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.