Ai miei nonni

"Nobody really believes that multivariate data is multivariate normal"

Statistical Modeling: The Two Cultures; Leo Breiman

Indice 3

Prima Parte

1.Introduzione

La teoria dell'apprendimento statistico si articola su tre problemi fondamentali: pattern recognition, regression estimation e density estimation. Ognuno di questi può essere visto come un caso particolare di un più generale problema che è quello dell'apprendimento attraverso esempi.

Cosa significa apprendere? Quando è auspicabile applicare tecniche basate sulla teoria statistica dell'apprendimento piuttosto che metodologie "classiche" fornite dalla letteratura?

Per meglio capire cosa si intende con apprendimento nell'ambito che andremo a trattare è utile partire con un esempio: si consideri uno studente che nel preparare un esame ha solo accesso al testo delle esercitazioni (con soluzione) ma che non può effettivamente studiare la teoria.

Lo studente quindi inizierà a fare i diversi esercizi e a confrontare le sue soluzioni con quelle riportate dal libro. Non potendo accedere alla teoria lo studente inizia ad intuire una sorta di regola che sta dietro al problema e quindi a rieseguire gli stessi esercizi ottenendo una soluzione sempre più vicina a quella corretta.

Quand'è che lo studente ha davvero imparato?

La risposta più semplice sarebbe dire che lo studente ha imparato quando smette di commettere errori sugli esercizi. Il problema di tale risposta è che se allo studente viene presentato un nuovo esercizio l'aver fatto correttamente tutti quelli del libro di addestramento non ci garantisce la sua capacità di rispondere correttamente al nuovo problema per il quale non è disponibile la soluzione.

Lo studente non ha realmente imparato la regola che genera gli esempi e le soluzioni ma ha semplicemente memorizzato una serie di input/output dal libro di addestramento che riprende solo una parte degli infiniti problemi che si possono presentare (una sorta di campione potremmo dire).

Quello che a noi interessa invece è la possibilità di non fare errori con i nuovi esempi che ci vengono proposti per i quali non abbiamo una soluzione pronta. Questo può essere fatto solo se si tenta di intuire la regola che opera dietro le quinte della realtà che osserviamo.

Va notato che lo studente non avrà mai accesso al testo della teoria quindi la regola "vera" che davvero genera il fenomeno non sarà mai nota. Quello che però conta è riuscire ad intuire una regola che non sia troppo diversa da quella presente nel libro della teoria.

Per quanto semplice questo esempio ben rappresenta tutte le componenti che sono presenti nella teoria dell'apprendimento statistico in ambito supervisionato (ossia nel caso in cui l'informazione campionaria disponga anche degli output della funzione ignota).

Più formalmente possiamo individuarli in tre componenti:

1. un generatore casuale di vettori x assunti i.i.d. rispetto ad una distribuzione ignota X (le tracce dei vari esempi sul libro degli esercizi)
2. un supervisore che per ogni vettore di input x restuisce un output y in base ad una probabilità condizionata $P(y|x)$ anch'essa ignota (le soluzioni ai vari esempi)
3. una learning machine in grado di implementare un set di funzioni $f(x, \alpha), \alpha \in \Lambda$ (la regola che lo studente intuisce dagli esempi)

Quindi il problema dell'apprendimento consiste nel scegliere un elemento dell'insieme di funzioni $f(x, \alpha), \alpha \in \Lambda$ che sia in grado di prevedere al meglio la risposta del supervisore.

La scelta della funzione che meglio approssima il supervisore deve essere fatta sulla base del c.d. training set ossia l'insieme delle l osservazioni campionarie $(x_1, y_1), \dots, (x_l, y_l)$ assunte i.i.d. secondo la probabilità ignota

$$P(x, y) = P(x)P(y|x)$$

ed utilizzando un algoritmo che condizionatamente all'informazione campionaria seleziona dal set di ipotesi predefinito la funzione che più si avvicina al comportamento del supervisore.

L'ultimo punto è di fondamentale importanza, infatti stiamo dicendo che la soluzione di ottimo che andremo a definire dipende da:

1. il campione casuale che abbiamo osservato (al variare del campione varia la soluzione)
2. il set di ipotesi scelto ossia la famiglia di funzioni che decidiamo di utilizzare per approssimare il supervisore (sostanzialmente scelto dal ricercatore)
3. l'algoritmo di ricerca nello spazio di funzioni del set di ipotesi (che quindi risolve un problema di ottimizzazione tipicamente non lineare con metodi di ricerca stocastica)

ciò significa che la soluzione del metodo sarà l'ottimo condizionatamente al campione utilizzato, alla famiglia di funzioni scelta e al fatto che l'algoritmo abbia individuato davvero la soluzione di ottimo globale (se esiste).

A questo punto sorge una domanda ovvia: perché dobbiamo definire una famiglia di funzioni tra cui cercare la soluzione? Perché non considerare la famiglia di tutte le possibili

funzioni? Data la complessità della domanda rimando la risposta al capitolo in cui si tratteranno le misure di capacità di un set di funzioni.

Qual' è dunque il criterio fondante per capire quale funzione scegliere per rappresentare al meglio il fenomeno? Evidentemente il modo più semplice è definire una misura di errore tra i risultati della funzione definita dall'algoritmo e l'output osservato y.

Definendo quindi con $f(x, \alpha)$ la risposta per un dato input x della funzione fornita dalla learning machine definiamo la perdita attesa come:

$$R(\alpha) = \int L\big(y, f(x, \alpha)\big) dP(x, y)$$

ove $L\big(y, f(x, \alpha)\big)$ è una funzione di perdita che misura la distanza tra l'output osservato e la previsione del modello per quel particolare x.

L'obiettivo dunque è individuare la funzione $f(x, \alpha_0)$ che rende minimo $R(\alpha)$ per la data classe di funzioni $f(x, \alpha), \alpha \in \Lambda$ posto di non conoscere $P(x, y)$ e che l'unica informazione disponibile sia contenuta nel training set.

Va notato che la perdita attesa $R(\alpha)$ non può essere quantificato perché è il valore atteso della funzione di perdita $L(y, f(\alpha))$ calcolato sull'intero spazio campionario $X \times Y$ utilizzando l'ignota probabilità $P(x, y)$. E' quindi interpretabile come l'errore atteso commesso dalla nostra approssimazione in generale ossia per ogni possibile coppia (x, y) senza limitarsi alle sole osservazioni campionarie.

Riprendendo l'esempio dello studente $R(\alpha)$ può essere visto come l'errore che ci attendiamo lo studente commetta con la sua regola di risoluzione considerando tutti i possibili problemi che gli si possono presentare. Infatti $R(\alpha)$ non si limita ai soli esempi forniti dal campione e può essere visto come una sorta di capacità di generalizzazione dello studente ossia il fatto di non commettere errori su problemi che non erano contenuti nel libro di addestramento.

Quindi in teoria dovremmo individuare una funzione $f(x, \alpha_0)$ da un dato set di ipotesi (scelto da noi) che rende minima una grandezza $R(\alpha)$ che non possiamo calcolare.

2.Definizione del problema di Pattern Recognition

Nell'introduzione abbiamo accennato al fatto che nella teoria dell'apprendimento statistico esistono tre problemi fondamentali ma che questi non sono altro che casi particolari del più generico problema di apprendere attraverso esempi.

Nel dare una definizione di cosa s'intenda con apprendere nel nostro contesto abbiamo affermato che in sostanza una learning machine è in grado di apprendere quando individua la funzione $f(x, \alpha_0)$ che rende minimo l'errore atteso

$$R(\alpha) = \int L\big(y, f(x, \alpha)\big) dP(x, y)$$

ossia quando è in grado di individuare una funzione che generalizzi il fenomeno osservato permettendogli una buona performance anche fuori dal campione osservato.

Fin'ora però non abbiamo dato alcuna definizione specifica alla funzione di perdita $L(y, f(x, \alpha))$ e ci siamo semplicemente limitati a dire che questa è una qualche misura dell'errore commesso dal modello rispetto agli output osservati sul campione. Questa generalità nel definire L si giustifica per il fatto che a seconda del problema che si affronta si definisce una diversa misura di errore. Ai nostri fini però il problema di apprendimento che più ci interessa è quello della classificazione quindi nel seguito ci concentreremo unicamente su questo caso.

Pattern Recognition: il supervisore ha un comportamento binario del tipo $y = \{0,1\}$ e l'insieme $f(x, \alpha), \alpha \in \Lambda$ rappresenta un set di funzioni indicatrici cioè che a loro volta possono assumere due soli valori a seconda che l'input ricevuto superi un determinato valore soglia. La funzione di perdita è quindi definita come:

$$L(y, f(x, \alpha)) = \begin{cases} 0 \text{ se } y = f(x, \alpha) \\ 1 \text{ se } y \neq f(x, \alpha) \end{cases}$$

data questa funzione di perdita la $R(\alpha)$ rappresenta l'errore atteso di classificazione.

Al fine di snellire la notazione che utilizzeremo più avanti indicheremo con z_i la singola coppia (x_i, y_i) e quindi con Z lo spazio campionario. Avendo definito la funzione di perdita L possiamo quindi ritradurre il problema dell'apprendimento come la ricerca della particolare funzione $Q(z, \alpha)$ dall'insieme $S = \{Q(z, \alpha), \alpha \in \Lambda\}$ tale da rendere minima la scrittura

$$R(\alpha) = \int Q(z, \alpha) dF(z)$$

che rappresenta l'errore atteso di classificazione del modello.

3 Empirical Risk Minimization

Dai capitoli precedenti abbiamo capito cosa si intende con apprendimento ed in particolare si è calato il problema nel caso del pattern recognition definendo nello specifico la funzione di perdita L che misura la discrepanza tra i risultati di modello e quelli osservati.

Dai ragionamenti fatti è però emerso il seguente problema:

apprendere significa individuare una funzione $f(x, \alpha_0)$ da un dato set di ipotesi che rende minima una grandezza $R(\alpha)$ che non possiamo calcolare perché la distribuzione Z da cui si generano i valori e i caratteri dello spazio campionario non saranno mai noti

ma se non possiamo calcolare $R(\alpha)$ come possiamo quantificare la bontà della soluzione ottenuta dal classificatore?

Ritornando all'esempio dello studente, è evidente che non avendo la soluzione dei nuovi problemi non è neanche possibile sapere se effettivamente la sua risposta risulti corretta. D'altro canto una grandezza che possiamo misurare è il numero di errori che lo studente commette durante lo studio perché quest'ultimo si basa su un insieme di esempi per i quali sono disponibili le soluzioni.

Possiamo quindi definire una nuova grandezza che è l'errore empirico ossia quello che farebbe il modello sui valori x del campione per i quali abbiamo l'output y del supervisore.

