere computazionalmente esosi in quanto lo stesso metodo statistico è ripetuto multeplici volte usando differenti sottoinsiemi del training data set.

Gli **obiettivi** sono:

- Model selection: selezionare il livello appropriato di model flexibility, identificare i principali regressori per descrivere una variabile dipendente ecc... in breve è la stima delle performance di diversi modelli al fine di scegliere il migliore;
- Model Assessment: una volta scelto un modello finale, quantificare l’incertezza del modello, stimare il suo errore di predizione (test error rate, errore di generalizzazione) su nuovi dati.

I **metodi** maggiormente usati in statistical learning sono:

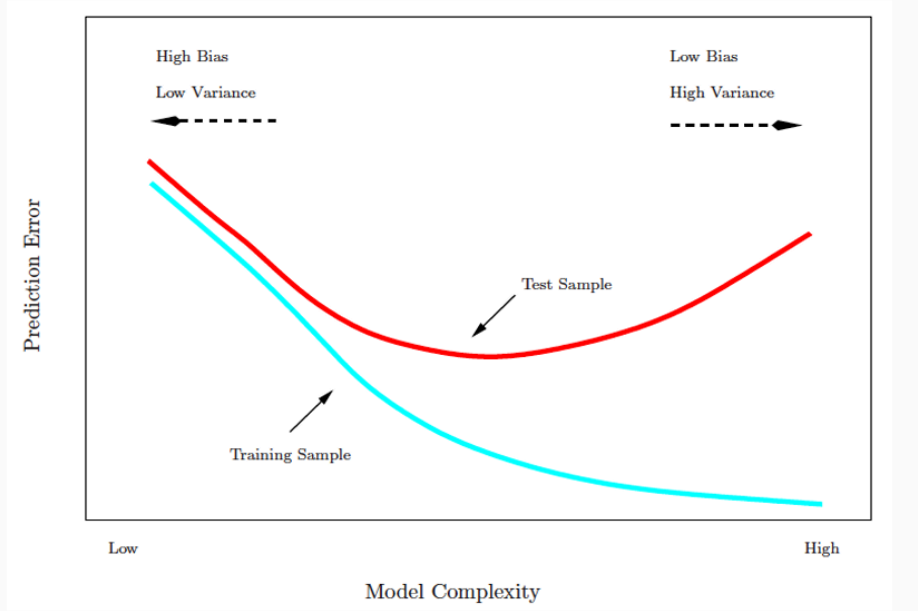
- Cross-validation;
  - Validation-set, Leave-one-out, K-fold
- Bootstrap.

Training error vs. test error

Richiamiamo la distinzione tra test error e training error:

- Il test error è l’errore medio che risulta dall’utilizzo di un metodo di statistical learning per predire la risposta su una nuova osservazione, una non usata nel training del metodo.
- Il training error può essere facilmente calcolato applicando il metodo di statistical learning alle osservazioni usate nel suo training.

> 💡 Il training error rate può spesso essere molto diverso dal test error rate e in particolare il training error rate tende a sottostimare drammaticamente il test error rate.



compromesso Bias-Varianza

All'aumentare della complessità del modello il training error tenderà sempre a decrescere, invece il test error calerà all'inizio (quando la riduzione del bias domina sull'aumento della varianza) ma aumenterà più avanti (quando l'aumento della varianza inizia a dominare).

Il test error rate è la chiave per la scelta di un metodo di learning.

Test error estimates

Migliore soluzione: un test set molto grande, questo spesso non è possibile.

Alcuni metodi effettuano delle correzioni matematiche al training error rate al fine di stimare il test error rate.

Alcuni di questi metodi di mathematical adjustment sono Cp statistic, AIC e BIC, li vedremo in seguito.

Ora consideriamo una classe di metodi che stimano il test error by holding out (trattenendo) un sottoinsieme delle osservazioni di training dal processo di fitting e poi applicando il modello di statistical learning a queste osservazioni che sono state trattenute.

Approccio Validation-set

Questo approccio consiste nel dividere randomicamente il set di campioni disponibili in due parti: un **training set** e un **validation set**, o hold-out set.

Il fit del modello viene effettuato sul training set, e il modello fitted è usato per predire le risposte per le osservazioni nel validation set.

