IMAD

2024-2025

Predizione, filtraggio e smoothing

Dati due processi stocastici stazionari $x(t)$ e $y(t)$ di cui uno solo osservabile (y), analizziamo il problema di stimare il processo ignoto x a partire da quello noto y. Il problema ha due casi:

- *caso banale*: i due processi sono uguali
- *caso comune*: $y(t) = x(t) + e(t)$, $e(t) \sim WN(0, \lambda^2)$, ovvero misuriamo tramite y una versione rumorosa di x

Vogliamo ottenere una stima a posteriori di x a tempi dato il valore ad un tempo dato, dato y. Se la stima avviene per tempi successivi a quello dell'osservazione, si parla di *predizione*. Se la stima avviene per un tempo uguale al tempo dell'osservazione, si parla di *filtraggio*. Se infine si stimano tempi precedenti a quello dell'osservazione di riferimento, si parla di *smoothing*. Quest'ultimo, al contrario degli altri due, non è effettuabile in tempo reale.

Analizziamo il caso della *predizione a k passi*. Consiste nel cercare di ottenere $\hat{x}(t|t-k)$, ovvero stimare $x(t)$ date le informazioni al più fino al tempo $t-k$ incluso. Si tratta, dunque, di una stima per istanti futuri a quello dell'osservazione. La stima per il momento presente data un'osservazione passata corrisponde, a livello di formula, alla stima futura data un'osservazione presente ($\hat{x}(t+k|t)$), purché k sia il medesimo. Questa proprietà è valida solo per i processi stazionari. Si considerino ora i due casi rimanenti, filtraggio e *smoothing.* Il filtraggio consiste nella stima di $\hat{x}(t|t)$, ovvero ottenere una stima di x all'istante corrente, pulita dal rumore. L'operazione ha senso solo se la misurazione $y(t)$ è diversa dal vero segnale $x(t)$ a causa di errori. Lo *smoothing* invece è l'operazione di stima $\hat{x}(t|t+k)$, che ripulisce il segnale a tempi passati, al fine di ricostruire la traiettoria del segnale ripulita dal rumore. Esattamente come il filtraggio, anche lo *smoothing* ha senso solo se la misurazione è diversa dal vero segnale a causa di errori.

Predizione ottima

Analizziamo la stima di $y(t)$ come $\hat{y}(t|t-k)$ sapendo che $x(t) = y(t)$. Dato che y è un processo stocastico, anche il predittore $\hat{y}$ basato sui valori passati di y è a sua volta un processo stocastico. Anche l'errore di predizione è un processo stocastico per la medesima ragione, e si definisce come:

$$\varepsilon_k(t) = y(t) - \hat{y}(t|t-k).$$

È desiderabile ottenere *predittori lineari ottimi*, che abbiano cioè un errore di predizione a MSE minimo. Consideriamo i predittori ottimi per ARMA e ARMAX, escludendo gli altri tipi di modelli, perché a fine analisi si arriva ad un'espressione generale valida anche per le altre famiglie. Riprendendo quanto spiegato in precedenza, $\hat{y}(t|t-k)$ e $\hat{y}(t+k|t)$ potrebbero avere valori diversi a livello numerico, ma sono caratterizzati dalla stessa formula del predittore ottimo.

Un processo stocastico stazionario si definisce *completamente predicibile* se esistono coefficienti per la formula:

$$y(t) = \sum_{i=1}^{+\infty} a_i y(t-i)$$

che permettano di prevedere $y(t)$ senza alcun errore. Un processo completamente predicibile è caratterizzato da una formula della densità spettrale di potenza nella seguente forma:

$$\Gamma_{yy}(\omega) = \sum_i \alpha_i \delta(\omega - \omega_i)$$

dove δ è detto *delta di Dirac* e produce una rappresentazione puntuale di una singola frequenza sinusoidale sul grafico. La densità spettrale di potenza di un processo completamente predicibile è dunque una combinazione lineare di delta di Dirac. Per contro, il segnale completamente impredicibile, il rumore bianco, ha densità spettrale di potenza costante, non rappresentabile come combinazione lineare di un numero finito di termini puntuali.

Un esempio è una somma di funzioni sinusoidali / cosinusoidali a coefficienti casuali incorrelati. con $\mathbb{E}[v_1] = \mathbb{E}[v_2]$ e $\mathrm{Var}[v_1] = \mathrm{Var}[v_2] = \sigma^2$:

$$y(t) = v_1 \sin(\bar{\omega}t) + v_2 \cos(\bar{\omega}t).$$

Il processo y è stazionario, non ergodico, con funzione di autocovarianza cosinusoidale. La densità spettrale di potenza è perfettamente a spilli. Dopo aver stimato i coefficienti, il processo diventa perfettamente predicibile in modo deterministico e periodico.

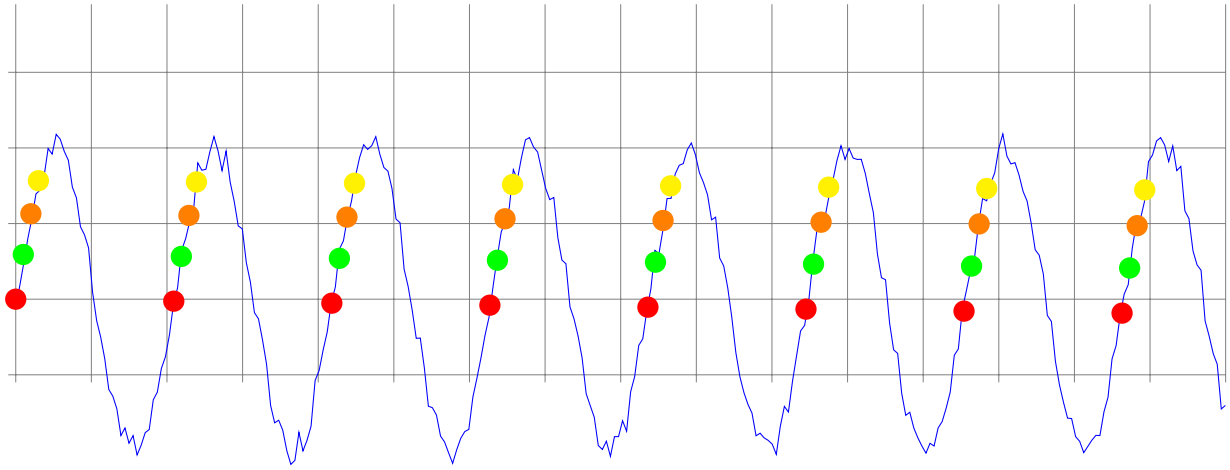
Scomposizione di Wold

Possiamo scrivere ogni processo stocastico stazionario come somma di due componenti totalmente incorrelate, una completamente predicibile e l'altra completamente stocastica:

$$y(t) = \bar{y}(t) + y_p(t)$$

$\bar{y}(t)$ tale che $\bar{y}(t) = \sum_{i=0}^{+\infty} c_i e(t-i)$, con e che è una realizzazione del rumore bianco.