Formalmente lo definiamo come:

$$R_{emp}(\alpha) = \frac{1}{l}\sum_{i=i}^{l} Q(z_i, \alpha)$$

ove $Q(z_i, \alpha) = L\big(y_i, f(x_i, \alpha)\big)$ per ogni $i = 1, \dots, l$.

L'errore empirico sembrerebbe richiamare la definizione di $R(\alpha)$ ossia di expected (media) degli errori che però viene calato nell'ambito del campione osservato.

L'intuizione quindi è la seguente: è possibile definire un legame tra $R_{emp}(\alpha)$ e $R(\alpha)$?

Se fosse vero che l'errore empirico *approssima* l'errore vero allora tramite una minimizzazione della prima componente sarebbe possibile ottenere un modello che minimizza anche l'errore vero e che quindi è in grado di generalizzare il fenomeno.

Questa è la sostanza del principio di Minimizzazione del Rischio Empirico, quello che faremo ora sarà studiare le condizioni necessarie affinché effettivamente l'andamento di $R_{emp}(\alpha)$ rifletta quello di $R(\alpha)$. In pratica vogliamo capire se la prima misura di errore converge alla seconda.

Una volta accertata tale relazione dovremo chiederci con che velocità effettivamente avviene la convergenza (ossia se nella pratica il principio è applicabile anche con una dimensione campionaria ridotta) e come sia possibile controllare il processo.

Infine sarà fondamentale utilizzare i risultati della teoria per poter costruire degli algoritmi di ricerca della funzione ottimale dal set di ipotesi che siano in grado di controllare il processo di convergenza.

3.1 Condizioni di Consistenza

L'obiettivo è capire sotto quali condizioni l'errore empirico commesso dal modello riflette il valore di $R(\alpha)$. Ciò si traduce nel verificare quando l'errore empirico risulta essere consistente ossia che all'aumentare dell'informazione la sua distribuzione tende a concentrarsi nell'intorno di $R(\alpha)$.

Sia dunque $Q(z, \alpha_l)$ la funzione che rende minimo l'errore empirico $R_{emp}(\alpha)$, allora l'ERM è consistente sul set di ipotesi S e per una data distribuzione Z quando si verificano contemporaneamente le seguenti condizioni:

$$R(\alpha_l) \xrightarrow{P} T$$

$$R_{emp}(\alpha) \xrightarrow{P} T$$

con $T = \inf_{\alpha \in \Lambda} R(\alpha)$.

La prima equazione ci dice che il rischio reale calcolato con la funzione che rende minimo il $R_{emp}(\alpha)$ converge in probabilità al più basso valore assumibile da $R(\alpha)$ su quel set di ipotesi (ossia T) al divergere della numerosità campionaria.

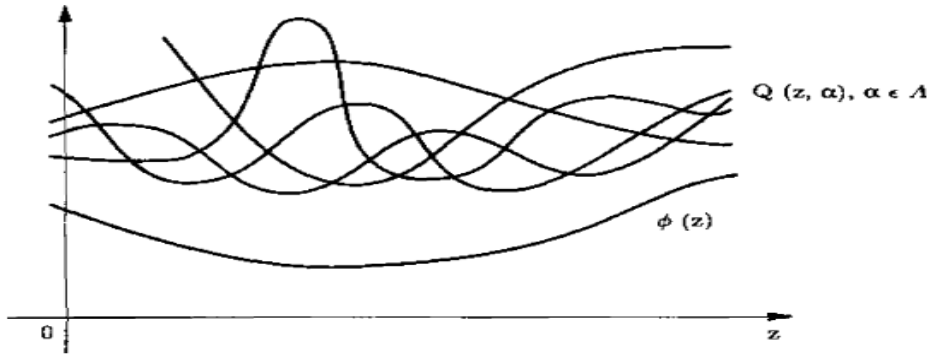
La seconda equazione invece ci garantisce che sia possibile stimare T dal il rischio empirico.

Questa definizione di consistenza però non si adatta ai nostri scopi perché richiederebbe uno studio delle caratteristiche delle funzioni contenute nel set di ipotesi scelto. Infatti si consideri un set S che non contiene alcuna funzione per cui l'ERM sia consistente.

Si supponga quindi di aggiungere a S una nuova funzione $\varphi(z)$ tale per cui

$$\inf_{\alpha \in \Lambda} Q(z, \alpha) > \varphi(z) \; \forall z$$

Quindi la nuova funzione sta sempre sotto a tutte le funzioni definite in S quale che sia il valore di z considerato.



Consideriamo ora un nuovo set di ipotesi $S^* = \{S, \varphi(z)\}$, per questo insieme l'ERM è consistente infatti per ogni spazio Z e numerosità campionaria l la funzione che rende minimo il rischio empirico è ottenuta da $\varphi(z)$ che è anche la funzione che minimizza $R(\alpha)$.

Chiamiamo questa situazione consistenza non stretta, a noi però serve una proprietà che non ci obblighi ogni volta a dover controllare se esistono funzioni minoritarie nel set di ipotesi. Per farlo si osservi che la consistenza non stretta è dovuta al fatto che esista una funzione che sta sempre sotto a tutte le altre funzioni di S, dunque perché l'ERM sia consistente in senso stretto deve valere una convergenza uniforme per ogni sottoinsieme di funzioni di S e livelli di errori.

Sia dunque $\Lambda(\mathrm{c}) = \{\alpha: R(\alpha) > c, \alpha \in \Lambda\}$ l'insieme di tutte le funzioni di S il cui livello di errore e maggiore ad un valore c. Affermiamo che l'ERM è strettamente consistente su S e Z quando:

$$\inf_{\alpha \in \Lambda(\mathrm{c})} R_{emp}(\alpha) \xrightarrow{P} T(c) \quad \forall c \in R$$

ove $T(c) = \inf_{\alpha \in \Lambda(\mathrm{c})} R(\alpha)$.

In sostanza per ogni insieme non vuoto di funzioni definite in base ad un valore soglia c la soluzione empirica converge al $R(\alpha)$ più basso per quel particolare set di funzioni.

Si consideri il seguente set di funzioni:

$$\Lambda(\mathrm{T} + \varepsilon) = \{\alpha: R(\alpha) > \mathrm{T} + \varepsilon, \alpha \in \Lambda\} \; con \; T = \inf_{\alpha \in \Lambda} R(\alpha)$$

e si assuma vera la condizione

$$\inf_{\alpha \in \Lambda(\mathrm{T}+\varepsilon)} R_{emp}(\alpha) \xrightarrow{P} T(c)$$

Allora sono vere le seguenti affermazioni:

$$\lim_{l \to \infty} P\left[R_{emp}(\alpha_l) \geq \mathrm{T} + \frac{\varepsilon}{2}\right] = 0$$

Perché se $R_{emp}(\alpha_l)$ è il più basso rischio empirico ottenibile su tutto S questo per la consistenza stretta tenderà al minimo assoluto T che è minore di $T + \frac{\varepsilon}{2}$.

$$\lim_{l \to \infty} P\left[\inf_{\alpha \in \Lambda(\mathrm{T}+\varepsilon)} R_{emp}(\alpha) \geq \mathrm{T} + \frac{\varepsilon}{2}\right] = 1$$

Perché se $R_{emp}(\alpha)$ è scelto nel insieme $\Lambda(\mathrm{T} + \varepsilon)$ per definizione di consistenza stretta tenderà ad un valore di errore $T + \varepsilon > T + \frac{\varepsilon}{2}$.

Se quindi è impossibile che α_l dia un errore maggiore di $T + \frac{\varepsilon}{2}$ e nell'insieme $\Lambda(\mathrm{T} + \varepsilon)$ ho solo funzioni il cui errore è strettamente superiore a $T + \frac{\varepsilon}{2}$ allora varrà che

$$\lim_{l \to \infty} P\left[\alpha_l \in \Lambda(\mathrm{T} + \varepsilon)\right] = 0$$

Se quindi $\alpha_l \notin \Lambda(\mathrm{T} + \varepsilon)$ allora $R(\alpha_l)$ non può superare $\mathrm{T} + \varepsilon$ e contemporaneamente non può essere inferiore al limite assoluto T dunque

$$T \leq R(\alpha_l) \leq \mathrm{T} + \epsilon$$

quindi $R(\alpha_l)$ converge al minimo assoluto T al divergere della dimensione campionaria.

Abbiamo dimostrato che se è verificata la convergenza stretta l'unica condizione che ci interessa è data dalla

$$R_{emp}(\alpha) \xrightarrow{P} T$$

perché la convergenza in probabilità di $R(\alpha_l)$ a T è implicata dalla definizione di consistenza stretta. Per poter verificare le condizioni affinché l'ERM sia consistente basta studiare il comportamento del rischio empirico rispetto al minimo assoluto T.

Notare che questo risultato è stato ottenuto assumendo vera la consistenza stretta sopra indicata ma non abbiamo detto nulla sulle condizioni affinché questa si verifichi.

Posto quindi che a noi interessa il caso di convergenza stretta ed avendo dimostrato che in essa interessa unicamente che

$$\inf_{\alpha \in \Lambda} R_{emp}(\alpha) \xrightarrow{P} T$$

lo studio delle condizioni necessarie e sufficienti perché l'ERM sia strettamente consistente si basano sullo studiare la variabile

$$R(\alpha) - R_{emp}(\alpha)$$

ma volendo definire una condizione che ci assicuri la consistenza dell'ERM sull'intero set di funzioni S studieremo il caso

$$\lim_{l\to\infty} P\left[\Delta(\alpha_{worst}) > \varepsilon\right] = 0 \quad con\ \Delta(\alpha_{worst}) = \sup_{\alpha\in\Lambda}\left(R(\alpha) - R_{emp}(\alpha)\right)$$

In pratica se l'ERM è consistente rispetto alla funzione peggiore di S (ossia quella che da la maggior distanza tra rischio empirico e reale) allora l'ERM è consistente su tutto S.

3.1.1 Teorema Chiave

Viene presentato ora uno dei risultati fondamentali della teoria dell'apprendimento statistico. Il teorema in sostanza ci dice che se si verifica una determinata convergenza in probabilità allora l'ERM è consistente.

Enunciato

Se per S vale che $A \leq R(\alpha) \leq B$ allora perché l'ERM sia consistente è necessario e sufficiente che

$$\lim_{l\to\infty} P\left[\Delta(\alpha_{worst}) > \varepsilon\right] = 0$$

Dimostrazione.

Sia l'ERM strettamente consistente su S, allora è vero che

$$\inf_{\alpha\in\Lambda(c)} R_{emp}(\alpha) \xrightarrow{P} T(c) \quad \forall c > 0$$

si consideri un set finito di numeri $a_1, \dots, a_n$ tali per cui $|a_{i+1} - a_i| < \varepsilon/2\ \forall i$ ove $a_1 = A$ e $a_n = B$.

