Regressione Lineare

L'obiettivo della regressione è quello di predirre il valore di uno o più target continui t, dato il valore di un vettore x D-dimensionale di variabili di input. La regressione polinomiale è un esempio specifico di **regressione lineare**: una classe di modelli di regressione che condividono la caratteristica di essere lineari nel rispetto dei parametri. Un sottogruppo di modelli lineari, ancora più semplici, prevede di essere lineare anche rispetto alle variablili di input, ma vedremo che può essere molto limitante. Conviene invece prendere combinazioni lineari di un set fissato di funzioni non lineari. Queste funzioni sono dette funzioni base (**basis functions**).

Modelli di funzioni base lineari

Il modello linere più semplice possibile per la regressione è quello formato da una combinazione lineare di variabili di input:

$$y(x, w) = {(w_0+w_1x_1+ \dots + w_Dx_D)}$$

Questo modello lineare è tale perché è lineare rispetto ai parametri w. Inoltre risulta lineare anche negli input $x = (x_1, \dots, x_D)^T$, ma questo impone delle limitazioni notevoli. È per questo motivo che estendiamo la classe di modelli di regressione lineare, includendo anche tutti quei modelli che sono combinazione lineare dei parametri $w = (w_1,\dots,w_D)$ e combinazione lineare di un set fissato di funzioni non lineare di x. I modelli prendono la seguente forma:

$$y(X,W)=w_0+\sum_{j=1}^{M-1} w_j \phi_j(X) $$

Dove $\phi_j(X)$ sono le funzioni base. Questo modello è lineare rispetto ai parametri w, ma non è lineare rispetto ai parametri ϕ. Questo modello è chiamato **modello di funzioni base lineari**. Il massimo valor di j è M-1, quindi il numero totale di parametri del modello sarà M.

Il parametro w_0 è chiamato **bias** e permette qualsiasi offset fisso nei dati.

Analiziamo adesso il caso di modello di regressione lineare polinomiale: questo è un caso particolare di regressione lineare, in cui è presenre una sola variabile di input x, e in cui le funzioni di base prendono la forma delle potensze di x $\phi_j(x) = x^j$. Il più grande problema delle funzioni base polinomiali è sono funzioni globale della variabile di input, ovvero che un cambiamento in una regione di x influisce in tutte le altre regioni. QUesto può essere risolto dividendo lo spazio degli inpunt, in diverse regioni e per ognuna usare un polinomio diverso, ottentendo così una spline, per esempio:

$$\phi_j(x) = \exp {-(x-\mu_j)^2 \over 2s^2}$$

Queto tipo di funzioni base sono chiamate **Gaussiane**.

Maximum likelihood e Least Squares

Assumiamo che la variabile di target t sia data da una funzione deterministica $y(X,W)$ con l'aggiunta di un errore Gaussiano ϵ:

$$t = y(X,W) + \epsilon$$

Dove ϵ è una variabile casuale con distribuzione Gaussiana con media zero e varianza β^{-1}. Possiamo quindi scrivere:

$$p(t|X,W,\beta) = N(t|y(X,W),\beta^{-1})$$

Se consideriamo una squared loss function, allora la predizinoe ottimanle per un nuovo valore di x, sarà data dalla media pesata della variabile di target:

$$E[t|X] = \int t p(t|X) dt = y(X,W)$$

Per via del rumore Gaussiano, la distribuzione condizidizionata di t data x è unimodale, ovvero presenta un solo picco di massimo, che può risultare limitante per alcune applicazioni.

Addesso consideriamo un set di input $X = {x_1, \dots, x_N}$ e un set di target $T = {t_1, \dots, t_N}$. Raggruppiam le variabili target in un vettore a colonna e lo chiamiamo t. Assumendo che questi dati sono indipendenti dalla distribuzione, otteniamo la seguente formula per la funzione di likelihood:

$$p(t|X,W,\beta) = \prod_{n=1}^{N} N(t_n|y(x_n,W),\beta^{-1})$$

Questa è una funzione dei parametri W e β. Applicando il logaritmo, ricavaviamo:

$$ln p(t|X,W,\beta) = \sum_{n=1}^N ln N(t_n|y(x_n,W),\beta^{-1})$$

$$= -{N \over 2} ln \beta - {N \over 2} ln(2\pi) - \beta E_D(W)$$

Dove $E_D(W)$ è la somma degli errori al quadrato:

$$E_D(W) = {1 \over 2} \sum_{n=1}^N { t_n - W^T \phi(X_n) }^2$$

Dato che abbiamo scritto la funzione di likelihood, possiamo massimizzarla per determinare w e β. Massimizziamola nel rispetto di w. Grazie alla fomra logaritmica della funzione di likelihood è evidente che massimizzare enl rispetto di W, significa massimizzare la somma dei quadrati degli errori $E_D(W)$:

$$ \nabla ln {p(t|W, \beta)} = \sum_{n=1}^N { t_n - W^T \phi(X_n) } \phi(X_n) $$

Impostanto il gradiente a 0 risolvendo nel rispetto di W, otteniamo la seguente equazione:

$$W_{ML} = (\Phi^T \Phi)^{-1} \Phi^T t$$

Detta **equazione normale**. Dove Φ è la matrice di dimensioni $N \times M$ e i cui elementi sono dati da $\Phi_{nj} = \phi_j(x_n)$.