La parte puramente stocastica può essere interpretata come un $MA(\infty)$. Anche la densità spettrale di potenza si può scomporre come la somma di un grafico a spilli, dovuto alla parte deterministica, e un grafico continuo, dovuto alla parte stocastica. Nella pratica la scomposizione fornisce una linea guida per stimare modelli di serie temporali. Per effettuare tale stima, bisogna innanzitutto osservare i dati o lo spettrogramma per riconoscere le componenti periodiche. Successivamente si stimano, manualmente o ai minimi quadrati, le componenti $y_p(t)$ necessarie per ottenere $\hat{y}_p(t)$. Dopodiché è possibile estrarre la componente puramente stocastica come differenza tra il segnale complessivo $y(t)$ e la componente deterministica appena stimata. Per effettuare predizioni, si predicono separatamente la componente deterministica e quella stocastica, e se ne sommano le predizioni. In particolare, la stima dei *trend* sinusoidali può essere effettuata riconoscendo innanzitutto il periodo, e poi campionandone l'ampiezza a intervalli costanti su un numero finito di periodi, per calcolarne la media, come mostrato in figura:



In questo caso calcoleremo la media degli 8 punti rossi, degli 8 punti verdi, ecc. per avere un campionamento dell'ampiezza della sinusoide in vari punti e ricavarne il *trend*.

Come ipotesi di lavoro, assumiamo che il processo $MA(\infty)$ possa essere approssimato da processi a spettro razionale, per poter impiegare modelli ARMA o simili. Esiste una categoria di modelli detti SARMA, che sono modelli ARMA in grado di stimare direttamente anche la stagionalità senza bisogno di essere scomposti manualmente. Non è sempre facile estrarre le componenti deterministiche dal periodogramma. Questo perché gli stimatori della densità spettrale di potenza tendono a non essere buoni, e le risonanze nel grafico potrebbero non essere delta di Dirac ma picchi dovuti ai poli della funzione di trasferimento.

Definiamo *filtro passa-tutto* un filtro che non filtra, cioè che lasci passare tutte le componenti. La sua funzione di trasferimento è:

$$T(z) = \frac{1}{a} \cdot \frac{z+a}{z+\frac{1}{a}}, \quad a \neq 0, a \in \mathbb{R}.$$

Calcolando la densità spettrale di potenza del segnale in uscita quando il segnale in ingresso è un processo stocastico stazionario, ci si accorge che è la stessa del segnale in ingresso. Il segnale in uscita non è però identico, perché il filtro causa uno sfasamento.

In alcuni casi la matrice può non essere di rango pieno. Solitamente la causa è un eccesso di regressori rispetto al numero di dati effettivi. Questo può avvenire quando abbiamo a disposizione diverse misure che in realtà fanno riferimento alla stessa grandezza, ad esempio due lunghezze espresse una in metrico, l'altra in imperiale. L'eccesso di regressori può essere corretto rimuovendo manualmente quelli ridondanti, oppure applicando la regolarizzazione.

A livello computazionale, è comunque possibile invertire una matrice di rango non pieno utilizzando la tecnica della pseudoinversa. Per ottenerla in Matlab, bisogna chiamare la funzione `pinv()`.

Il metodo delle *normal equations* diventa lento per dati di grandi dimensioni. Questo perché l'inversione di matrice è un'operazione onerosa che scala male. Esistono metodi numerici iterativi in sostituzione alle normal equations. Uno di essi è il *gradient descent*, che si utilizza per minimizzare le funzioni differenziabili. La sua formula iterativa è:

$$\hat{\theta}^{(k+1)} = \theta^{(k)} - \alpha \cdot \left.\frac{\partial J(\theta)}{\partial \theta}\right|_{\theta=\hat{\theta}^{(k)}}$$

Nel caso multivariabile, l'equazione diventa:

$\hat{\theta}^{(k+1)} = \theta^{(k)} - \alpha \cdot \nabla J(\theta)|_{\theta=\hat{\theta}^{(k)}}\$$

L'algoritmo segue il gradiente per farsi portare fino ai punti di minimo. Non convergendo mai perfettamente, va fermato imponendo un criterio. Solitamente si definisce un numero massimo di passaggi, oppure una soglia minima di variazione del gradiente. Dato che il metodo analitico con le matrici è comunque soggetto a imprecisioni dovuti alle approssimazioni nel calcolo della matrice inversa, i metodi iterativi non sono necessariamente meno precisi rispetto al metodo esatto. Notiamo inoltre che il *gradient descent* è velocizzato, a livello di calcolo, se applicato a matrici normalizzate colonna per colonna.

Modelli lineari e verosimiglianza

Consideriamo di voler costruire un modello lineare utilizzando la stima ai minimi quadrati, ovvero stimando parametri (coefficienti angolari e intercetta) che minimizzino lo scarto quadratico totale tra ogni punto reale e il corrispettivo punto predetto dal modello per gli stessi ingressi. Se il sistema è veramente lineare, e se il rumore è incorrelato, a media nulla e a varianza definita, allora lo stimatore a minimi quadrati risulta corretto e consistente.