Il **validation-set error** risultante fornisce una stima dell'errore di test.

Questo è tipicamente valutato usando:

- l'**MSE** nel caso di una risposta quantitativa;

- il **misclassification rate** (tasso di classificazione sbagliata) nel caso di una risposta qualitativa (discreta).

Questo schema mostra l'approccio validation set. Un set di n osservazioni sono divise randomicamente in un training set (blu) e un validation set (arancione). Il metodo di statistical learning viene fittato sul training set e le sue performance sono valutate sul validation set.

Example: Auto data

Supponiamo di voler predire mpg (miles per gallon) da horsepower (cavalli).

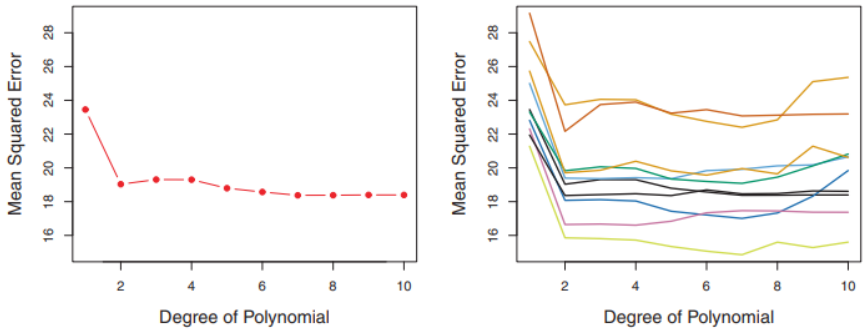
Effettuiamo quindi il fit di un modello di regressione polinomiale:

mpg ~ poly(horsepower, d) dove d è il grado del polinomio.

Quale grado da il miglior fit?

- dividiamo in maniera casuale il data set in dati di training e dati di validazione;
- effettuiamo il fit di diversi modelli con diversi gradi usando il training data set;
- valutiamo tutti i modelli fittati usando il validation data set;
- il modello con il più basso validation (testing) MSE è il vincitore.

Il modello finale sarà il modello vincitore fittato usando l'intero dataset.



L'approccio Validation Set è stato usato sul data set Auto al fine di stimare il test error che risulta dalla predizione di mpg usando funzioni polinomiali di horsepower.

A sinistra vediamo le stime di Validation Error per un singolo split in training e validation data sets.

A destra il metodo di validazione è stato ripetuto 10 volte, ogni volta usando uno split casuale differente delle osservazioni in training e validation set.

Questo illustra la variabilità nell'MSE di test stimato che risulta da questo approccio.

Validation-set: Drawbacks

L'approccio validation set è concettualmente facile ed è semplice da implementare ma ha due principali svantaggi:

1. Come mostrato nel grafico a destra, la stima di validation del test error rate può essere altamente variabile, sulla base di quali osservazioni precisamente sono incluse nel validation set.
2. Nell'approccio di validazione solo un sottoinsieme delle osservazioni, cioè quelle che sono incluse nel training set invece che nel validation set, sono usate per fare il fit del model. Visto che i metodi statistici tendono a performare peggio quando allenati su meno osservazioni, questo suggerisce che il validation set error rate potrebbe tendere a sovrastimare il test error rate per il modello fittato sull'intero dataset.

Questi due problemi sono approcciati da una versione rifinita dell'approccio validation-set chiamato cross-validation.

Leave-one-out cross-validation (LOOCV)

Validazione incrociata lasciandone uno fuori.

Si dividono i dati $\{(x_i, y_i)\}_{i=1}^n$ in:

- validation set: (x_1, y_1);
- training set: $\{(x_2, y_2), \ldots, (x_n, y_n)\}$.

Eseguiamo il fit del modello usando il training set.