Si consideri il generico evento T_k definito come

$$T_k = \left\{\inf_{\alpha\in\Lambda(\alpha_k)} R_{emp}(\alpha) < T(\alpha_k) - \frac{\varepsilon}{2}\right\} \quad con\ \Lambda(\alpha_k) = \{\alpha: R(\alpha) > \alpha_k, \alpha \in \Lambda\}$$

evidentemente $P(T_k) \to 0$ perché il miglior rischio empirico ottenibile su quel set di funzioni tenderà a $T(\alpha_k)$ che è il minimo valore assumibile $R(\alpha)$ su $\Lambda(\alpha_k)$, quindi il rischio empirico non potrà essere inferiore a $T(\alpha_k)$ ne tanto meno a $T(\alpha_k) - \frac{\varepsilon}{2}$. Dato che quest'ultima condizione vale per ogni insieme k-esimo avremo che:

$$P\left(\bigcup_{k=1}^{n} T_k\right) \to 0$$

Si consideri ora l'evento $A = \{\Delta(\alpha_{worst}) > \varepsilon\}$ ove $\Delta(\alpha_{worst})$ considera il caso peggiore su tutto S. Se A accade vorrà dire che $\exists\ \alpha^* \in \Lambda$ tale per cui

$$R(\alpha^*) - R_{emp}(\alpha^*) > \varepsilon \Rightarrow R(\alpha^*) - \varepsilon > R_{emp}(\alpha^*)$$

Notare che α^* non è necessariamente la funzione peggiore perché la diseguaglianza $\Delta(\alpha) > \varepsilon$ potrebbe essere verificata da un più di una funzione; questa funzione α^* sarà contenuta in un subset di S del tipo

$$\Lambda(\alpha_k) = \{\alpha : R(\alpha) > \alpha_k, \alpha \in \Lambda\}$$

per cui varrà che

$$R(\alpha^*) > \alpha_k \Rightarrow R(\alpha^*) - \alpha_k > 0 \Rightarrow R(\alpha^*) - \alpha_k < \frac{\varepsilon}{2}$$

infatti $\alpha^* \in \Lambda(\alpha_k)$ e il livello minimo di questo insieme è definito da α_k ma per assunzione la struttura dei diversi subset è tale per cui la distanza tra due livelli minimi adiacenti non possa superare il valore $\varepsilon/2$.

Sostituendo ad α_k il limite inferiore di $R(\alpha)$ sull'insieme $\Lambda(\alpha_k)$ non modifichiamo la diseguaglianza di cui sopra (sempre per il fatto che la distanza tra due livelli minimi adiacenti non possa superare il valore $\varepsilon/2$) e quindi

$$R(\alpha^*) - \inf_{\alpha \in \Lambda(\alpha_k)} R(\alpha) < \frac{\varepsilon}{2}$$

$$- \inf_{\alpha \in \Lambda(\alpha_k)} R(\alpha) < -R(\alpha^*) + \frac{\varepsilon}{2}$$

$$\inf_{\alpha \in \Lambda(\alpha_k)} R(\alpha) > R(\alpha^*) - \frac{\varepsilon}{2}$$

sottraendo $\frac{\varepsilon}{2}$ ad entrambi i lati della diseguaglianza e riprendendo la definizione di α^*

$$\inf_{\alpha \in \Lambda(\alpha_k)} R(\alpha) - \frac{\varepsilon}{2} > R(\alpha^*) - \varepsilon > R_{emp}(\alpha^*) \geq \inf_{\alpha \in \Lambda(\alpha_k)} R_{emp}(\alpha)$$

quindi

$$\inf_{\alpha \in \Lambda(\alpha_k)} R(\alpha) - \frac{\varepsilon}{2} > \inf_{\alpha \in \Lambda(\alpha_k)} R_{emp}(\alpha)$$

$$T(\alpha_k) - \frac{\varepsilon}{2} > \inf_{\alpha \in \Lambda(\alpha_k)} R_{emp}(\alpha)$$

ma quest'ultima diseguaglianza è la definizione dell'evento T_k la cui probabilità di accadimento per l che diverge a più infinito è zero. Visto quindi che l'evento A implica l'evento T_k dovrà valere che

$$\lim_{l \to \infty} P\left[\Delta(\alpha_{worst}) > \varepsilon\right] = 0$$

Abbiamo quindi dimostrato che esiste un legame tra la condizione di consistenza stretta dell'ERM e la convergenza uniforme *one side* tra $R_{emp}(\alpha)$ ed $R(\alpha)$.

Per poter utilizzare più avanti un importante risultato della teoria probabilistica (la diseguaglianza di Hoeffding) vogliamo dimostrare che la consistenza stretta ha un legame analogo con la convergenza uniforme two sides ossia quella che considera $\Delta|\alpha_{worst}|$ al posto di $\Delta(\alpha_{worst})$.

Sia A l'evento $A = \left\{\left|\inf_{\alpha \in \Lambda(c)} R_{emp}(\alpha) - T(c)\right| > \varepsilon\right\}$ a fronte del valore assoluto lo possiamo scomporre nell'unione di due eventi disgiunti

$$A_1 = \left\{T(c) + \varepsilon < \inf_{\alpha \in \Lambda(c)} R_{emp}(\alpha)\right\}$$

$$A_2 = \left\{T(c) - \varepsilon > \inf_{\alpha \in \Lambda(c)} R_{emp}(\alpha)\right\}$$

con $A = A_1 \cup A_2$.

Supponiamo che accada A_1 e consideriamo un α^* per cui valga $R(\alpha^*) < T(c) + \varepsilon/2$ allora vale che

$$R(\alpha^*) + \frac{\varepsilon}{2} < T(c) + \varepsilon < \inf_{\alpha \in \Lambda(c)} R_{emp}(\alpha) < R_{emp}(\alpha^*)$$

quindi

$$R(\alpha^*) + \frac{\varepsilon}{2} < R_{emp}(\alpha^*) \Rightarrow R_{emp}(\alpha^*) - R(\alpha^*) > \frac{\varepsilon}{2}$$

La probabilità di quest'ultima diseguaglianza non può essere minore di quella dell'evento A_1 (perché α^* è tale per cui $R(\alpha^*) < T(c) + \varepsilon/2$) ed assumendo vera

$$\lim_{l \to \infty} P\left[\Delta(\alpha_{worst}) > \varepsilon\right] = 0$$

avremo

$$P(A_1) \leq P\left(R_{emp}(\alpha^*) - R(\alpha^*) > \frac{\varepsilon}{2}\right) \to 0$$

Supponiamo che accada A_2 e consideriamo un α^{**} tale per cui

$$R_{emp}(\alpha^{**}) + \frac{\varepsilon}{2} < T(c) < R(\alpha^{**})$$

quindi

$$R_{emp}(\alpha^{**}) + \frac{\varepsilon}{2} < R(\alpha^{**})$$

$$R_{emp}(\alpha^{**}) - R(\alpha^{**}) < -\frac{\varepsilon}{2}$$

$$R(\alpha^{**}) - R_{emp}(\alpha^{**}) > \frac{\varepsilon}{2}$$

ne segue che

$$P(A_2) < P\left[R(\alpha^{**}) - R_{emp}(\alpha^{**}) > \frac{\varepsilon}{2}\right] < P\left[\sup_{\alpha \in \Lambda(c)}\left(R(\alpha^{**}) - R_{emp}(\alpha^{**})\right) > \frac{\varepsilon}{2}\right] \xrightarrow{l\to\infty} 0$$

$$P(A_2) \xrightarrow{l\to\infty} 0$$

Essendo A l'unione dei due sotto eventi avremo che $P(A) = P(A_1) + P(A_2)$ ma visto che i due addendi tendono ad azzerarsi al divergere della dimensione campionaria otteniamo

$$\lim_{l\to\infty} P\left\{\left|\inf_{\alpha \in \Lambda(c)} R_{emp}(\alpha) - T(c)\right| > \varepsilon\right\} = 0 \; \forall c$$

Quindi il teorema chiave è di fondamentale importanza perché afferma che *la consistenza stretta dell'ERM si ottiene quando avviene la convergenza uniforme tra il rischio empirico e quello effettivo*. Nei prossimi capitoli sfrutteremo il risultato di questo teorema ed andremo a studiare le condizioni per le quali effettivamente si verifica tale convergenza uniforme sapendo che queste contemporaneamente garantiscono la consistenza stretta del metodo.

3.1.2 Diseguaglianza di Hoeffding

Con il teorema Chiave abbiamo dimostrato che la di consistenza stretta dell'ERM equivale alla convergenza uniforme definita da

$$\lim_{l\to\infty} P\left[\Delta|\alpha_{worst}| > \varepsilon\right] = 0 \ \ con\ \Delta|\alpha_{worst}| = \sup_{\alpha\in\Lambda} \left|R(\alpha) - R_{emp}(\alpha)\right|$$

ora dobbiamo capire sotto quali condizioni si verifica quest'ultima condizione, per farlo avremo bisogno dei seguenti risultati:

1.Teorema di Bernoulli

al divergere della dimensione campionaria la frequenza con cui accade un evento su $z_1, \dots, z_l$ converge alla probabilità dell'evento stesso.

2.Diseguaglianza di Hoeffding

definendo con $P\{Q(z,\alpha) > 0\}$ la probabilità dell'evento e con $v_l\{Q(z_l,\alpha) > 0\}$ la frequenza osservata vale che

$$P\{|P\{Q(z,\alpha) > 0\} - v_l\{Q(z_l,\alpha) > 0\}| > \varepsilon\} \le 2e^{-2\varepsilon^2 l}$$

con $l \in N$ e $\varepsilon > 0$.

L'analogia con il risultato del teorema chiave è evidente, infatti il nostro obiettivo è proprio quello di capire qualcosa sulla probabilità $R(\alpha)$ a partire dalla frequenza osservata $R_{emp}(\alpha)$.

Notare come la diseguaglianza di Hoeffding definisca un limite superiore che è unicamente funzione del grado di precisione desiderato (ε) e dalla dimensione campionaria l.

Considerando quindi un insieme di funzioni $S = \{Q(z,\alpha), \alpha \in \Lambda\}$ a cardinalità finita pari a N ($|S| = N$ quindi il numero di funzioni contenute in S è finito e pari a N) possiamo riscrivere la Hoeffding calandola nel nostro caso

$$P\{\Delta|\alpha_{worst}| > \varepsilon\} \le \sum_{k=1}^{N} P\{\Delta|\alpha_k| > \varepsilon\} \le \sum_{k=1}^{N} 2e^{-2\varepsilon^2 l} = 2Ne^{-2\varepsilon^2 l}$$

ove

$$\Delta|\alpha_{worst}| = \max_{1\le k\le N} \left|R(\alpha) - R_{emp}(\alpha)\right|$$

$$\Delta|\alpha_k| = \left|R(\alpha_k) - R_{emp}(\alpha_k)\right|$$

Il risultato di cui sopra si giustifica dal fatto che la probabilità del caso peggiore su S sarà comunque minore o al più uguale alla somma delle probabilità di tutti gli elementi dell'insieme (a maggior ragione se non si considerano tutti gli eventi intersezione) perché il caso peggiore è comunque un di cui di S.