Supponiamo che le osservazioni utilizzate per la costruzione del modello siano indipendenti e identicamente distribuite, e che seguano una distribuzione gaussiana. La funzione di distribuzione di probabilità congiunta è data dal prodotto delle singole distribuzioni di ogni punto, e rappresenta la probabilità che si realizzi lo specifico vettore di dati osservato. La distribuzione congiunta è multivariata quando è funzione dell'ingresso noto, come nel caso considerato. Si potrebbero stimare le uscite avendo a disposizione media e varianza della distribuzione. Non avendole a disposizione, bisogna dapprima stimarle con il metodo della massima verosimiglianza. Esso consiste nella massimizzazione di una funzione di verosimiglianza per ottenere valori di media e varianza che più permettono di avvicinare i dati stimati ai dati reali, a parità di ingresso. Costruiamo dunque una funzione di verosimiglianza da massimizzare, corrispondente alla distribuzione di probabilità congiunta parametrizzata per media e varianza. Spesso è conveniente massimizzare il logaritmo naturale della verosimiglianza. Il logaritmo, essendo monotono crescente, conserva l'ascissa dei punti di massimo e di conseguenza non altera l'arg max che stiamo cercando di stimare. Dato che le distribuzioni da noi considerate sono gaussiane, basate su esponenziali, l'applicazione del logaritmo semplifica i calcoli. Nel caso considerato, la massimizzazione si effettua per *gradient descent* perché è molto complesso determinare la forma chiusa necessaria per una massimizzazione standard. Otteniamo uno stimatore consistente, asintoticamente corretto, asintoticamente efficiente ed asintoticamente normale. Massimizzare la log-verosimiglianza è come minimizzare il suo opposto: è per questo che possiamo applicare il *gradient descent* a questo problema di massimizzazione nonostante si tratti di una tecnica di minimizzazione. Esistono anche altre tecniche numeriche di ottimizzazione alternative al *gradient descent*, dette *gradient free* perché funzionano su funzioni non differenziabili.

Generalized Linear Models

Il metodo della massima verosimiglianza può essere usato per creare diversi tipi di modelli lineari. La regressione lineare è solo uno dei possibili esempi. Essa assume che il rumore sia distribuito secondo una gaussiana intorno alla media rappresentata dalla retta di classificazione. Stimando media e varianza si ottengono i parametri del modello di regressione lineare. Ripetendo lo stesso processo, ma cambiando le ipotesi sulla distribuzione del rumore, possiamo ottenere nuove funzioni di costo la cui minimizzazione genera altri tipi di modelli. Ad esempio, se consideriamo i dati come distribuiti in due classi secondo una bernoulliana, con la retta del modello lineare come spartiacque, otteniamo un modello di classificazione noto come regressione logistica. Esiste anche un modello chiamato regressione di Poisson, basato sull'omonima distribuzione. La classe di modelli accomunata dall'essere lineari, basati sul metodo della stima a massima verosimiglianza e distinti dall'avere diverse distribuzioni come ipotesi è chiamata *Generalized Linear Models*.

Modelli di classificazione e regressione logistica

La classificazione non può essere effettuata con modelli di regressione. Non è possibile semplicemente codificare le classi come numeri da utilizzare in uscita, perché questo introduce implicitamente un concetto di ordine e di distanza che non ha senso di esistere in questo ambito. La numerazione delle classi sarebbe inoltre completamente arbitraria, e produrrebbe modelli non univoci. Possiamo rimuovere questo secondo problema creando un classificatore puramente binario, ma in ogni caso la stima potrebbe cadere al di fuori del range di valori utilizzato e produrre quindi risultati non interpretabili. Introduciamo dunque una funzione sigmoide (detta anche logistica) che permette di saturare l'uscita del modello lineare nell'intervallo tra 0 e 1. Possiamo interpretare questo valore come la probabilità che il campione appartenga alla classe 1. È così che si ottiene, a livello pratico, il modello di regressione logistica descritto in precedenza.

Si costruisca ora la funzione di costo per la regressione logistica. Si supponga di avere un insieme di osservazioni bernoulliane (appartenenze a due possibili classi, 0 e 1) indipendenti e identicamente distribuite. Interpretiamo l'uscita del modello logistico come probabilità che l'ingresso appartenga alla classe 1. Ogni dato dell'insieme è una realizzazione della distribuzione. I passaggi da affrontare sono: 1. calcolo della meno-log-verosimiglianza 2. calcolo del gradiente 3. ottimizzazione per determinare il minimo La funzione di verosimiglianza è il prodotto delle formule di Bernoulli per le singole osservazioni. Applichiamo l'opposto del logaritmo naturale, per ottenere una formula che sia una semplice sommatoria di termini. Si osservi che nel caso monocampione la funzione di costo è quasi nulla se il valore reale e la predizione sono entrambi 1, e tende invece all'infinito quando il valore predetto è 1 e quello reale è 0. Questo è coerente con l'obiettivo che vogliamo ottenere, e ci rassicura sulla bontà della funzione di verosimiglianza. Come nel caso della regressione lineare, non è possibile effettuare una minimizzazione in formula chiusa, quindi ci si basa sull'algoritmo *gradient descent*. Il modello ottenuto permette di classificare nuovi punti in base al valore dell'uscita del modello, solitamente utilizzando il valore 0.5 come soglia.

Il modello è lineare, perché sono lineari i suoi parametri ma la stessa assunzione non necessita di essere effettuata sul regressore. Esso può essere qualsiasi, anche non lineare, a patto di mantenere lineari i parametri. Possiamo dunque utilizzare modelli lineari per rappresentare sistemi non lineari. Questo non è sempre facile perché non è possibile determinare algoritmicamente quale sia il tipo migliore di trasformazione non lineare da applicare al regressore. Si testa dunque una serie di funzioni standard. Anche trovando un modello buono per i dati di addestramento, però, non è detto che il modello sia riutilizzabile per nuovi dati.

Fondamenti di Machine Learning

È sensato applicare algoritmi di *machine learning* soltanto quando i dati a disposizione sembrano mostrare un *pattern* di cui sia ignota la funzione analitica. L'ambito del *machine learning* si distingue dalla *data science*. Il primo cerca di ottenere la migliore predizione possibile su nuovi dati a partire dal modello creato dai dati noti. La seconda, invece, cerca di capire i dati e le loro relazioni interne. Dato che l'obiettivo del *machine learning* è trovare un modello, è importante definire i criteri di accettazione di tale modello. È necessario imporre una soglia minima di accuratezza, e valutare le prestazioni dei modelli ottenuti su un insieme di dati diverso da quello impiegato per l'addestramento. Questo secondo insieme è chiamato *insieme di validazione*. La fase di addestramento, invece, è anche nota come *identificazione* nell'ambito dei modelli dinamici.