$$\Phi = \begin{bmatrix} \phi_0(x_1) & \phi_1(x_1) & \dots & \phi_{M-1}(x_1) \ \phi_0(x_2) & \phi_1(x_2) & \dots & \phi_{M-1}(x_2) \ \vdots & \vdots & \ddots & \vdots \ \phi_0(x_N) & \phi_1(x_N) & \dots & \phi_{M-1}(x_N) \end{bmatrix}$$

Bias

Ripendiamo la formula ricavata per $E_D(W)$ e esplicitiamo il bias ($w_0):

$$E_D(W) = {1 \over 2} \sum_{n=1}^N { t_n - w_0 - \sum_{j=1}^{M-1}w_j\phi_j(X_n) }^2$$

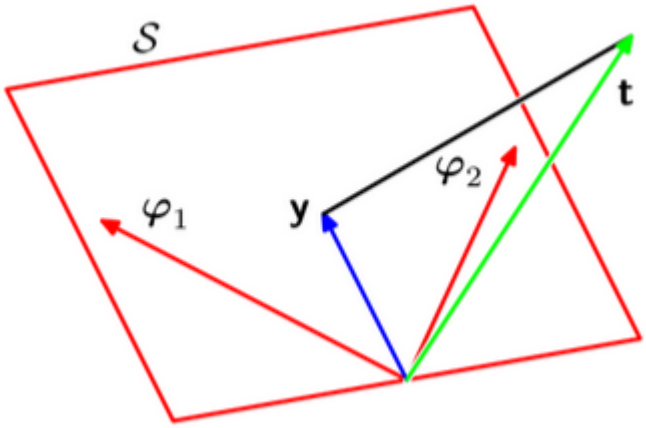
Derivando rispetto a w_0 e uguagliando a zero, otteniamo il seguente risultato:

$$w_0 = {1\over N}\sum_{n=1}^N t_n - \sum_{j=1}^{M-1}w_j {1\over N}\sum_{n=1}^{N}\phi_j (X_n) = \bar{t} - \sum_{j=1}^{M-1} w_j \bar{\phi_j}$$

Quindi il bias w_0 **compensa la differenza tra la medie dei valori target e la somma pesata delle medie delle funzioni di base**.

Intepretazione geometrica dei minimi quadrati

Consideriamo uno spazio N-dimensinale i cui assi sono dati da t_n, così che il vettore target t è un vettore in tale spazio. Ongi funzione base $\phi_j(x)$ valorizzata negli N punti, può anch'essa essere visata come un vbettore nello stesso spazio, chiamato α_j.



α_j corrisponde alla $j-$esima colonna della mapatrice Φ. Se M < N, allora il vettore i vettori $\phi_j(X_n)$, formeranno un sottospazio lineare S di dimensione M all'interno dello spazio N-dimensionale. Dato che y è una combinazione lineare arbitraria dei vettori α_j, il vettore y giacerà nello spazio S. Il vettore target t giacerà in uno spazio N-dimensionale

Ripendiamo adesso la formula della somma dei quadrati degli errori:

$$E_D(W) = {1 \over 2} \sum_{n=1}^N { t_n - W^T \phi(X_n) }^2$$

Questa formula diventa uguale a:

$$E_D(W) = {1 \over 2} || t - \Phi W ||^2$$

ovvero la distanza Euclidea tra il vettore y e t (a meno di $1 \over 2$). Dato che y giace su S, la soluzione di minimo quadrato è data dalla proiezione ortogonale di t su S. Questa cosa è verificabile, notando che la soluzone per y è data da ΦW_ML, dove W_ML è il vettore dei pesi che ottimizza i minimi quadrati.

Potrebbero sorgere delle difficoltà numeriche: se la matriche $\Phi^T \Phi$ è quasi singolare (determinante vicno a zero). Questo accade quando due vettori o più, tra i vetotir base α_j sono quasi collineari. Questo può portare a parametri risultanti molto molto grandi.