Considerando poi che la diseguaglianza di Hoeffding definisce un limite superiore che non fa riferimento ad una specifica funzione di S giustifichiamo l'ultimo risultato.

Abbiamo quindi ottenuto che

$$P\{\Delta|\alpha_{worst}| > \varepsilon\} \leq 2Ne^{-2\varepsilon^2 l}$$

che possiamo riscrivere come

$$P\{\Delta|\alpha_{worst}| > \varepsilon\} \leq 2e^{\left[\left(\frac{\ln(N)}{l}-2\varepsilon^2\right)l\right]}$$

Quest'ultima equazione è fondamentale e ci permette di ricollegarci a quanto visto nel capitolo sul Teorema Chiave. Avevamo infatti detto che

$$ERM\ strettamente\ consistente\ se\ \lim_{l\to\infty} P\left[\Delta(\alpha_{worst}) > \varepsilon\right] = 0$$

ora abbiamo definito un limite superiore alla $P\{\Delta|\alpha_{worst}| > \varepsilon\}$.

La domanda che dobbiamo farci è: quand'è che la parte a destra della diseguaglianza ricavata da Hoeffding si azzera per l che diverge a più infinito?

$$\lim_{l\to\infty} 2e^{\left[\left(\frac{\ln(N)}{l}-2\varepsilon^2\right)l\right]} = 0$$

Si osservi che l'argomento dell'esponenziale andrebbe a $-\infty$ quando $\lim_{l\to\infty}\frac{\ln(N)}{l} = 0$ infatti rimarrebbe solamente $e^{[(-2\varepsilon^2)l]}$ e nonostante ε sia un valore contenuto in termini operativi non potrà essere *troppo* contenuto e quindi al divergere di l otterremmo l'esponenziale di meno infinito che è zero.

Mettendo insieme il Teorema Chiave e quest'ultimo risultato otteniamo quindi che

$$ERM\ strettamente\ consistente\ se\ \lim_{l\to\infty}\frac{\ln(N)}{l} = 0$$

Questa affermazione è sicuramente interessante ma poco utile ai fini applicativi perché si ricordi che con N indichiamo la cardinalità dell'insieme S.

Segue dunque che solo quando S ha un numero finito di elementi otteniamo la consistenza stretta dell'ERM ma basti pensare al più semplice set di ipotesi (classificatore lineari) per capire che il risultato di cui sopra risulta essere difficilmente applicabile.

In sostanza il risultato è applicabile solo quando ci limitiamo a cercare la soluzione di ottimo in un set finito di funzioni ma praticamente ogni insieme di ipotesi utilizzabile ai nostri fini ha cardinalità infinita.E' quindi necessario definire una misura alternativa alla mera cardinalità dell'insieme per definire il *grado di complessità di un set di ipotesi*

Questo sarà l'oggetto del prossimo capitolo ma prima è bene studiare più approfonditamente il risultato dell'ultima disequazione.

$$P\{\Delta|\alpha_{worst}| > \varepsilon\} \leq 2e^{\left[\left(ln\frac{N}{l}-2\varepsilon^2\right)l\right]} = \eta \Rightarrow \varepsilon = \sqrt{\frac{\ln N - \ln \eta}{2l}}$$

quindi

$$P\left\{\Delta|\alpha_{worst}| > \sqrt{\frac{\ln N - \ln \eta}{2l}}\right\} \leq \eta$$

e visto che questo è il caso peggiore possiamo dire che con probabilità $1-\eta$ accadrà che

$$\Delta|\alpha_k| \leq \sqrt{\frac{\ln N - \ln \eta}{2l}} \qquad \forall k$$

Siano dunque

1. $Q\left(z, \alpha_{k(0)}\right)$ la funzione che rende minimo $R(\alpha)$
2. $Q\left(z, \alpha_{k(l)}\right)$ la funzione che rende minimo $R_{emp}(\alpha)$

se l'ultima diseguaglianza vale per ogni funzione di S allora vale anche per $Q\left(z, \alpha_{k(l)}\right)$ e quindi con probabilità $1-\eta$ vale che

$$R\left(\alpha_{k(l)}\right) - R_{emp}\left(\alpha_{k(l)}\right) \leq \sqrt{\frac{\ln N - \ln \eta}{2l}} = \delta \Rightarrow R\left(\alpha_{k(l)}\right) \leq R_{emp}\left(\alpha_{k(l)}\right) + \delta$$

il valore assoluto è stato tolto perché se $\alpha_{k(l)}$ rende minimo $R_{emp}(\alpha)$ allora vale che $R\left(\alpha_{k(l)}\right) > R_{emp}\left(\alpha_{k(l)}\right)$.

Abbiamo quindi definito un limite superiore (in probabilità) del rischio reale (out of sample) per una data funzione $Q\left(z, \alpha_{k(l)}\right)$.

Per capire quanto il rischio di una funzione selezionata (ossia quella che rende minimo l'errore empirico) sia vicino al minimo assoluto si noti che per $Q\left(z, \alpha_{k(0)}\right)$ vale che

$$P\left\{R_{emp}\left(\alpha_{k(0)}\right) - R\left(\alpha_{k(0)}\right) > \varepsilon\right\} \leq 2e^{-2\varepsilon^2 l} = \eta \Rightarrow \varepsilon = \sqrt{\frac{-\ln \eta}{2l}}$$

infatti se consideriamo una funzione che rende minimo $R(\alpha)$ questo sarà sicuramente più piccolo del rischio empirico calcolato su quella stessa funzione.

Con probabilità $1 - \eta$ varrà che

$$R_{emp}\left(\alpha_{k(0)}\right) - R\left(\alpha_{k(0)}\right) < \varepsilon \Rightarrow R\left(\alpha_{k(0)}\right) \geq R_{emp}\left(\alpha_{k(0)}\right) - \sqrt{\frac{-\ln \eta}{2l}}$$

Considerando che per definizione $R\left(\alpha_{k(0)}\right) < R\left(\alpha_{k(l)}\right)$ e unendo i due risultati otteniamo

$$R_{emp}\left(\alpha_{k(0)}\right) - \sqrt{\frac{-\ln \eta}{2l}} \leq R\left(\alpha_{k(0)}\right) < R\left(\alpha_{k(l)}\right) \leq R_{emp}\left(\alpha_{k(l)}\right) + \delta$$

$$\left[R\left(\alpha_{k(l)}\right) - R_{emp}\left(\alpha_{k(l)}\right)\right] - \left[R\left(\alpha_{k(0)}\right) - R_{emp}\left(\alpha_{k(0)}\right)\right] \leq \delta + \sqrt{\frac{-\ln \eta}{2l}}$$

$$\left[R\left(\alpha_{k(l)}\right) - R\left(\alpha_{k(0)}\right)\right] + \left[R_{emp}\left(\alpha_{k(0)}\right) - R_{emp}\left(\alpha_{k(l)}\right)\right] \leq \delta + \sqrt{\frac{-\ln \eta}{2l}}$$

ed essendo $R_{emp}\left(\alpha_{k(0)}\right) - R_{emp}\left(\alpha_{k(l)}\right)$ per definizione positivo otteniamo

$$R\left(\alpha_{k(l)}\right) - R\left(\alpha_{k(0)}\right) \leq \delta + \sqrt{\frac{-\ln \eta}{2l}}$$

Quindi la distanza tra il rischio vero calcolato sulla mia funzione (ossia quella che rende minimo il rischio empirico) ed il minimo assoluto è superiormente limitata dalla grandezza a destra della diseguaglianza.

Possiamo quindi affermare che con probabilità $1 - 2\eta$ valga che

$$\Delta\left(\alpha_{k(l)}\right) \leq \sqrt{\frac{\ln N - \ln \eta}{2l}} + \sqrt{\frac{-\ln \eta}{2l}}$$

Ricapitolando, nel caso di set di ipotesi S a cardinalità finita l'ERM risulta è strettamente consistente quando

$$\lim_{l\to\infty} \frac{\ln(N)}{l} = 0$$

Inoltre si dimostra che il limite superiore del rischio atteso ottenibile con la funzione $Q(z, \alpha_{k(l)})$ che rende minimo l'errore empirico $R_{emp}(\alpha)$ è pari a

$$R(\alpha_{k(l)}) \leq R_{emp}(\alpha_{k(l)}) + \sqrt{\frac{\ln N - \ln \eta}{2l}}$$

mentre al distanza tra l'errore atteso effettivamente ottenuto ed il minimo assoluto T è superiormente limitata da

$$R(\alpha_{k(l)}) - T = \Delta(\alpha_{k(l)}) \leq \sqrt{\frac{\ln N - \ln \eta}{2l}} + \sqrt{\frac{-\ln \eta}{2l}}$$

Le ultime due equazioni forniscono tutte le informazioni sulla **capacità di generalizzazione di una funzione ottenuta con il principio di ERM** nel caso in cui $|S| = N$.

Si osservi che nel caso in cui l diverga a più infinito al condizione che garantisce la consistenza stretta dell'ERM implicherebbe anche che

$$R_{emp}(\alpha_{k(l)}) + \sqrt{\frac{\ln N - \ln \eta}{2l}} \to R_{emp}(\alpha_{k(l)}) \Rightarrow R(\alpha_{k(l)}) \leq R_{emp}(\alpha_{k(l)})$$

perché se $\lim_{l\to\infty} \frac{\ln(N)}{l} = 0$ anche η tende a zero ed il numeratore di quanto sopra è più lento nel tendere a meno infinito del denominatore. Per la seconda diseguaglianza avremo che

$$\sqrt{\frac{\ln N - \ln \eta}{2l}} + \sqrt{\frac{-\ln \eta}{2l}} \to 0 \Rightarrow R(\alpha_{k(l)}) \to T$$

Il primo risultato ci dice che se la dimensione campionaria è sufficientemente grande attraverso la minimizzazione del rischio empirico siamo anche in grado di ottenere una buona generalizzazione perché il fattore relativo alla complessità del set di ipotesi viene meno. Questa buona generalizzazione è a sua volta confermata dal fatto che il rischio effettivo che otteniamo converge al minimo assoluto T.

3.1.3 Un ritorno alla metafora dello studente

Nel capitolo precedente abbiamo ottenuto importanti risultati che ci confermano come il principio di minimizzazione del rischio empirico possa essere una soluzione al problema dell'apprendimento tramite esempi.