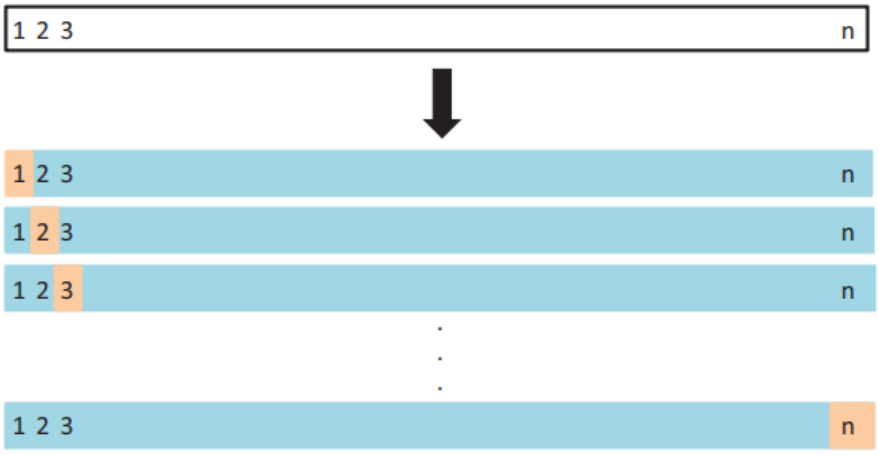
Validiamo il modello usando il validation set e calcoliamo il corrispondente test error $MSE_1 = (y_1 - \hat{y}_1)^2$.

Si ripete il processo n volte per ottenere n test errors.

La stima LOOCS per il test MSE è: $CV_{(n)} = \frac{1}{n} \sum_{i=1}^n MSE_i$.

$$\mathrm{CV}_{(n)} = \frac{1}{n} \sum_{i=1}^{n} \mathrm{MSE}_i.$$

Segue una rappresentazione schematica del LOOCV.



Un insieme di n data point è ripetutamente divisa in un training set (in blu) contenente tutte le osservazioni eccetto una e in un validation set (in arancione) contenente solo quell'osservazione. Il primo training set contiene tutte le osservazioni tranne l'1, il secondo training set contiene tutte le osservazioni tranne il 2 e così via. Il Test Error è poi stimato calcolando la media degli n MSE risultanti.

Leave-one-out cross-validation (LOOCV): vantaggi e svantaggi

L'approccio Leave-one-out cross-validation ha dei vantaggi rispetto all'approccio validation-set:

- LOOCV ha meno bias
  - questo ha senso in quanto facciamo ripetutamente il fit del metodo di statistical learning usando training data contenente n-1 osservazioni, cioè quasi tutto il data set è usato per il training.
- LOOCV produce un MSE meno variabile
  - l'approccio validation-set produce MSE differenti quando applicato ripetutamente a causa della casualità del processo di splitting (che tipicamente divide in due il data set);
  - effettuare LOOCV più volte produce sempre gli stessi risultati in quanto non c'è casualità nel processo di data split (1 osservazione per volta).

L'approccio LOOCV ha però un **significativo svantaggio**:

- LOOCV è computazionalmente intensivo, dispendioso, in quanto ogni modello deve essere fitted n volte.

k-fold cross validation

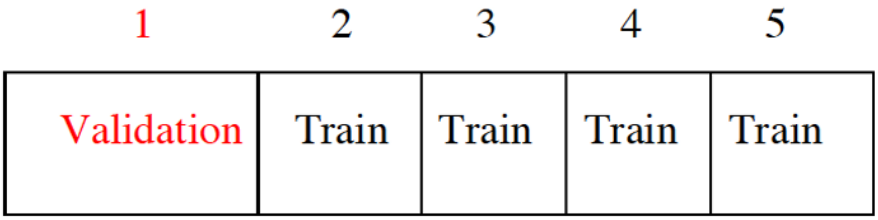
Una alternativa a LOOCV è k-fold CV, si tratta di un approccio ampiamente utilizzato per la stima del test error.

Le stime possono essere usate per selezionare il modello migliore e per dare un'idea del test error del modello finale scelto.

L'idea alla base di questo approccio è dividere in maniera casuale i dati in K parti di dimensione uguale.