Si riprenda la differenza tra apprendimento supervisionato e apprendimento non supervisionato. Le componenti dell'apprendimento supervisionato sono: un vettore di regressori o *feature*; una funzione obiettivo, che è la formula ideale ignota; dei dati, ovvero un insieme di coppie regressore-uscita. Il modello generato dall'addestramento appartiene ad una famiglia di possibili modelli, chiamata *spazio delle ipotesi*. Si parla di apprendimento supervisionato perché l'uscita corretta è nota e viene utilizzata per aggiustare i parametri del modello. Con l'apprendimento supervisionato si possono creare sia modelli di regressione, a uscita continua, che di classificazione, a uscita discreta. L'apprendimento non supervisionato, invece, cerca un criterio secondo il quale raggruppare i dati, senza apprendere funzioni obiettivo. L'insieme dei dati di addestramento contiene soltanto i regressori, perché non è specificata un'uscita desiderata. Si utilizza dunque per esplorare le proprietà dei dati. Esiste un'ulteriore categoria di apprendimento, detta *apprendimento per rinforzo*, in cui i dati sono composti da terne ingresso-uscita-ricompensa. In questo caso, il modello tenta varie strategie per raggiungere un obiettivo, ed è ricompensato o punito in base all'efficienza con cui ci si avvicina. Questa tecnica è principalmente impiegata in robotica.

Per determinare i parametri di un modello ad apprendimento supervisionato è necessario costruire una funzione di costo, dipendente dai parametri del modello. Essa deve essere poi minimizzata per ottenere i valori più probabili per i parametri. Una possibile misura dell'errore è lo scarto quadratico medio tra l'uscita corretta e quella predetta dal modello. È necessario distinguere tra due tipologie di errori. La prima è l'errore *in-sample*, definito come errore nella predizione sui dati di addestramento. L'errore *out-of-sample*, invece, si considera sull'intero dominio dei possibili dati, esclusi quelli di addestramento, ed è dunque impossibile da stimare con esattezza. L'errore di

validazione, ovvero lo scarto nella predizione sui dati dell'insieme di validazione, è un'approssimazione dell'errore *out-of-sample*.

$$e = \sum_i (\hat{y}_i - y_i)^2$$

A livello teorico il problema dell'apprendimento dai dati è mal posto. Si può infatti ricondurre al problema filosofico dell'induzione, ovvero della generalizzazione a partire da un numero finito di osservazioni. Un modello potrebbe essere in grado di azzerare completamente l'errore *in-sample*, spiegando perfettamente i dati di addestramento, ma sbagliare completamente per i campioni *out-of-sample.* Non è dunque possibile determinare con esattezza una funzione continua a partire da un numero discreto di campioni. È tuttavia possibile massimizzare la probabilità di avere stimato la funzione corretta. Questo è l'obiettivo della *teoria della generalizzazione*, che si occupa di studiare i casi nei quali sia possibile generalizzare in modo non erroneo. Da essa apprendiamo l'utilità di studiare il compromesso tra approssimazione e generalizzazione, sempre mantenendo l'obiettivo di poter affermare con alta confidenza che l'errore *out-of-sample* sia piccolo. È utile a tal fine effettuare una decomposizione *bias-varianza*. Definiamo *bias* l'errore derivante dalla scelta dello spazio di ipotesi, come distanza tra la migliore funzione nello spazio delle ipotesi e l'effettiva funzione obiettivo. Utilizzare un modello lineare per rappresentare dati provenienti da una distribuzione quadratica, ad esempio, genera un alto *bias*. La varianza, invece, è lo scarto quadratico tra il modello ottenuto e il migliore possibile modello all'interno dello spazio delle ipotesi scelto. Misura dunque quanto l'ipotesi finale differisca dalla migliore ipotesi. Tale ipotesi migliore è definita come il modello medio, ovvero il modello i cui parametri siano la media tra i parametri di tutti i modelli che si possano ottenere dai diversi insiemi di dati appartenenti al dominio. Esso non è dunque limitato dall'insieme dei dati di addestramento, ma soltanto dalla scelta dello spazio delle ipotesi.

$$e_{out} = bias^2 + varianza$$

In caso di dati senza rumore, l'errore *out-of-sample* può essere descritto con una semplice funzione di costo quadratica. Possiamo scomporlo, come spiegato in precedenza, nella somma tra la varianza e il quadrato del *bias*. Il *bias* è concettualmente simile all'MSE dei modelli lineari stimati a massima verosimiglianza. La varianza è interpretabile come la sensibilità del modello alla specifica realizzazione dell'insieme dei dati. È impossibile minimizzare contemporaneamente entrambe le metriche, perché la riduzione di una avviene a scapito dell'altra. È però possibile trovare, almeno approssimativamente, il migliore compromesso che minimizzi la loro composizione, ovvero l'errore *out-of-sample*.

Indicativamente, per avere una buona probabilità di generalizzare, il numero di dati deve essere più di 10 volte il numero di parametri del modello. La complessità del modello deve seguire il numero di dati e non la complessità della funzione obiettivo. L'errore *in-sample* tende a ridursi con il numero di parametri. L'errore *out-of-sample* è generalmente elevato per un numero basso di parametri, ha un punto di minimo definito, e oltre tale soglia riprende a crescere. Possiamo rappresentare una stima dei due tipi di errori al variare del numero di dati di addestramento, su un grafico cartesiano chiamato *learning curve*. Dividiamo i dati in un n sottoinsiemi e, alternatamente, ne utilizziamo uno per la validazione e tutti gli altri $n-1$ per l'addestramento. In un modello semplice con pochi parametri l'errore *out-of-sample* e l'errore *in-sample* tendono a convergere simmetricamente ad un valore medio. Nei modelli più complessi la curva dell'*out-of-sample* parte da un valore molto maggiore, e converge verso la media più rapidamente rispetto all'*in-sample*.

Copiare grafici slide 40.

Il *bias* può essere ridotto aumentando il numero di regressori o usando la tecnica del *boosting*. La varianza, invece, si risolve riducendo il numero di regressori, aumentando le dimensioni dell'insieme dei dati di addestramento, o con le tecniche della *regolarizzazione* e del *bagging*.

L'*overfitting*, inteso come eccessivo adattamento del modello ai dati di addestramento a scapito della generalizzazione su nuovi dati, può essere causato dal rumore sui dati. Tale rumore viene interpretato come segnale, e il modello impara a seguirlo. Questo fenomeno è più evidente nei modelli complessi. A livello di decomposizione dell'errore, l'*overfitting* è caratterizzato da un basso *bias* e da un'elevata varianza. Sul grafico della *learning curve* la regione di overfitting inizia laddove l'errore *out-of-sample* riprende ad aumentare dopo aver raggiunto un minimo. Il fenomeno opposto all'*overfitting* si definisce *underfitting* ed è caratterizzato da elevato *bias* e ridotta varianza.