Ritornando al nostro esempio dello studente possiamo ritradurre il risultato di cui sopra nel seguente ragionamento: l'errore che lo studente commette in generale (su problemi diversi da quelli usati nella fase di studio) è superiormente limitato dalla somma di due componenti:

1. l'errore commesso sugli esercizi per i quali ha la soluzione
2. la complessità della regola teorica che adottata per risolvere i problemi

Più lo studente definisce una regola complessa e articolata e maggiore è la sua capacità di risolvere correttamente gli esercizi. Questa maggior performance è però garantita solo sugli esercizi già trattati perché la regola di risoluzione è stata formulata sulla base di questi.

Infatti una regola molto complessa è anche molto specifica per i problemi sui quali lo studente si è preparato quindi difficilmente andrà bene "in generale" ossia per nuovi problemi che gli si presentano.

Una regola più generale sarebbe in grado di dare una soluzione accettabile (ma non perfettamente corretta) ad un più vasto insieme di problemi al costo di non riuscire a risolvere perfettamente gli esercizi di addestramento.

Se però il numero di esempi con cui lo studente si esercita diverge a più infinito allora è come se lo studente avesse già una esperienza di tutti i possibili problemi che si possono presentare e quindi in questo caso l'importante è definire una regola che sbagli il meno possibile in fase di addestramento.

3.1.4 Misurare la complessità di un set di ipotesi

Come accennato nel capitolo sulla diseguaglianza di Hoeffding il risultato fin'ora ottenuto è caratterizzato dal fatto di essere valido solo quando la ricerca della funzione di ottimo viene fatta da un insieme di ipotesi S con un numero di elementi N finito.

Si è inoltre fatto cenno al fatto che normalmente un set di ipotesi adottabile nell'ambito del pattern recognition potrebbe essere quello dei classificatori lineari che è una famiglia di funzioni a cardinalità infinita.

Cosa accadrebbe alla diseguaglianza di Hoeffding in questo caso? Sostituendo

$$P\{\Delta|\alpha_{worst}| > \varepsilon\} \leq 2Ne^{-2\varepsilon^2 l}$$

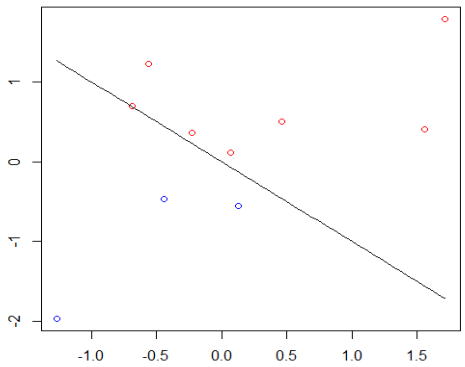
se $N = \infty$ la diseguaglianza di cui sopra diventa

$$P\{\Delta|\alpha_{worst}| > \varepsilon\} \leq \infty$$

il che non è molto informativo ai nostri fini.

Serve quindi una grandezza alternativa a N che inserita nella diseguaglianza di Hoeffding ci dia un limite finito sul quale ricercheremo una condizioni perché si azzeri al divergere di l.

Per meglio comprendere il seguente ragionamento si consideri un esempio: si supponga di avere 10 punti su uno piano cartesiano e che questi siano suddivisioni in due categorie rappresentate dai colori rosso e blu a seconda che si trovino al di sopra o al di sotto della retta definita come $y = -x$.

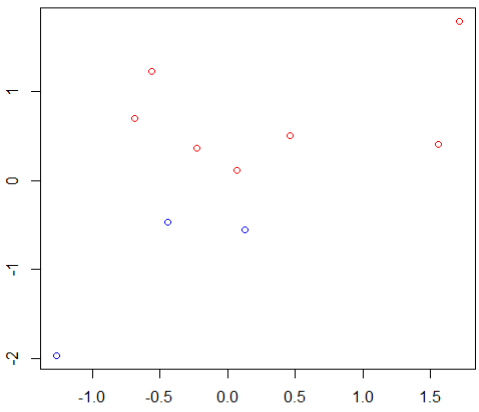


Notare che qui stiamo assumendo di conoscere la regola reale che genera le osservazioni campionarie e quindi possiamo definire con certezza la classe di appartenenza di ogni punto.

Si osservi che pur conoscendo la regola reale vi sono altre rette diverse da quella indicata che permetterebbero una classificazione identica a quella del grafico.

Immaginando di poter ruotare la retta rappresentata nel grafico notiamo che esistono infinite rette diverse da $y = -x$ che restituiscono la stessa classificazione di ogni punto ma non appena "ci scontriamo" con una osservazione qualsiasi si modificherebbe la sua classificazione e quindi la performance complessiva.

Il problema è che nella realtà non si conosce la vera regola che genera i dati e l'unica cosa che possiamo osservare è il grafico seguente.



Supponendo di utilizzare come set di ipotesi l'insieme dei classificatori lineari l'algoritmo di ricerca della soluzione di ottimo cercherà di minimizzare l'errore empirico e quindi otterremmo una retta che classifica correttamente tutti i punti ma che non sarà pari a $y = -x$.

Ai nostri occhi finché esistono infiniti classificatori lineari che applicati ai dati del training set non modificherebbero la classificazione effettivamente osservata.

Possiamo quindi considerare questo insieme di infinite rette che danno la stessa classificazione come una sorta di subset del set di ipotesi di partenza.

Ne segue che il set di ipotesi S a cardinalità infinita può essere in qualche modo ricondotto all'unione di un *numero finito di subset di ipotesi* ove in ogni sottoinsieme si ha un numero infinito di funzioni che però danno al stessa classificazione del training set.

L'intuizione è quella di sostituire nella diseguaglianza di Hoeffding alla N una grandezza che dipende dal numero di questi sottoinsiemi contenuti in S. Questo ci permetterebbe di ritornare al caso a cardinalità finita pur utilizzando set di infinite ipotesi.

Ma quanti subset sono contenuti in S?

Ricordando che il singolo subset viene definito come l'insieme di rette che danno la stessa classificazione dei punti e visto che la classificazione complessiva cambia non appena la nostra retta (ma vale per una qualsiasi funzione) si scontra con una osservazione qualsiasi è intuitivo pensare che il numero di subset sia legato in qualche modo a quante classificazioni diverse si possono ottenere su l punti.

Fatte queste premesse possiamo iniziare ad introdurre una notazione più formale, si definisce set di ipotesi H l'insieme di funzioni

$$\mathrm{H} = \{h: X \to [-1, +1]\} \; con \; |\mathrm{H}| = \infty$$

si definisce dicotomia l'insieme

$$\mathrm{H}(x_1, \dots, x_l) = \{h: \{x_1, \dots, x_l\} \to [-1, +1]\}$$

Dunque la dicotomia è una sorta di set di ipotesi "ristretta" perché al posto di basarsi sull'intero spazio campionario X ha come dominio i soli punti $(x_1, \dots, x_l)$.

Evidentemente H è costituito da infiniti elementi perché si basa su uno spazio X che a sua volta è formato da infiniti punti. Supponendo per semplicità di applicare h al vettore (x_1, x_2, x_3) il risultato sarà un vettore

$$h\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} h(x_1) \\ h(x_2) \\ h(x_3) \end{bmatrix}$$

che potrà assumere otto diverse modalità (dicotomie)

0	0	0	1	1	1	0	1
0	0	1	0	1	0	1	1
0	1	0	0	0	1	1	1

essendo H a cardinalità infinita posso applicare un numero di funzioni h infinito allo stesso vettore di osservazioni ottenendo tutte le dicotomie possibili. Ma per i motivi detti pocanzi in molte situazioni al variare di h otterrò un identico vettore binario.

Ne segue che il numero **massimo** di dicotomie è definito dal numero dei diversi vettori binari che posso ricavare dalle funzioni del set H

$$|\mathrm{H}(x_1, \dots, x_l)| \leq 2^l$$

ove 2^l è dato dal fatto che ogni osservazione può avere al massimo due diverse classificazioni (nel nostro esempio $2^3 = 8$).

Ma perché abbiamo specificato che 2^l è il massimo numero di dicotomie?

Non dobbiamo dimenticare che con H stiamo considerando un famiglia di funzioni che vogliamo utilizzare per classificare un insieme di osservazioni. E' quindi fondamentale chiedersi in quanti modi diversi un certo tipo di funzioni riesce a classificare un dato insieme di punti.

Ricordando il Perceptron di F. Rosenblatt è noto che un classificatore lineare su due dimensioni non è in grado di classificare correttamente 4 punti. Ne segue dunque che il massimo numero di dicotomie di un classificatore lineare definito su due dimensioni è 3. In generale si dimostra che il numero massimo di punti dicotomie (punti frantumabili) ottenibili da un classificatore lineare su n dimensioni è $n + 1$.

Chiamiamo con

$$N^{\mathrm{H}}(x_1, \dots, x_l) = \max_{x_1,\dots,x_l \in X} |\mathrm{H}(x_1, \dots, x_l)|$$

il più alto numero di dicotomie ottenibili su un generico set di l punti dal set di ipotesi H.

Va notato che $x_1, \dots, x_l$ non rappresenta lo specifico campione osservato ma semplicemente un generico set di punti definiti nello stesso spazio. Il motivo è che qui vogliamo definire una misura di capacità generale del set di ipotesi.

Per non confondere la notazione torniamo a quella utilizzata nel capitolo sulla diseguaglianza di Hoeffding (ove Λ rappresentava il set di ipotesi) e chiamiamo con

$$H^{\Lambda}(z_1, \dots, z_l) = \ln\left(N^{\Lambda}(z_1, \dots, z_l)\right)$$

l'entropia casuale di un set di funzioni Λ.

Questa non è nient'altro che la grandezza $N^{\Lambda}(x_1, \dots, x_l)$ prima indicata riscalata in termini logaritmici e descrive il grado di diversità (complessità) di un set di funzione su un insieme di punti.

Se ora consideriamo $z_1, \dots, z_l$ come un insieme di punti effettivamente osservati e assunti come i.i.d. da una variabile casuale Z ricaviamo che anche $H^{\Lambda}(z_1, \dots, z_l)$ è una variabile aleatoria che quindi dipende dal campione che si è osservato.

Se quindi volessimo misurare il grado di complessità di un set di ipotesi su l punti *in generale,* un indicatore ragionevole è il numero *medio di dicotomie ottenibili con quel set di ipotesi Λ su un insieme di l punti.*

Per calcolare l'expected dovremo assumere che $H^{\Lambda}(z_1, \dots, z_l)$ sia misurabile su Z cioè che ne esista il valore atteso rispetto alla probabilità congiunta $P(z_1, \dots, z_l)$

$$H^{\Lambda}(l) = E_Z\left[\ln\left(N^{\Lambda}(z_1, \dots, z_l)\right)\right]$$

questa quantità viene chiamata entropia di un set Λ di funzioni indicatrici su un campione di dimensione l.