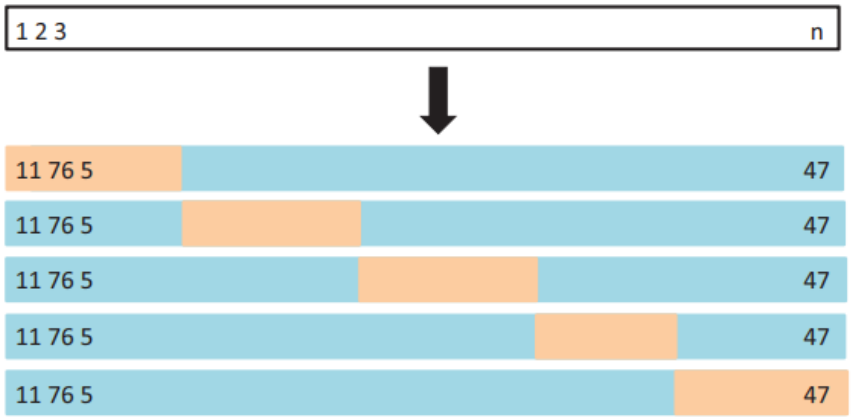
A questo punto mettiamo da parte la K-esima parte del dataset ed effettuiamo il fit del modello alle altre K-1 parti (combinate) e poi otteniamo previsioni per la K-esima parte che avevamo messo da parte.

Questa procedura è eseguita a turno per ognuna delle K parti e poi i risultati sono combinati.



In questa foto non sembra ma teoricamente tutte le parti dovrebbero essere di pari dimensioni.

Segue una rappresentazione schematica di una 5-fold Cross-Validation.



Un set di n osservazioni è randomicamente diviso in 5 gruppi non-overlapping (che non si sovrappongono).

A turno ognuno di questi quinti agisce da validation set (in arancione) e i rimanenti da training set (in blu). Il Test Error è stimato facendo la media delle 5 stime di MSE.

Siano le K parti $C_1, C_2, \ldots, C_K$, dove C_K denota gli indici delle osservazioni nella parte K. Ci sono n_k osservazioni nella parte k: se n è un multiplo di K allora $n_k = n/K$.

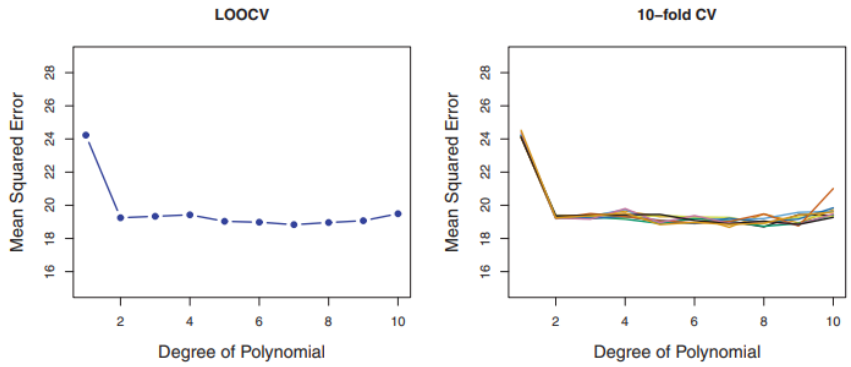
Calcoliamo:

$$CV_{(K)} = \frac{1}{K}\sum_{k=1}^{K} MSE_k = \frac{1}{K}\sum_{k=1}^{K}\sum_{i\in C_k}\frac{1}{n_k}(y_i - \hat{y}_i)^2$$

dove $MSE_k = \sum_{i\in C_k}\frac{1}{n_k}(y_i - \hat{y}_i)^2$ ed $\hat{y}_i$ è il fit per l'osservazione i, ottenuto dal data set con la parte k rimossa cioè dal training set.

Settando K=n si ritorna alla Leave-one-out cross-validation (LOOCV).

LOOCV vs. k-fold CV: Auto data example



La cross-validation è stata usata sul data set Auto al fine di stimare il test error che risulta dalla predizione di mpg usando funzioni polinomiali di horsepower.

A sinistra la error curve di LOOCV. Ha senso che sia una curva sola perché LOOCV è deterministico, non c'è casualità in quanto gli N split che si eseguono sono ottenuti prendendo ogni volta un elemento solo come test set uno per volta. Anche se lo avessimo fatto più volte (which we didn't) sarebbe comunque uscita la stessa curva.

A destra la 10-fold CV è stata eseguita 9 volte diverse, ogni volta con un diverso split casuale dei dati in 10 parti. La figura mostra le 9 CV error curves che sono leggermente differenti l'una dall'altra.

💡 LOOCV e K-fold CV sono entrambi stabili ma LOOCV è più computazionalmente intensivo.