Regolarizzazione

In presenza di rumore è necessario aggiungere un termine nel calcolo del valore atteso dell'errore out-of-sample. Il termine in questione è definito *errore stocastico* σ^2, rappresenta la varianza del rumore, ed è irriducibile, perché

non dipende né dal modello né dalla distribuzione, ma è puramente stocastico.

$$e_{out} = bias^2 + varianza + \sigma^2$$

La tecnica della regolarizzazione è considerata il primo metodo da applicare in presenza di *overfitting*. Come spiegato nel paragrafo precedente, un modello complesso è in grado di seguire anche il rumore come se fosse un reale trend nei dati. Questo genera un'elevata varianza. Impiegare modello più semplice riduce la varianza a scapito di un aumento del *bias*, ma spesso la riduzione della prima è maggiore rispetto all'aumento del secondo, quindi l'errore complessivo si riduce. L'idea su cui si basa la regolarizzazione è penalizzare la complessità del modello nella funzione di costo dalla cui minimizzazione si ricavano i parametri del modello. A questo fine definiamo un errore aumentato, che sia funzione sia dell'errore, che di un termine $\Omega(\theta)$ chiamato *regolarizzatore* anch'esso dipendente dai parametri, a meno di un iperparametro moltiplicativo λ_l.

$$\text{costo}(\theta) = \sum_i (y_i - \hat{y}_i^{\theta})^2 + \lambda_l \cdot \Omega(\theta)$$

Il regolarizzatore deve essere una stima dell'*errore di generalizzazione*, ovvero la differenza tra errore *out-of-sample* ed errore *in-sample*. Un tipico esempio è la regolarizzazione L_2, nella quale il regolarizzatore è la somma dei quadrati dei coefficienti. Viene dunque favorita la forte riduzione dei coefficienti meno significativi, al fine di individuarli e rimuoverli. L'applicazione della regolarizzazione L_2 alla regressione lineare prende il nome di *ridge regression*. In tal caso spesso si decide di non penalizzare il parametro corrispondente all'intercetta. La regolarizzazione L_1, detta anche *lasso*, utilizza come regolarizzatore la sommatoria dei valori assoluti dei parametri. Se la penalizzazione *ridge* tende a ridurre il valore di tutti i coefficienti, la *lasso* azzera invece con forza i coefficienti meno significativi. La penalizzazione *elastic net*, invece, è una combinazione lineare dei regolarizzatori delle due tecniche. La regolarizzazione è un problema di ottimizzazione vincolata. Il regolarizzatore produce un vincolo che limita la regione in cui cercare il minimo della funzione di costo. La forma di tale vincolo è curvilinea per la penalizzazione *ridge* e spigolosa per la regressione *lasso*. Nella seconda, i minimi tendono a trovarsi sui vertici che, posizionandosi sugli assi cartesiani, azzerano alcuni parametri. Un eccesso nella sua applicazione finisce per azzerare tutti i parametri e cancellare il modello. In ogni caso, se calcolato correttamente, l'errore aumentato è in grado di approssimare l'errore out-of-sample molto meglio rispetto all'uso del semplice errore in-sample come stima.

$$\text{ridge: } \Omega(\theta) = \sum_i \theta_i^2$$

$$\text{lasso: } \Omega(\theta) = \sum_i |\theta_i|$$

$$\text{elastic net: } \Omega(\theta) = \beta \cdot \sum_i \theta_i^2 + (1-\beta) \cdot \sum_i |\theta_i|$$

Validazione, cross-validazione e regolarizzazione

Si può interpretare l'errore *out-of-sample* come la somma dell'errore *in-sample* con la penalità dovuta alla complessità. L'errore *out-of-sample* complessivo si stima per validazione, mentre la penalità si stima per regolarizzazione. Queste due tecniche non sono utilizzate solo per i modelli di machine learning ma anche per l'identificazione dei modelli dinamici Questi ultimi sono modelli di regressione ad apprendimento supervisionato per segnali a tempo continuo.

L'obiettivo dell'uso dell'insieme di validazione è la stima delle prestazioni *out-of-sample* del modello. Si rimuove un sottoinsieme di dati dal totale, si stima il modello sui dati rimanenti e infine se ne verificano le prestazioni sui dati rimossi. Solitamente nella divisione si tende ad assegnare il 70% dei dati al sottoinsieme di addestramento e il restante 30% alla validazione. Tale proporzione è da aggiustare in modo più fine sulla base delle curve di apprendimento. Utilizzando un grande sottoinsieme di validazione si ottiene una migliore stima dell'errore, ma essa sarà riferita ad un modello mal stimato, perché allargare l'insieme di validazione significa restringere quello di addestramento. Dopo la fase di validazione, se il modello è accettabile, si procede a riaddestrarlo sulla totalità dei dati. Ci si aspetta che il modello addestrato sui dati interi abbia mediamente un errore minore o uguale rispetto all'errore stimato dalla validazione. La divisione dei dati si effettua di solito con una tecnica chiamata *stratified sampling*. Essa mantiene il medesimo bilanciamento delle classi in entrambi i sottoinsiemi di addestramento e di validazione. Lo sbilanciamento di classi, seppure sempre da evitare, è meno problematico nei problemi di regressione che in quelli di classificazione. Esiste una teoria, chiamata teoria PAC, che fornisce dei limiti superiori teorici all'errore *out-of-sample* , che tuttavia spesso risultano talmente elevati da essere inutili.

La validazione è utilizzata per risolvere due diverse categorie di problemi. La prima, già affrontata, riguarda la valutazione delle prestazioni del modello. La seconda, invece, riguarda la scelta del modello migliore tra un insieme di modelli possibili. Tra queste decisioni ricadono la scelta del tipo di modello (lineare, non lineare, ...),

la scelta del numero di regressori, la scelta del parametro di regolarizzazione λ_{reg}, e varie altre scelte riguardanti l'architettura del modello e i suoi iperparametri. Per questo secondo tipo di validazione si addestrano e validano i diversi modelli sulla stessa suddivisione di dati tra sottoinsiemi di addestramento e di validazione, per poi scegliere il modello con l'errore di validazione minore. Per la scelta di iperparametri, si testano varie versioni dello stesso modello con valori dell'iperparametro scelti da una griglia tra un valore minimo e un valore massimo.