Infine chiamiamo *annealed VC-entropy* (il motivo verrà spiegato in seguito) la grandezza

$$H^{\Lambda}_{ann}(l) = ln\left[E_Z\left(N^{\Lambda}(z_1, \dots, z_l)\right)\right]$$

e per la diseguaglianza di Jensen varrà che

$$H^{\Lambda}(l) \leq H^{\Lambda}_{ann}(l)$$

3.1.5 Sostituire alla cardinalità l'entropia in Hoeffding

Posto quindi che $H^{\Lambda}(l)$ sia una grandezza ragionevole per valutarla capacità di un set di funzioni dobbiamo trovare ora un modo per sostituirla alla N all'interno della diseguaglianza di Hoeffding così da poter derivare le disequazioni che descrivono la capacità di generalizzazione di un set di ipotesi.

La dimostrazione completa del teorema che andremo a trattare è molto lunga e complessa quindi in questa sede ci limiteremo alla intuizione che vi è alla base.

Sia $\varepsilon_0 = \varepsilon - \frac{1}{l}$ e supponiamo di avere un database di $2l$ elementi $z_1, \dots, z_{2l}$ che viene ripartito in maniera casuale in due sotto campioni di dimensione l.

Si dimostra che

$$P\left\{\sup_{\alpha\in\Lambda}\left|\frac{1}{l}\sum_{i=1}^{l}Q(z_i,\alpha) - \frac{1}{l}\sum_{i=l+1}^{2l}Q(z_i,\alpha)\right| > \varepsilon_0\right\} \le 2e^{-{\varepsilon_0}^2 l} \Rightarrow P\{\Delta|\alpha_{worst}^{emp}| > \varepsilon_0\} \le 2e^{-{\varepsilon_0}^2 l}$$

quanto richiama evidentemente la diseguaglianza di Hoeffding come se avessimo assunto che il rischio empirico calcolato sui primi l elementi sia il rischio effettivo ossia

$$\frac{1}{l}\sum_{i=1}^{l}Q(z_i,\alpha) = \int Q(z,\alpha)dF(z)$$

Per ogni campione Z^{2l} esiste un solo set di funzioni distinguibili

$$Q(z,\alpha^*) \in \Lambda(z_1,\dots,z_{2l}) = \Lambda^* : |\Lambda^*| = N^{\Lambda^*}(z_1,\dots,z_{2l})$$

In sostanza dal set di infinite funzioni Λ estraiamo un sottoinsieme finito Λ^* con cardinalità $N^{\Lambda^*}(z_1,\dots,z_{2l})$ che è pari al più alto numero di dicotomie ottenibili con Λ^* su questo set di punti.

Avendo un subset finito possiamo applicare gli stessi ragionamenti visti la prima volta che abbiamo utilizzato la diseguaglianza di Hoeffding:

$$P\{\Delta|\alpha_{worst,\Lambda}^{emp}| > \varepsilon_0 | z_1,\dots,z_{2l}\} = P\{\Delta|\alpha_{worst,\Lambda^*}^{emp}| > \varepsilon_0 | z_1,\dots,z_{2l}\}$$

con

$\Delta|\alpha_{worst,\Lambda}^{emp}| = \sup_{\alpha\in\Lambda}\left|\frac{1}{l}\sum_{i=1}^{l}Q(z_i,\alpha) - \frac{1}{l}\sum_{i=l+1}^{2l}Q(z_i,\alpha)\right|$ caso peggiore su tutto Λ

$\Delta|\alpha_{worst,\Lambda^*}^{emp}| = \sup_{\alpha\in\Lambda^*}\left|\frac{1}{l}\sum_{i=1}^{l}Q(z_i,\alpha) - \frac{1}{l}\sum_{i=l+1}^{2l}Q(z_i,\alpha)\right|$ caso peggiore su $\Lambda^* \subset \Lambda$

Essendo Λ^* a cardinalità finita vale che

$$P\{\Delta|\alpha^{emp}_{worst,\Lambda^*}| > \varepsilon_0|z_1,\dots,z_{2l}\} \leq \sum_{\alpha\in\Lambda^*} P\{\Delta|\alpha^{emp}_{k,\Lambda^*}| > \varepsilon_0|z_1,\dots,z_{2l}\}$$

ove gli elementi della sommatoria sono un numero finito pari a $N^{\Lambda^*}(z_1,\dots,z_{2l})$ e quindi essendo il limite superiore uguale per ogni funzione avremo

$$\sum_{\alpha\in\Lambda^*} P\{\Delta|\alpha^{emp}_{k,\Lambda^*}| > \varepsilon_0|z_1,\dots,z_{2l}\} = 2N^{\Lambda^*}(z_1,\dots,z_{2l})e^{-{\varepsilon_0}^2 l} = 2e^{H^{\Lambda}(z_1,\dots,z_l)-{\varepsilon_0}^2 l}$$

Calcoliamo quindi il valore atteso rispetto alla probabilità congiunta $P(z_1,\dots,z_{2l})$ otteniamo

$$P\{\Delta|\alpha^{emp}_{worst,\Lambda^*}| > \varepsilon_0\} = E_Z\left[P\{\Delta|\alpha^{emp}_{worst,\Lambda^*}| > \varepsilon_0|z_1,\dots,z_{2l}\}\right] < 2e^{\left(\frac{H^{\Lambda}_{ann}(2l)}{l}-{\varepsilon_0}^2\right)l}$$

considerando il seguente lemma (la cui dimostrazione non sarà presentata)

$$P\{\Delta|\alpha_{worst}| > \varepsilon\} < 2P\{\Delta|\alpha^{emp}_{worst,\Lambda^*}| > \varepsilon_0\}$$

otteniamo il risultato finale

$$P\{\Delta|\alpha_{worst}| > \varepsilon\} < 4e^{\left[\frac{H^{\Lambda}_{ann}(2l)}{l}-\left(\varepsilon-\frac{1}{l}\right)^2\right]l}$$

Quest'ultima disequazione è estremamente simile a quella vista nel caso di S a cardinalità finita. Dobbiamo quindi chiederci quando la parte a destra della diseguaglianza si azzera per l che diverge a più infinito.

$$\lim_{l\to\infty} 4e^{\left[\frac{H^{\Lambda}_{ann}(2l)}{l}-\left(\varepsilon-\frac{1}{l}\right)^2\right]l} = 0$$

Per motivazioni analoghe a quelle viste quando si trattava $\ln(N)/l$ osserviamo che affinché la parte destra della diseguaglianza si azzeri è necessario e sufficiente che

$$\lim_{l\to\infty} \frac{H^{\Lambda}_{ann}(2l)}{l} = 0$$

Possiamo quindi affermare

$$ERM\ strettamente\ consistente\ se\ \lim_{l\to\infty} \frac{H^{\Lambda}_{ann}(2l)}{l} = 0$$

Sfruttando ragionamenti analoghi a quelli visti nel caso a cardinalità finita possiamo riscrivere la cosa come

$$P\{\Delta|\alpha_{worst}| > \varepsilon\} < 4e^{\left[\frac{H_{ann}^{\Lambda}(2l)}{l} - \left(\varepsilon - \frac{1}{l}\right)^2\right]l} = \eta$$

quindi con probabilità $1 - \eta$ dovrà valere simultaneamente per ogni equazione di S che

$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\frac{H_{ann}^{\Lambda}(2l) - \ln(\eta/4)}{l}} + \frac{1}{l} = R_{emp}(\alpha) + \delta$$

Considerando quindi le funzioni che rendono minimo rispettivamente il rischio empirico e il rischio atteso ricaviamo

$$R(\alpha_l) \leq R_{emp}(\alpha_l) + \delta$$

$$R(\alpha_0) > R_{emp}(\alpha_0) - \sqrt{\frac{-\ln\eta}{2l}}$$

ed essendo $R(\alpha_0) < R(\alpha_l)$ e $R_{emp}(\alpha_l) < R_{emp}(\alpha_0)$ otteniamo con probabilità $1 - 2\eta$

$$R(\alpha_l) - T = \Delta(\alpha_l) \leq \sqrt{\frac{H_{ann}^{\Lambda}(2l) - \ln(\eta/4)}{l}} + \frac{1}{l} - \sqrt{\frac{-\ln\eta}{2l}}$$

Ricapitolando, nel caso di set di ipotesi S a cardinalità infinita l'ERM risulta è strettamente consistente quando

$$\lim_{l\to\infty} \frac{H_{ann}^{\Lambda}(2l)}{l} = 0$$

ove $H^{\Lambda}(2l)$ è una misura di capacità del set di funzioni Λ alternativa alla cardinalità.

Inoltre si dimostra che il limite superiore del rischio atteso ottenibile con la funzione $Q(z, \alpha_l)$ che rende minimo l'errore empirico $R_{emp}(\alpha)$ è pari a

$$R(\alpha_l) \leq R_{emp}(\alpha_l) + \sqrt{\frac{H_{ann}^{\Lambda}(2l) - \ln(\eta/4)}{l}} + \frac{1}{l}$$

mentre al distanza tra l'errore atteso effettivamente ottenuto ed il minimo assoluto T è superiormente limitata da

$$R(\alpha_l) - T = \Delta(\alpha_l) \leq \sqrt{\frac{H_{ann}^{\Lambda}(2l) - \ln(\eta/4)}{l}} + \frac{1}{l} - \sqrt{\frac{-\ln\eta}{2l}}$$

Le ultime due equazioni forniscono tutte le informazioni sulla **capacità di generalizzazione di una funzione ottenuta con il principio di ERM** nel caso in cui $|S| = +\infty$.

Si osservi che nel caso in cui l diverga a più infinito al condizione che garantisce la consistenza stretta dell'ERM implicherebbe anche che

$$R_{emp}(\alpha_l) + \sqrt{\frac{H_{ann}^{\Lambda}(2l) - \ln(\eta/4)}{l}} + \frac{1}{l} \rightarrow R_{emp}(\alpha_l) \Rightarrow R(\alpha_l) \leq R_{emp}(\alpha_l)$$

perché se $\lim_{l\to\infty} \frac{\ln H_{ann}^{\Lambda}(2l)}{l} = 0$ anche η tende a zero ed il numeratore di quanto sopra è più lento nel tendere a meno infinito del denominatore. Per la seconda diseguaglianza avremo invece che

$$\sqrt{\frac{H_{ann}^{\Lambda}(2l) - \ln(\eta/4)}{l}} + \frac{1}{l} - \sqrt{\frac{-\ln\eta}{2l}} \rightarrow 0 \Rightarrow R(\alpha_l) \rightarrow T$$

Il primo risultato ci dice che se la dimensione campionaria è sufficientemente grande attraverso la minimizzazione del rischio empirico siamo anche in grado di ottenere una buona generalizzazione perché il fattore relativo alla complessità del set di ipotesi viene meno. Questa buona generalizzazione è a sua volta confermata dal fatto che il rischio effettivo che otteniamo converge al minimo assoluto T.