Bias-variance trade-off per K-fold CV

IN BREVE

K-fold CV con $K < n$ ha un vantaggio computazionale rispetto a LOOCV, e spesso da una stima del test error rate più accurata rispetto a LOOCV a causa del compromesso tra Bias e Varianza.

BIAS

LOOCV ha bias più basso rispetto a K-fold, tuttavia LOOCV usa un training dataset (più grande) contenente $n-1$ osservazioni, mentre il training set di K-fold CV è composto da $\frac{(K-1)n}{K} < n-1$ osservazioni (ma ha comunque più osservazioni rispetto all'approccio validation-set).

VARIANZA

LOOCV ha varianza più alta di K-fold CV con $K < n$.

In LOOCV facciamo la media degli output di n modelli fittati, ognuno è trained su un, quasi identico, data set di osservazioni; questo vuol dire che questi output sono altamente (positivamente) correlati tra loro.

In contrasto con K-fold CV (K<n), facciamo la media degli output di K modelli fitted, che sono meno correlati, visto che l'overlap tra i training set di ogni modello è più piccolo.

💡 Visto che la media di molte quantità altamente correlate ha una varianza più alta rispetto alla media di tante quantità che non sono altrettanto correlate, la stima del test error di LOOCV tende ad avere varianza più alta rispetto a quella di K-fold CV.

CONCLUSIONE

C'è un compromesso bias-varianza associato alla scelta di k in K-fold cross validation. Tipicamente, date queste considerazioni, si sceglie $k = 5$ o $k = 10$ poiché questi valori hanno dimostrato empiricamente di restituire stime del test error rate che non soffrono nè di eccessivo bias nè di eccessiva varianza.

Il bootstrap

Il bootstrap è un tool statistico ampiamente applicabile ed estremamente potente che può essere utilizzato per quantificare l'incertezza associata ad un dato stimatore o metodo di statistical learning.

Ad esempio può fornire una stima dello standard error di un coefficiente di una regressione lineare, oppure un confidence interval per quel coefficiente, ovviamente per lo standard error non è utile in quanto molti software statistici come R restituiscono questi dati in automatico.

Tuttavia, il potere di bootstrap sta nel fatto che può essere applicato, semplicemente, ad un ampia gamma di metodi di statistical learning; inclusi alcuni per i quali una misura di variabilità è, altrimenti, difficile da ottenere e non è automaticamente restituita da un software statistico.

Toy model

Illustriamo il bootstrap su un toy example nel quale vogliamo determinare la migliore allocazione per un investimento con un semplice modello.

Nello specifico esploriamo l'utilizzo di bootstrap per stabilire la variabilità associata ai coefficienti di regressione nel fit di un modello lineare.

Supponiamo di voler investire una somma fissa di denaro in due asset finanziari che restituiscono ritorni di investimento X ed Y, dove X ed Y sono quantità casuali. Investiremo una frazione di α del nostro denaro in X e investiremo la restante quantità $1 - \alpha$ in Y.

Visto che c'è variabilità associata al ritorno di investimento di questi due assets desideriamo scegliere α tale da minimizzare il rischio totale, o la varianza, del nostro investimento.

In altre parole vogliamo minimizzare $Var(\alpha X + (1 - \alpha)Y)$.

Si può dimostrare (noi non lo abbiamo fatto) che il valore di α che minimizza il rischio è dato da

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}},$$

dove $\sigma_X^2 = Var(X), \sigma_Y^2 = Var(Y)$ *e* $\sigma_{XY} = Cov(X, Y)$.

In realtà le quantità σ_X^2, σ_Y^2, σ_{XY} sono sconosciute.

Possiamo calcolare le stime per queste quantità, cioè $\hat{\sigma}_X^2$, $\hat{\sigma}_Y^2$, $\hat{\sigma}_{XY}$, usando un data set che contiene le misurazioni passate per X ed Y.

Possiamo poi stimare il valore di α che minimizza la varianza del nostro investimento usando.

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}.$$

Nella figura 5.9 è illustrato questo approccio per la stima di α su un data set simulato. In ogni panel abbiamo simulato 100 coppie di ritorni di investimento per X ed Y.

Abbiamo usato questi ritorni per stimare σ_X^2, σ_Y^2, σ_{XY}, queste stime le abbiamo usate per stimare α. Il valore di $\hat{\alpha}$ ottenuto da ogni data set simulato va da 0.532 a 0.657.