Sequential Learning

Il problema della tecnica del maximum likelihood è che richiede di processare tutto il seti di addestramento in un una sola volta. Con UN set molto grande questo inizia a esse oneroso. Se i il set è sufficientemente grande può convenire addestrare il modello sequenzialmente, ovvero aggiungendo un dato alla volta (one-

line algotihm), in modo da aggiornare i parametri del modello a ogni presentazione dal dataset. Questo approccio è l'ideale anche per applicaizoni in real time, in cui i dati arrivano costantemente. Per applicare questo approccio, possiamo usare la tecnica conosciuta come seque4ntial gradient descent. Se la funzione di errore, consiste nella somma degli errori dei singoli punti, allora dopo la presentazione del pattern n l'algoritmo del gradiente sequenziale, aggiornerà il vettore dei parametri W come seque:

$$W^{(\tau+1)} = W^{(\tau)} + \eta \nabla E_n(W)$$

dove τ indica in numero dell'interazione, η è il learning rate e $\nabla E_n(W)$ è il gradiente dell'errore. della singola presentazione n. Per il caso della somma degli errori al quadrato, abbiamo:

$$w^{(\tau+1)} = w^{(\tau)} + \eta (t_n - w^{(\tau)^T} \phi(X_n)) \phi(X_n)$$

Questa formual è nota col nome di leas-mean square (LMS) algorithm.

Regualrized Least Squares

Andiamo a aggiungere un termine di regolarizzazione alla funzione di errore per evitare overfitting. La funzione di errore da minimizzare diventa:

$$E_D(W) + \lambda E_W(W)$$

dove λ è il coefficiente di regolarizzazione che controlla l'importanza dell'errore dipendente dai dati $E_D(W)$ e il temine di regolarizzazione $E_W(W)$.

Il modo più comune di regolarizzare è quello di usare un regolarizzatore dato dalla somma dei quadrati dei pesi:

$$E_W(W) = {1 \over 2} W^T W$$

Consideriamo la funzione di errore come somma dei quadrati degli errori $E_D(W) = {1 \over 2} \sum_{n=1}^N { t_n - W^T \phi(X_n) }^2$. Otteniamo:

$${1 \over 2} \sum_{n=1}^N { t_n - W^T \phi(X_n) }^2 + {1 \over 2} \lambda W^T W$$

Questa forma di regolarizzatore è detta **weight decay**, perché negli algoritmi di apprendimento sequenziale fa tendere i pesi dei valori a zero, a meno che non siano supportati dai dati.

Il vantaggio di questo tipo di regolarizzatore è che la funzione di errore rimane quadratica rispetto ai pesi, quindi la minimizzaizone avviene nello stesso modo. Calcoliamo quindi il gradiente della funzione di errore regolarizzata:

$$\nabla E(W) = \sum_{n=1}^N { t_n - W^T \phi(X_n) } \phi(X_n) + \lambda W$$

Uguagliando a zero il gradiente, otteniamo la seguente equazione:

$$W = (\lambda I + \Phi^T \Phi)^{-1} \Phi^T t$$

Ricordiamo a cosa è uguale Φ:

$$\Phi = \begin{bmatrix} \phi_0(x_1) & \phi_1(x_1) & \dots & \phi_{M-1}(x_1) \ \phi_0(x_2) & \phi_1(x_2) & \dots & \phi_{M-1}(x_2) \ \vdots & \vdots & \ddots & \vdots \ \phi_0(x_N) & \phi_1(x_N) & \dots & \phi_{M-1}(x_N) \end{bmatrix}$$

Possiamo generalizzare la formula del regolarizzatore ponendo l'esponente uguale a q:

$${1 \over 2} \sum_{n=1}^N { t_n - W^T \phi(X_n) }^2 + {\lambda \over 2} \sum_{j=1}^M |w_j|^q$$

q=2 corrisponde al regolarizzatore visto fino ad ora.

Output multipli

Fino ad ora abbiamo considerato il caso di un solo target. Possiamo estendere il modello in modo da predirre contemporaneaeente K>1 target diversi, raggruppati in un vettore target t. È possibile ottenere questo risultato introducendo un set di funzioni base differente per ogni target, ottentendo problemi di regressione tra loro indipendenti. Un approccio più comune è quello di usare un singolo set di funzioni base per modellare tutti i termini del vettore target. Definiamo:

$$y(X,W) = W^T \phi(X)$$

dove y è K-dimensionale e W è M x K (matrice di parametri), mentre $\phi(X)$ è M-dimensionale.

Supponiamo di prendere una distribuzione condizionata Gaussiana per i target:

$$p(t|X,W,\beta) = N(t|W^T \phi(X), \beta^{-1} I)$$

Se abbiamo un set di osservazioni $t_1, \dots, t_N$, possiamo creare una matrice T N x K in modo che la riga n-esima contenga il target per l'osservazione n-esima t_n^T. In modo analogo creaiamo a matrice X. Quindi la forma logaritmica della funzione di likelihood diventa:

$$ln p(T|X,W,\beta) = \sum_{n=1}^N ln N(t_n|W^T \phi(X_n), \beta^{-1} I)$$

che è uguale a:

$${NK \over 2} ln ({\beta \over 2\pi}) - {\beta \over 2} \sum_{n=1}^N ||t_n - W^T \phi(X_n)||^2$$

Decomposizione bias-varianza