3.1.6 Funzione di crescita e vincoli costruibili

Nel capitolo precedente abbiamo generalizzato il nostro ragionamento introducendo una nuova modalità per misurare il grado di complessità di un set di ipotesi così da poter utilizzare la diseguaglianza di Hoeffding anche nel caso in cui la cardinalità del insieme S fosse infinita.

Con passaggi analoghi a quelli visti nel caso $|S| = N$ abbiamo definito due disequazioni in grado di descrivere interamente la capacità di generalizzazione definita dalla funzione

$$Q(z, \alpha_l) = \arg\min R_{emp}(\alpha)$$

ossia quella funzione che rende minimo il rischio empirico secondo quanto suggerito dalla intuizione dell'ERM.

Queste due disequazioni definiscono un limite superiore al rischio reale che otterrei utilizzando $Q(z, \alpha_l)$ ossia quant'è al massimo il rischio di generalizzazione che posso avere nonché la distanza tra questo errore generico ed il minimo assoluto ottenibile dall'insieme di funzioni S.

Sicuramente il ragionamento fatto è stato utile per potersi liberare dal vincolo della cardinalità degli insiemi di funzioni ma d'altro canto i risultati ottenuti non sono utilizzabili nell'ambito applicativo.

Il problema risiede proprio nella nuova misura di capacità di S perché in sostanza non esiste una regola che ci permetta di calcolare $H_{ann}^{\Lambda}(2l)$ per un qualsiasi set di ipotesi.

Se prima con la cardinalità bastava misurare il numero di elementi che appartengono ad S ora dovremmo calcolare una funzione del numero massimo di punti frantumabili dalle soluzioni contenute in S.

Non solo, oltre al fatto di non essere facilmente quantificabile per una generica S la grandezza $H_{ann}^{\Lambda}(2l)$ è stata calcolata come

$$H_{ann}^{\Lambda}(l) = ln\left[E_Z\left(N^{\Lambda}(z_1, \dots, z_l)\right)\right]$$

ove l'avere l'expected di $N^{\Lambda}(z_1, \dots, z_l)$ rispetto a Z implica il dover conoscere le probabilità congiunte $P(z_1, \dots, z_l)$ e quindi la distribuzione di Z.

Quando abbiamo descritto il problema dell'apprendere tramite esempi abbiamo specificato che i dati campionari sono assunti come manifestazioni i.i.d. da una variabile aleatoria la cui distribuzione non è nota e non lo sarà mai.

L'unica condizione che poniamo è che le osservazioni campionarie siano tutte estratte dalla stessa variabile aleatoria in maniera indipendente ma non imponiamo alcuna ipotesi distributiva su Z.

In definita per poter davvero applicare nella pratica i risultati teorici qui presentati abbiamo bisogno di definire delle diseguaglianze che descrivano la capacità di generalizzazione di una data funzione

- indipendentemente dalla distribuzione della Z posto che valga l'ipotesi i.i.d. e che
- adotti una misura di capacità del set di ipotesi facilmente definibile

così facendo tutte le grandezze a destra delle due diseguaglianze fondamentali (rischio empirico e capacità del set) sarebbero calcolabili.

A questo fine dovremmo introdurre due nuovi concetti fondamentali nell'ambito della teoria dell'apprendimento statistico: la funzione di crescite e la dimensione di Vapnik e Chervonenkis.

Definiamo **funzione di crescita** sul set di ipotesi Λ il più alto numero di vettori definiti nell'input space Z frantumabili da una qualche funzione di Λ

$$G^{\Lambda}(l) = \ln\left(\sup_{z_1,\dots,z_l \in Z} N^{\Lambda}(x_1, \dots, x_l)\right)$$

quindi la growth function ci dice il più alto numero di vettori l per i quali esiste in Λ almeno una funzione in grado di ottenere tutte le possibili dicotomie (che ricordiamo 2^l).

Va sempre ricordato che qui stiamo considerando una proprietà generica del set di funzioni Λ quindi i vettori $z_1, \dots, z_l$ non sono da intendersi come osservazioni campionarie quanto più come punti generici definiti nello spazio di input.

La growth function ha quindi un evidente legame con le misure di capacità introdotte precedentemente, in particolare avevamo

$$H^{\Lambda}(l) = E_Z\left[\ln\left(N^{\Lambda}(z_1, \dots, z_l)\right)\right]$$

$$H^{\Lambda}_{ann}(l) = ln\left[E_Z\left(N^{\Lambda}(z_1, \dots, z_l)\right)\right]$$

ma evidentemente la $G^{\Lambda}(l)$ è maggiore della $H^{\Lambda}_{ann}(l)$ visto che quest'ultima fa una media rispetto a Z mentre la growth function prende il valore più alto.

Otteniamo

$$H^{\Lambda}(l) \leq H^{\Lambda}_{ann}(l) \leq G^{\Lambda}(l)$$

Riprendendo la prima delle equazioni fondamentali possiamo dire quindi che

$$P\{\Delta|\alpha_{worst}| > \varepsilon\} < 4e^{\left[\frac{H_{ann}^{\Lambda}(2l)}{l} - \left(\varepsilon - \frac{1}{l}\right)^2\right]l} \leq 4e^{\left[\frac{G^{\Lambda}(2l)}{l} - \left(\varepsilon - \frac{1}{l}\right)^2\right]l}$$

perché tutta la parte a destra della diseguaglianza resta costante salvo la sostituzione di $G^{\Lambda}(l)$ in $H_{ann}^{\Lambda}(2l)$.

Proseguendo con il solito ragionamento avremmo che

$$ERM\ strettamente\ consistente\ se\ \lim_{l\to\infty}\frac{G^{\Lambda}(l)}{l} = 0$$

Si osservi come la $G^{\Lambda}(l)$ non dipenda dalla conoscenza della distribuzione di Z visto che non viene fatto il valore atteso quindi la growth function sembra essere una buona candidata a sostituire $H_{ann}^{\Lambda}(l)$ come misura di capacita di un set di ipotesi.

Dobbiamo solo verificare che sia anche facilmente calcolabile per un generico insieme S, per farlo ci appoggeremo al seguente teorema. Anche la dimostrazione di questo risultato è molto lunga e complessa quindi ci limiteremo a dare l'intuizione su cosa significhi questo teorema rispetto alle equazioni fondamentali che ci interessano.

Enunciato:

Si dimostra che la $G^{\Lambda}(l)$ di un set di funzioni indicatrici $S = \{Q(z,\alpha): \alpha \in \Lambda\}$ è pari a

$$G^{\Lambda}(l) = l\ ln\,2$$

oppure limitata superiormente dalla diseguaglianza

$$G^{\Lambda}(l)\begin{cases} = l\ ln\,2 & se\ l \leq h \\ \leq h\left(1 + ln\frac{h}{l}\right) & se\ l > h \end{cases}$$

con h pari al più grande intero per cui $G^{\Lambda}(l) = h\ ln\,2$

Quindi le funzioni di crescita si organizzano in due insiemi

- Lineari con pendenza positiva pari a ln 2 (illimitate al crescere di l)
- Superiormente limitate da un valore h

dove il parametro h è chiamato dimensione di Vapnik e Chervonenkis ed è in definitiva ciò che caratterizza la capacità di un set di funzioni.

Dalla definizione di funzione di crescita superiormente limitata possiamo definire h come il più alto numero di vettori $z_1, \dots, z_h$ che sono classificabili in tutti i 2^h modi da almeno una funzione di $S = \{Q(z, \alpha): \alpha \in \Lambda\}$.

E' quindi evidente che se $l \leq h$ nel caso di funzioni superiormente limitate avremo $G^{\Lambda}(l) = l \ln 2 = \ln 2^l$ perché se siamo in grado di frantumare h punti allora ogni insieme di punti inferiori ad h è frantumabile.

Se invece la funzione di crescita è di tipo lineare allora la $G^{\Lambda}(l)$ riesce a ottenere tutte le 2^l possibili dicotomie per ogni possibile l quindi $h = \infty$.

Ricollegandoci alla affermazione

$$ERM\ strettamente\ consistente\ se\ \lim_{l \to \infty} \frac{G^{\Lambda}(l)}{l} = 0$$

possiamo quindi osservare che questa convergenza a zero di $G^{\Lambda}(l)/l$ può avvenire solamente quando la funzione di crescita è superiormente limitata.

Infatti nel caso di dimensione VC infinita avremo

$$\frac{G^{\Lambda}(l)}{l} = \ln 2$$

che evidentemente non può convergere allo zero. Nel caso di funzioni superiormente limitate avremo invece

$$\frac{G^{\Lambda}(l)}{l} = \frac{h}{l}\left(1 + ln\frac{h}{l}\right)$$

ed essendo h in questo caso un numero reale evidentemente la scrittura di cui sopra converge a zero essendo h/l più veloce di $\ln(h/l)$.

Quindi potremmo riscrivere in definitiva che:

$ERM\ strettamente\ consistente\ se\ la\ dimensione\ di\ Vapnik\ e\ Chervonenkis$ è $finita$

Per rafforzare questo risultato dimostriamo come un set di funzioni S con $h = \infty$ non può convergere in maniera uniforme.

Supponiamo di considerare un insieme di funzioni $S = \{Q(z, \alpha): \alpha \in \Lambda\}$ con $d_{VC} = \infty$ allora sappiamo che questo S è in grado di frantumare qualsiasi set di punti $z_1, \dots, z_l$ quale che sia l ossia

$$\sup_{z_1,\dots,z_l \in Z} N^{\Lambda}(z_1, \dots, z_l) = 2^l \quad \forall l$$

fissato un l qualsiasi scegliamo un valore di ε piccolo a piacere ed indichiamo con n un intero tale per cui

$$n > \frac{l}{\varepsilon}$$

essendo la dimensione VC infinita il set S è in grado di frantumare anche $z_1, \dots, z_n$ in tutti i 2^n modi.

Definendo sinteticamente con Z^n il campione $z_1, \dots, z_n$ assumiamo che tutta la probabilità dello spazio campionario Z sia concentrata su Z^n in modo tale per cui

$$P(z_i) = \frac{1}{n} \quad \forall i = 1, \dots, n$$

In sostanza stiamo dicendo che un valore osservato z_i proviene necessariamente da Z^n (perché è lì che si concentra tutta la probabilità) ed ha probabilità di estrazione pari a $1/n$ quindi è come se lo spazio campionario Z sia collassato nello spazio definito dalle sole combinazioni dei punti di Z^n.