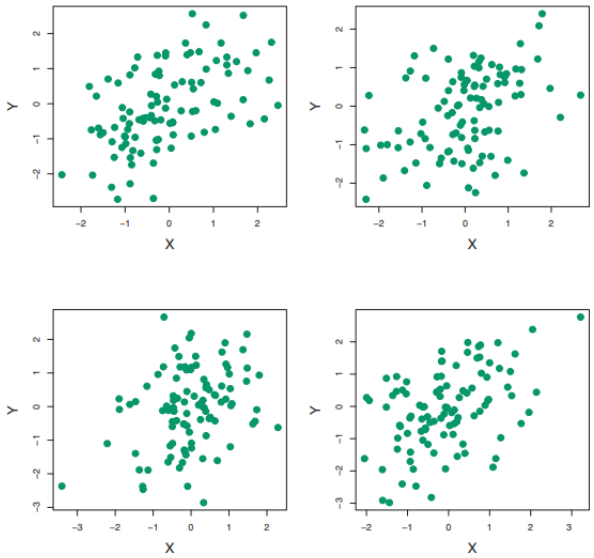


FIGURE 5.9. *Each panel displays* 100 *simulated returns for investments* X *and* Y. *From left to right and top to bottom, the resulting estimates for* α *are* 0.576, 0.532, 0.657, *and* 0.651.

Risulta naturale stimare l'accuratezza delle nostre stime di α. Per stimare la deviazione standard di $\hat{\alpha}$ ripetiamo il processo di simulazione di 100 osservazioni a coppie di X ed Y, a questo punto stimiamo α usando l'equazione che abbiamo visto per $\hat{\alpha}$ 1000 volte.

Possiamo chiamare le 1000 diverse stime di α che abbiamo ottenuto: $\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_{1000}$.

Per queste simulazioni i parametri sono stati settati a

$\sigma_X^2 = 1, \sigma_Y^2 = 1.25, \sigma_{XY}^2 = 0.5$
e quindi sappiamo che il valore vero di
α è 0.6.

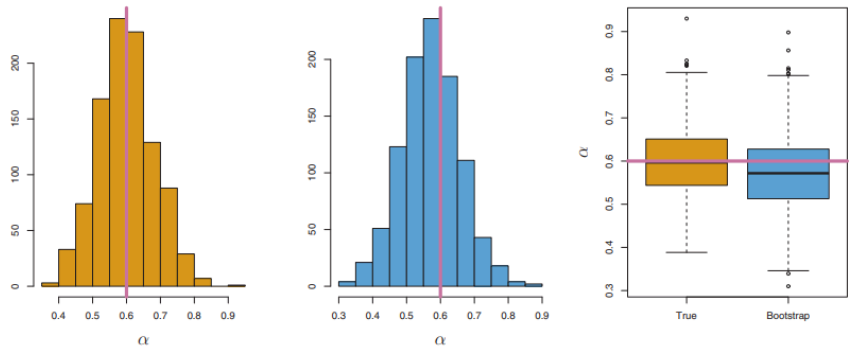
La media delle mille stime di α è

$$\bar{\alpha} = \frac{1}{1,000} \sum_{r=1}^{1,000} \hat{\alpha}_r = 0.5996,$$

che è molto vicino a $\alpha = 0.6$ e la deviazione standard sarà

$$\sqrt{\frac{1}{1,000 - 1} \sum_{r=1}^{1,000} (\hat{\alpha}_r - \bar{\alpha})^2} = 0.083.$$

Questo fornisce una stima molto buona dell'accuratezza di $\hat{\alpha}$: $SE(\hat{\alpha}) \approx 0.083$. Quindi per un campione casuale dalla popolazone ci aspettiamo che $\hat{\alpha}$ differisca da α approssimativamente di 0.08 in media.



A sinistra un istogramma delle stima di α ottenute generando 1000 data set simulati (ognuno da 100 coppie) dalla vera popolazione.

Al centro un istogramma delle stime di α ottenute da 1000 campionamenti bootstrap da un singolo data set.

A destra le stime di α mostrata nei panel a sinistra e al centro sono mostrati come boxplots. In ogni panel la linea rosa indica il vero valore di α.