L'uso ripetuto dello stesso insieme di validazione può causare una nuova forma di *overfitting*. Scegliere iperparametri in base allo stesso sottoinsieme di validazione sul quale si prova il modello, infatti, ottimizza le prestazioni di quest'ultimo su quegli specifici dati. Per questo motivo è utile fare una ulteriore divisione nei dati. Si separa un'ulteriore porzione di dati, chiamata *sottoinsieme di test*, sul quale valutare il modello finale. Questo insieme non è riutilizzabile, pena incorrere nello stesso errore che si è cercato di evitare separando la fase di validazione dalla fase di prova. Il sottoinsieme di addestramento è totalmente contaminato, quello di validazione lo è parzialmente; l'unico completamente incorrelato al modello è il sottoinsieme di prova. Utilizzare i dati di addestramento o di validazione per la prova causa una stima al ribasso dell'errore *out-of-sample*. Le più comuni proporzioni nei quali dividere i dati sono: 60% addestramento, 20% validazione, 20% prova.

Non sempre il numero di dati è sufficiente per una divisione in tre sottoinsiemi. Questa insufficienza si riconosce osservando le curve di apprendimento: se l'errore *in-sample* risulta eccessivamente elevato in corrispondenza del 70-80% dei dati disponibili, la divisione non è accettabile. Utilizzando questi dati per l'addestramento, ne risulterebbe un modello insufficiente. Aumentare la proporzione ridurrebbe l'efficacia del sottoinsieme di validazione. Per risolvere questa problematica si introduce una tecnica chiamata *validazione incrociata*. Il tipo più semplice è chiamato *leave-one-out*. In questa variante si addestra il modello su $N-1$ dati e lo si valida sul singolo dato restante. Questo crea N possibili coppie insieme di addestramento / dato di validazione. L'errore di validazione complessivo, detto *errore di cross-validazione*, è la media degli N errori di validazione calcolati per le coppie addestramento / dato di validazione. Una metrica utile è la deviazione standard campionaria dei singoli errori di validazione. Valori elevati di tale varianza indicano grande sensibilità ai dati di addestramento, che a sua volta è sintomo di *overfitting*. Una tecnica alternativa è la validazione incrociata a k pieghe (*k-fold*), che divide i dati in k porzioni, di cui $k-1$ da usare per l'addestramento, e una per la validazione. Si calcola poi l'errore di cross-validazione come media dei k errori di validazione ottenuti da tutte le possibili combinazioni di dati di addestramento e di validazione. Questa variante, rispetto alla *leave-one-out*, implica una minore complessità computazionale e produce una stima dell'errore *out-of-step* con varianza inferiore. Indicativamente un buon compromesso è scegliere $k = 10$.

Come già descritto, una sproporzione di regressori rispetto al numero di dati di addestramento tende a causare *overfitting*. La complessità ottimale del modello non è tuttavia una semplice funzione del numero di dati, ma anche di altri fattori legati alla loro qualità. Per questo può risultare utile effettuare una selezione dei regressori. Si scelgono dunque i regressori con la massima correlazione univariata all'uscita, per validazione incrociata; si addestra il modello su tali regressori; si utilizza nuovamente la validazione incrociata per stimare l'errore *out-of-sample* ed eventuali iperparametri. Ogni scelta legata alla configurazione deve essere sottoposta a validazione incrociata.

Le *formule di complessità ottima* sono strumenti matematici per casi ancora più estremi rispetto a quelli che richiedano validazione incrociata. Esse permettono di stimare la complessità ottimale del modello assumendo di utilizzare per il solo addestramento la totalità dei dati, dunque senza lasciare sottoinsiemi ai fini della validazione o della prova. Sono state sviluppate in tempi in cui la raccolta, lo stoccaggio e la suddivisione dei dati erano lente e laboriose, impedendo di averne abbastanza per poter effettuare la validazione secondo gli approcci precedentemente descritto. Al contrario di metodi come la regolarizzazione, le formule di complessità ottima regolano la complessità in modo discreto. Una delle formule di complessità ottima è nota come *Akaike Information Criterion*, oppure come *Final Prediction Error* quando impiegata in ambito di sistemi dinamici:

$$AIC(d) = 2 \cdot \frac{d}{N} + \ln[J(\hat{\theta}_N; d)].$$

Questa formula tende a sovrastimare l'ordine con probabilità non nulla quando il campione contiene più di otto dati e il modello appartiene alla stessa classe della funzione origine dei dati (ad esempio quando si usa un modello lineare su dati distribuiti linearmente). Esiste dunque un criterio alternativo, chiamato *Minimum Description Length*, che è asintoticamente corretto:

$$MDL(d) = \ln[N] \cdot \frac{d}{N} + \ln[J(\hat{\theta}_N; d)]$$

L'assunzione che rende impreciso il criterio di Akaike è raramente verificata, quindi nella maggior parte dei casi d'uso si preferisce quest'ultimo rispetto all'MDL per la maggiore velocità di calcolo. In entrambe le formule il primo termine è crescente all'aumentare della complessità del modello, e il secondo termine è decrescente. È dunque possibile trovare un d (discreto) che sia argomento minimo, e che rappresenti l'ordine del modello a complessità ottima.

Per ricapitolare: dove possibile, è ideale utilizzare la suddivisione in sottoinsiemi di addestramento, validazione e prova; quando non è possibile, si impiega la validazione incrociata; in caso di estrema penuria di dati si ricorre alle

formule di complessità ottima. Si anticipa che, nei modelli dinamici, queste tecniche rimangono applicabili ma è fondamentale ricordare che i dati sono una serie storica, e dunque il loro ordine non va rotto nel suddividere l'insieme.

Stima bayesiana

Si introduca ora la *stima bayesiana*. Essa si basa sull'assunzione che i parametri, come i dati, siano variabili casuali. A questo fine, recuperiamo alcuni concetti di statistica multivariata. Supponiamo di avere due variabili casuali discrete e binarie a e b. Definiamo *distribuzione di probabilità congiunta* la tabella delle probabilità $P(a, b)$ della realizzazione di coppie di a e b. La sommatoria di tutti i valori della tabella delle probabilità congiunte deve essere 1.

$$\sum_{a=0}^{1}\sum_{b=0}^{1} p(a,b) = 1$$

	$b=0$	$b=1$
$a=0$	0.4	0.3
$a=1$	0.2	0.1

Si definisce *distribuzione marginale* la distribuzione congiunta di un sottoinsieme di variabili casuali. Nel caso discreto il calcolo si riconduce alla somma delle probabilità di interesse lungo l'asse delle variabili non di interesse, all'interno della matrice delle probabilità congiunte. Nel caso continuo la somma è sostituita da un'integrazione. Con una coppia di variabili casuali binarie, l'operazione equivale alla sommatoria di un'intera riga o colonna della tabella delle congiunte.

