IMAD

2024-2025

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.

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

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

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