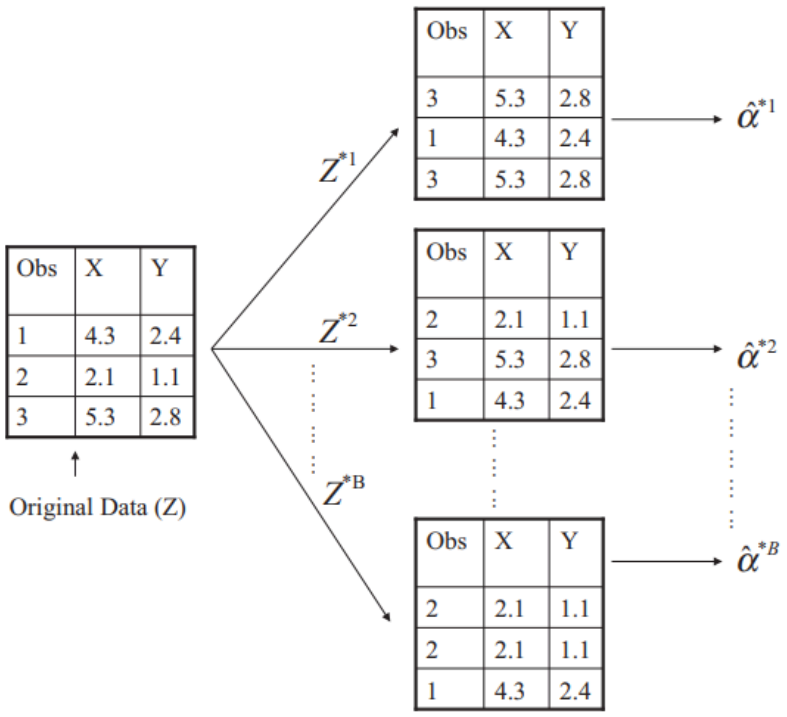
La procedura bootstrap

In pratica però, la procedura di stima dell'$SE(\hat{\alpha})$ illustrata non può essere applicata in quanto per i dati veri non possiamo generare nuovi samples dalla popolazione originale.

Però, l'approccio bootstrap permette di usare un computer per simulare il processo di nuovi insiemi di campioni in modo da permettere di stimare la variabilità di $\hat{\alpha}$ senza generare campioni aggiuntivi.

💡 Piuttosto che ripetutamente ottenere, dalla popolazione, dei data set indipendenti, si procede a ottenere data set distinti tramite il ripetuto campionamento di osservazioni dal data set originale.

Ognuno di questi "bootstrap data sets" è creato facendo il campionamento con replacement (dopo che estrai il numero lo rimetti nel bussolotto) ed ha la stessa dimensione del nostro dataset originale. Di conseguenza alcune osservazioni potrebbero apparire più di una volta in un dato data set bootstrap mentre altri potrebbero non apparire affatto.



non per forza vanno generati così i diversi dataset bootstrap, a seconda dei dati si sceglie la strategia migliore

Questa è una illustrazione grafica dell'approccio bootstrap su un piccolo sample contenente n=3 osservazioni. Ogni bootstrap data set contiene n osservazioni, campionate con replacement dal data set originale. Ogni bootstrap data set è usato per ottenere una stima di α.

PROCEDURA

Denotiamo il primo data set bootstrap come Z^{*1}, usiamo Z^{*1} per produrre una nuova stima bootstrap per α che chiameremo $\hat{\alpha}^{*1}$.

Questa procedura è ripetuta B volte per un certo valore grande di B (come 100 o 1000), al fine di produrre B diversi data set bootstrap, Z^{*1}, ..., Z^{*B}, e B stime di α corrispondenti, $\hat{\alpha}^{*1}$, ..., $\hat{\alpha}^{*B}$.