	$b=0$	$b=1$	**congiunte**
$a=0$	0.4	0.3	$P(a=0)=0.7$
$a=1$	0.2	0.1	$P(a=1)=0.3$
congiunte	$P(b=0)=0.6$	$P(b=1)=0.4$	

Definiamo *distribuzione condizionata* la distribuzione delle probabilità al restringersi della popolazione considerata. Al realizzarsi di uno specifico valore di una variabile casuale, ridistribuiamo la probabilità delle possibili realizzazioni delle variabili casuali restanti in modo che sommino nuovamente a 1. Dati N_A e N_B, $P(A,B) = \frac{N_{AB}}{N}$ e dunque

$$P(A|B) = \frac{N_A B/N}{N_B/N} = \frac{P(A,B)}{P(B)}.$$

Si introduca ora il *teorema di Bayes*. Dato che $P(A,B) = P(B,A)$ e che $P(B,A) = P(B|A)P(A)$, allora $P(A|B)P(B) = P(B|A)P(A)$, e quindi :

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}.$$

Esso permette di ridistribuire la probabilità di A a posteriori dell'informazione portata da B. Notare che "a posteriori" non intende sequenzialità temporale, ma solo di conoscenza. $P(B)$ agisce come una sorta di fattore di normalizzazione. La probabilità condizionata degenera al prodotto tra le probabilità delle due classi soltanto quando i due eventi sono tra loro indipendenti (ovvero quando $P(A|B) = P(A)$ e $P(B|A) = P(B)$.

Il principio della probabilità a posteriori introdotto dal teorema di Bayes può essere applicato alla stima dei parametri dei modelli. Invece che vedere il vettore dei parametri θ come una variabile vettoriale deterministica, possiamo interpretarlo come una variabile casuale dotata di una certa distribuzione. Tale distribuzione è scelta in base alle conoscenze a priori del creatore del modello. Informazioni relative alla natura del fenomeno osservato possono, ad esempio, portare ad aspettarsi che il parametro abbia un certo valore atteso, e che le sue possibili realizzazioni si ammassino attorno a tale valore atteso secondo una certa distribuzione. Questa distribuzione è a priori, perché costruita prima della raccolta dei dati. I dati effettivi rappresentano invece l'informazione a posteriori in base alla quale aggiustare la distribuzione. L'informazione, nella stima bayesiana, è dunque portata da due elementi: la distribuzione a priori, e l'informazione dei dati. Quest'ultima è interpretabile come la verosimiglianza.

$$\underset{\text{a posteriori}}{f_{\theta|Y}(\theta|Y)} = \frac{\overset{\text{verosimiglianza}}{f_{Y|\theta}} \cdot \overset{\text{a priori}}{f_\theta(\theta)}}{\underset{\text{verosimiglianza marginale}}{f_Y(Y)}}$$

Possiamo insomma aggiustare la verosimiglianza a posteriori in base a quello che sappiamo a priori sulla distribuzione dei dati, prima di raccoglierli effettivamente. La stima bayesiana è un compromesso tra la stima a priori e la stima di verosimiglianza proveniente dai dati. Si tratta di una sorta di regolarizzazione. La stima che massimizza la verosimiglianza a posteriori è detta *MAP (Maximum A Posteriori)*, definita come $\hat{\theta} = \arg\max_\theta f_{\theta|Y}(\theta|Y)$. Tecniche alternative per ricavare valori puntali dalla distribuzione a posteriori includono anche il *valore atteso a posteriori* ($\hat{\theta} = \mathbb{E}_\theta[f_{\theta|Y}(\theta|Y)] \equiv \mathbb{E}[\theta|Y]$) e la mediana.

Il calcolo della distribuzione a posteriori è solitamente complesso per via analitica. Per questo, spesso si utilizza una tecnica di integrazione numerica nota come *Markov Chain Monte Carlo (MCMC)*. Il calcolo è invece immediato quando sia la distribuzione a priori che la distribuzione dei dati sono gaussiane. In tal caso, anche la distribuzione a posteriori è gaussiana.

Chiamiamo $\hat{\theta} = T(\mathcal{D})$ ogni stimatore che sia funzione dei dati. Consideriamo il MSE nel caso scalare $MSE = \mathbb{E}[(\hat{\theta} - \theta)^2] = \mathbb{E}[(T(\mathcal{D}) - \theta)^2]$. Definiamo *stimatore ottimo di Bayes* lo stimatore $T^{opt}(\ )$ tale da avere il minimo MSE tra tutti i possibili stimatori. Esso ha il valore atteso condizionato della variabile:

$$T^{opt}(Y) = \mathbb{E}[\theta|\mathcal{D} = Y]$$

Supponiamo di avere un parametro scalare ignoto $\theta \sim \mathcal{N}(0, \lambda^2_{\theta\theta})$ che sia realizzazione di una variabile casuale gaussiana e di avere un singolo dato $y \sim \mathcal{N}(0, \lambda^2_{yy})$ anch'esso realizzazione di una variabile casuale gaussiana. Allora la funzione di densità di probabilità congiunta, una bivariata, è anch'essa gaussiana, così come è gaussiana la distribuzione a posteriori. In particolare, essa avrà i seguenti valori di media e varianza:

- $\mu_{\theta|y} = \frac{\lambda_{\theta y}}{\lambda^2_{yy}} \cdot y$
- $\lambda^2_{\theta|y} = \lambda^2_{\theta\theta} - \frac{\lambda^2_{\theta y}}{\lambda^2_{yy}}$

Notiamo che il secondo termine nella varianza è sempre negativo, quindi l'incertezza della distribuzione a posteriori (la varianza) è minore rispetto a quella a priori. Interpretiamo questo fenomeno come l'aumento dell'informazione grazie all'arrivo dei dati. A denominatore di tale secondo termine abbiamo la varianza dei dati che, crescendo, abbassa il termine. Dati ad alta varianza, infatti, portano poca informazione.