Stimiamo lo standard error di $\hat{\alpha}$ usando la formula

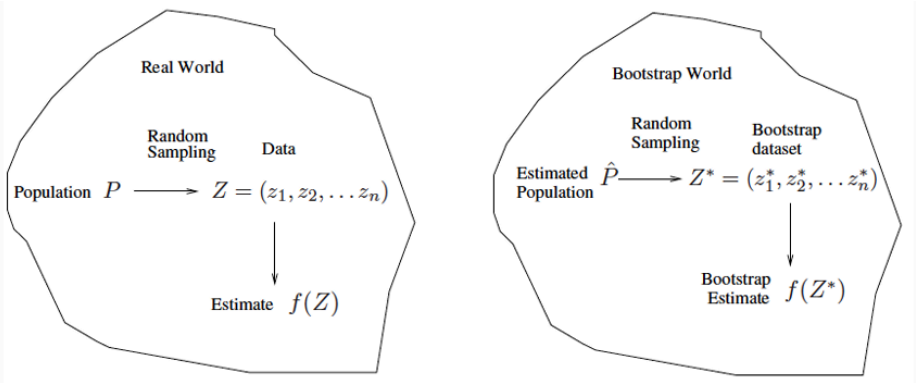
$$\mathrm{SE}_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}\left(\hat{\alpha}^{*r} - \frac{1}{B}\sum_{r'=1}^{B}\hat{\alpha}^{*r'}\right)^2}.$$

Questo fornisce una stima dello standard error di $\hat{\alpha}$ stimato dal data set originale. Per questo esempio $SE(\hat{\alpha}) = 0.087$, molto vicino alla stima di 0.083 ottenuta usando 1000 data set simulati.

Il Bootstrap in generale

In situazioni con dati più complessi, capire il modo appropriato per generare campioni bootstrap può richiedere una certa quantità di ragionamento.

Ad esempio, se i dati sono una serie temporale non possiamo campionare le osservazioni con replacement; possiamo invece creare blocchi di osservazioni consecutive e campionare quei blocchi con replacement. Poi incolliamo insieme i blocchi campionati per ottenere un dataset bootstrap.



Usi di bootstrap

Bootstrap è principalmente utilizzato per ottenere una **stima dello standard error.**

Bootstrap fornisce anche intervalli di confidenza approssimati per un parametro di popolazione. Nell'esempio toy i percentili al 5% e 95% dei 1000 valori sono (0.43, 0.72).

Questo rappresenta un intervallo di confidenza approssimato al 90% per il vero α.

L'intervallo di cui sopra è chiamato Bootstrap Percentile confidence interval ed è il metodo più semplice (tra tanti) per ottenere un intervallo di confidenza a partire da Bootstrap.

Osservazioni finali

Se siamo in una situazione **data-rich**, l'approccio migliore per risolvere i problemi di Model Selection e Model Assessment è: dividere il dataset in maniera casuale in tre parti:

- un **training set**, usato per effettuare il fit del modello;
- un **validation set**, usato per stimare il prediction error per la model selection;
- un **test set**, usato per l'assessment del generalization error del modello finale scelto.

Idealmente il test set dovrebbe essere prodotto solo ai fini della data analysis. Se invece di usiamo il test-set ripetutamente, scegliendo il modello con il minor test-set error, finiremo per sottostimare il vero test error.

Risulta difficile fornire una regola generale su come scegliere il numero di ossservazioni in ognuna delle tre parti.

Una divisione tipica può essere: 50% per il training set, 25% per il validation set e 25% per il test set.

In molte situazioni non si dispone di abbastanza dati per dividerli in 3 parti, di conseguenza il dataset è diviso in due parti: training set e test set.

Anche in questo caso è troppo difficile dare una regola generale di quanti dati di training sono abbastanza, tra le altre cose questo dipende dal signal-to-noise ratio della underlying function to fit (dal rapporto segnale rumore della funzione vera che noi stiamo approssimando facendo il fit).

In queste circostantanze lo step di validazione è approssimativamente o analitico (usando AIC, BIC, ...) o per riutilizzo efficiente dei campioni (cross-validation).

Esperimento - riassunto dall'AI di Notion

Key Points

- I metodi di resampling permettono di ottenere informazioni aggiuntive sul modello attraverso il ricampionamento ripetuto dal training set
- Il bootstrap consente di stimare la variabilità dei parametri senza generare nuovi campioni, utilizzando il campionamento con reinserimento
- Gli obiettivi principali sono il model assessment (quantificare l'incertezza) e il model selection (scegliere tra modelli alternativi)
- In situazioni data-rich, il dataset viene diviso in training (50%), validation (25%) e test set (25%)
- Con dati limitati si utilizza solo la divisione in training e test set, supportata da metodi analitici o cross-validation