Non sempre il dato e il parametro sono congiuntamente gaussiani. Rimuovendo questa ipotesi, proviamo a costruire uno stimatore solamente sapendo che le due variabili sono casuali, i loro valori attesi sono entrambi nulli, e che le loro varianze sono definite.

$$\hat{\theta}^{lin} = \alpha \cdot y + \beta, \quad \alpha, \beta \in \mathbb{R}$$

Possiamo ottenere uno stimatore ottimo minimizzando il MSE rispetto ai due parametri α e β. Lo stimatore lineare ottimo è pari a

$$\hat{\theta}^{lin}_{opt} = \hat{\alpha} \cdot y + \hat{\beta} = \frac{\lambda_{\theta y}}{\lambda^2_{yy}} \cdot y.$$

Possiamo notare la somiglianza alla formula del valore atteso per il caso gaussiano. Lo stimatore lineare ottimo è il migliore stimatore che si possa ottenere non facendo ipotesi. Aggiungendo ipotesi aggiuntive si può ottenere una stima più precisa, a meno che non si tratti di un'ipotesi gaussiana. In tal caso, il miglioramento è nullo perché le formule coincidono. Esistono versioni generalizzate dello stimatore lineare ottimo per casi con valore atteso non nullo e per parametri vettoriali. Le formule dello stimatore lineare ottimo ammettono una formula ricorsiva. Essa permette di non dover ricalcolare l'intero stimatore all'arrivo di un nuovo dato, basandosi invece sulla stima precedente. Questa proprietà è particolarmente utile per i sistemi dinamici. Un'applicazione in tale ambito è il *filtro di Kalman*, un algoritmo che stima lo stato di un sistema dinamico in funzione di un ingresso in serie storica. Lo stato attuale è la stima a priori, mentre gli ingressi del sistema sono i nuovi dati osservati. La predizione dello stato all'istante successivo è la stima a posteriori.

Sistemi dinamici

D'ora in avanti il termine *sistema ingresso-uscita* farà riferimento a un sistema SISO LTI. Questi modelli lavorano con ingressi in serie storica. Solitamente questi modelli sono costruiti come unione di due modelli secondari, di cui uno modella la componente nota e l'altro il disturbo. Il termine di disturbo modella disturbi quali: errori di misura; disturbi di processo ed effetti di segnali esogeni non misurabili nell'ingresso; effetti della linearizzazione del sistema. Entrambi i termini, componente nota e disturbo, sono modellizzati come funzioni di trasferimento di sistemi SISO LTI a tempo discreto. Distinguiamo tra sistemi ingresso uscita propriamente detti, e *serie temporali*, ovvero modelli il cui ingresso non è misurabile. A livello implementativo le serie temporali sono rappresentate come serie storiche il cui ingresso è utilizzato come ausilio matematico. Per entrambi i tipi di modello ci si pongono due obiettivi. Innanzitutto si vuole prevedere l'uscita a istanti futuri $t + k$ in base alle informazioni presenti a t: $\hat{y}(t + k|t)$. Si desidera inoltre identificare (ovvero stimare, nel gergo dei modelli dinamici) i modelli descritti.

Queste due fasi devono essere precedute da un'adeguata fase di scelta della classe dei modelli da impiegare. Per convenzione, lavorando d'ora in avanti con modelli a tempo discreto con periodo di campionamento T_s, si intenderà implicitamente $y(t) = y(t \cdot T_s)$.

Si assuma che l'uscita del modello sia sempre affetta da un disturbo e che anch'essa sia funzione della variabile temporale. In un modello statico il disturbo è rappresentabile come una variabile casuale. In un modello dinamico il disturbo è invece da intendersi come una successione di variabili casuali. Tale successione prende il nome di *processo stocastico*. Un processo stocastico è una successione infinita di variabili casuali definite a partire dallo stesso esperimento casuale, e ordinate secondo un indice temporale. Fissando l'esito si ottiene la realizzazione; fissando il tempo si ottiene una variabile casuale; fissando entrambi si ottiene un numero. I processi stocastici sono utili per analizzare fenomeni che non si possono (o non si vogliono) descrivere in modo analitico e deterministico. Un processo stocastico è completamente caratterizzato dal punto di vista probabilistico se per ogni sottoinsieme di variabili casuali è definita la congiunta. Un esempio di processo stocastico è il *random walk.* Ad ogni passo il valore assunto dal processo è pari alla somma valore del passo precedente con la realizzazione di una variabile casuale.

Un processo stocastico è completamente caratterizzato a livello probabilistico se è definita la distribuzione congiunta per ogni sottoinsieme di sue variabili casuali. Per i processi discreti, questo implica di dover conoscere la distribuzione di probabilità congiunta di un numero infinito di variabili casuali. A meno che non si tratti di variabili gaussiane o pochi altri casi facilmente studiabili, questa assunzione è inottenibile. Spesso, dunque, non ci si può spingere oltre alla stima del valore atteso e della funzione di covarianza o *autocorrelazione*, detta *caratterizzazione del secondo ordine*. Il valore atteso, momento del primo ordine, è funzione del tempo e rappresenta la media tra le possibili realizzazioni del processo stocastico ad un determinato istante. Non va confuso con la *media storica*, ovvero la media dei valori effettivamente realizzati dal processo durante la sua evoluzione temporale. L'autocorrelazione, invece, si può intendere come la correlazione di una variabile con sé stessa ad istanti diversi. Un concetto simile è l'autocovarianza, ovvero la covarianza di una variabile con sé stessa ad istanti diversi. Imponendo due tempi uguali, otteniamo la varianza della variabile. Una generalizzazione è l'*autovarianza normalizzata*, per il cui calcolo l'autovarianza viene divisa per la radice quadrata del prodotto delle varianze ai due tempi. Essa viene calcolata da Matlab al posto della covarianza.

Si parli ora di *processi stocastici congiunti*. Dati due processi stocastici v e x, si possono definire le loro funzioni di *cross-correlazione* e *cross-covarianza*:

$$R_{vx}(t_1, t_2) \equiv \mathbb{E}_s[v(t_1, s) \cdot x(t_2, s)] = R_{xv}(t_2, t_1)$$

$$\gamma_{vx}(t_1, t_2) \equiv \mathbb{E}[(v(t_1, s) - m_v(t)) \cdot (x(t_2, s) - m_x(t))] = \gamma_{xv}(t_2, t_1)$$

Due processi stocastici si dicono incorrelati se la loro cross-covarianza è sempre nulla. Anche per la cross-covarianza esiste una versione normalizzata.