All'interno di Z^n possiamo definire Z^l che sarà necessariamente un sottoinsieme del primo e quindi

$$Z^n = Z^l \cup Z^0$$

essendo

$$N^{\Lambda}(z_1, \dots, z_n) = 2^n$$

Esiste in S una funzione α^* che assume valori

- 1 per tutti i punti appartenenti a Z^0 che rappresentano una quota $\frac{n-l}{n}$ in Z^n
- 0 per tutti i punti appartenenti a Z^l che rappresentano una quota $\frac{l}{n}$ in Z^n

Quindi avremo un errore empirico nullo ($R_{emp}(\alpha^*) = 0$) ed un errore atteso almeno pari alla quota di punti di Z^0 in Z^n ($R(\alpha^*) \geq (n-l)/l$) e si osservi che

$$n > \frac{l}{\varepsilon} \Rightarrow \frac{n-l}{n} > 1 - \varepsilon \Rightarrow R(\alpha^*) > 1 - \varepsilon$$

quindi con probabilità pari a 1 avviene che

$$\sup_{\alpha \in \Lambda} R(\alpha^*) > 1 - \varepsilon \Rightarrow \sup_{\alpha \in \Lambda} \left| R(\alpha^*) - R_{emp}(\alpha^*) \right| = \Delta|\alpha_{worst}| > 1 - \varepsilon$$

E' quindi certo che $\Delta|\alpha_{worst}| > 1-\varepsilon$ ed essendo ε il grado di precisione per valori operativi (tra il 5 e 1%) vale che $1-\varepsilon > \varepsilon$ quindi

$$P[\Delta|\alpha_{worst}| < \varepsilon] = 0$$

A differenza di quanto visto nei capitoli precedenti questo è un risultato esatto e non asintoticamente vero.

L'avere una dimensione VC finita è quindi condizione necessaria perché l'ERM sia consistente. D'altro canto dalla diseguaglianza di Hoeffding abbiamo visto che la consistenza è ottenuta solo quando $G^{\Lambda}(l)/l$ converge a zero e ciò accade solo per funzioni di crescita superiormente limitate ossia con dimensione VC finita.

L'avere una d_{VC} finita è quindi condizione necessaria e sufficiente alla consistenza dell'ERM. Assumendo che ciò avvenga possiamo ricavare le diseguaglianze fondamentali sostituendo alla $G^{\Lambda}(l)$ la sua definizione nel caso di funzione limitata

$$G^{\Lambda}(l) \leq h\left(1 + ln\frac{h}{l}\right)$$

dentro

$$P\{\Delta|\alpha_{worst}| > \varepsilon\} < 4e^{\left[\frac{G^{\Lambda}(2l)}{l} - \left(\varepsilon - \frac{1}{l}\right)^2\right]l} \leq 4e^{h\left(1+ln\frac{h}{l}\right) - {\varepsilon_0}^2 l}$$

Con passaggi analoghi a quelli visti nel caso a cardinalità finita e non otteniamo le seguenti diseguaglianze fondamentali

$$R(\alpha_l) \leq R_{emp}(\alpha_l) + \frac{\varepsilon(l)}{2}\left(1 + \sqrt{1 + 4\frac{R_{emp}(\alpha_l)}{\varepsilon(l)}}\right) = R_{emp}(\alpha_l) + \Omega(h, l, \eta)$$

$$R(\alpha_l) - T = \Delta(\alpha_l) \leq \sqrt{\frac{-\ln\eta}{2l}} + \frac{\varepsilon(l)}{2}\left(1 + \sqrt{1 + 4\frac{R_{emp}(\alpha_l)}{\varepsilon(l)}}\right) = \sqrt{\frac{-\ln\eta}{2l}} + \Omega(h, l, \eta)$$

con

$$\varepsilon(l) = 4\frac{h\left(\ln\left(\frac{2l}{h}\right) + 1\right) - \ln\frac{\eta}{4}}{l}$$

$$\eta = 4e^{h\left(1+ln\frac{h}{l}\right) - {\varepsilon_0}^2 l}$$

Come sempre il risultato è analogo a quelli visti precedentemente, ancora una volta osserviamo che al divergere di l a più infinito la componente che spiega la capacità del set

di ipotesi si azzera lasciando il solo rischio empirico come limite superiore al valore di rischio effettivo che a sua volta tende al minimo assoluto ottenibile su quel set di ipotesi.

La differenza con tutti i casi fin'ora visti è che la grandezza $\Omega(h, l, \eta)$ ora è calcolabile e quindi a seconda di quale errore empirico otteniamo potremmo definire il limite in probabilità del rischio effettivo così come la distanza rispetto al valore di ottimo assoluto.

Vediamo la cosa con un esempio: si supponga di voler utilizzare come classificatore lineare l'insieme degli iperpiani di separazione. Si dimostra che in generale il massimo numero di punti frantumabili da questo insieme funzionale è $n + 1$ ove n rappresenta il numero di dimensioni su cui è definito lo spazio di input.

Dalla definizione di dimensione VC ricaviamo quindi che quella degli iperpiani di separazione è appunto $n + 1$.

Si assuma di avere i seguenti dati

- Input space definito in R^2 quindi $h = 3$
- Dimensione campionaria disponibile $l = 100$
- Misura di precisione scelta $\varepsilon = 0.05$ quindi $\varepsilon_0 = \varepsilon - \frac{1}{l} = 0.04$

Otteniamo che

$$\eta = 4e^{h\left(1+ln\frac{h}{l}\right)-\varepsilon_0{}^2 l} = 4e^{3\left(1+ln\frac{3}{100}\right)-0.04^2 100} = 0.001848502$$

$$\varepsilon(100) = 4\frac{3\left(\ln\left(\frac{200}{3}\right)+1\right)-\ln\frac{0.001848502}{4}}{100} = 0.931151571$$

e quindi

$$R(\alpha_l) \leq R_{emp}(\alpha_l) + 0.465757855\left(1 + \sqrt{1 + 4\frac{R_{emp}(\alpha_l)}{0.931151571}}\right)$$

$$\Delta(\alpha_l) \leq 0.177389116 + 0.465757855\left(1 + \sqrt{1 + 4\frac{R_{emp}(\alpha_l)}{0.931151571}}\right)$$

con probabilità rispettivamente

$$1 - \eta = 0.998151498$$

$$1 - 2\eta = 0.996302996$$

3.2 Principio di Minimizzazione del Rischio Strutturale

Data la lunghezza e complessità dei ragionamenti fatti nei capitoli precedenti è bene presentare una sorta di riassunto dei passaggi logici fin'ora fatti. L'aver presente la logica di fondo fin'ora seguita sarà fondamentale per capire il problema che verrà affrontato in questo capitolo.

I nostri ragionamenti sono iniziati chiedendosi cosa significasse apprendere e quando effettivamente si può affermare di aver appreso. Abbiamo visto che nel nostro ambito con apprendere indichiamo la capacità di definire una regola che dia una buona soluzione "in generale" ossia per ogni possibile problema che si può avere senza limitarsi agli esempi utilizzati per l'addestramento (ossia per definire la regola).

Potevamo quindi definire l'errore come l'expected definito su tutto lo spazio campionario Z di una qualche funzione di errore che dipende dalla soluzione α scelta.

Per calcolare questo expected è però necessario conoscere la distribuzione di Z e nella pratica ciò non avviene mai. Quindi apprendere si traduce in un problema di ottimizzazione ove la quantità da minimizzare non è calcolabile.

Si è però osservato che l'errore empirico ha una definizione che richiama quella dell'errore vero limitata però ai soli esempi campionari invece che all'intero input space. Se quindi esiste una relazione tra rischio reale ed empirico tramite una minimizzazione di quest'ultimo saremmo in grado di ottenere una buona performance di classificazione in generale e quindi di apprendere.

Questa è l'intuizione che sta alla base dell'ERM, il problema è però definire appunto questa relazione tra i due tipi di errori. Dopo una serie di ragionamenti e teoremi siamo giunti alla conclusione che l'errore reale ottenuto con la funzione che rende minimo il rischio empirico e superiormente limitato in probabilità da una grandezza definita come la somma tra il rischio empirico e una quantità che dipende dalla complessità del set di ipotesi utilizzate.

Abbiamo inoltre definito una seconda equazione che ci dice quanto l'errore effettivo ottenuto con la nostra funzione di ottimo sia vicino in probabilità al minimo assoluto che si potrebbe avere sul set di funzioni che abbiamo scelto come possibili soluzioni.

Tutti i ragionamenti che abbiamo fin'ora fatto sono serviti a dirci quanto la soluzione di ottimo che troviamo in fase di addestramento è buona **ma operativamente non abbiamo aggiunto nulla a come si trova la soluzione.**

Infatti potevamo comunque ricavare lo stesso risultato operativamente con un qualche algoritmo in grado di ottenere un rischio empirico estremamente contenuto se non addirittura nullo.

L'ERM alla fine afferma semplicemente che per ottenere una buona performance in generale bisogna minimizzare il rischio empirico. Tutti i ragionamenti fatti non fanno altro che giustificare questa affermazione.

Vi è però una critica fondamentale al criterio dell'ERM che è rimasta sottointesa lungo questi capitoli ossia che l'ERM funziona solo quando la dimensione campionaria l è molto elevata.

Infatti si ricordi come quasi tutti i teoremi e le affermazioni fondamentali definivano una qualche grandezza che rapportata ad l doveva convergere a zero per $l \to \infty$ affinché l'ERM fosse consistente. Persino quando si afferma che la dimensione VC deve essere finita questo si giustificava col fatto di riuscire ad ottenere $G^{\Lambda}(l)/l \to 0$ quando l converge a più infinito.

Nella realtà però la dimensione campionaria non è infinita e quindi è bene chiedersi se un criterio che guarda solo alla minimizzazione del rischio empirico sia l'ideale nel ricercare una funzione ottimale.

Per rispondere si riprenda l'esempio numerico, posto di voler utilizzare la famiglia degli iperpiani di separazione avevamo i seguenti dati:

- Input space definito in R^2 quindi $h = 3$
- Dimensione campionaria disponibile $l = 100$
- Misura di precisione scelta $\varepsilon = 0.05$ quindi $\varepsilon_0 = \varepsilon - \frac{1}{l} = 0.04$

Otteniamo che

$$\eta = 0.001848502$$

$$\varepsilon(100) = 0.931151571$$

e quindi

$$R(\alpha_l) \leq 0.465757855\left(1 + \sqrt{1 + 4\frac{R_{emp}(\alpha_l)}{0.931151571}}\right)$$

con probabilità

$$1 - \eta = 0.998151498$$

Una dimensione campionaria l = 100 non è dopotutto così elevata ed infatti se anche ponessimo il rischio empirico pari a zero otterremmo

$$R(\alpha_l) \leq 0{,}93151571$$

con una probabilità di circa 0,998.

Con una dimensione campionaria di l = 100.000 otteniamo invece

$$R(\alpha_l) \leq 0{,}012578961$$

con una probabilità approssimata a 1.

La funzione che rende minimo il rischio empirico da anche una buona performance out of sample solo quando la dimensione campionaria è sufficientemente elevata, altrimenti si incorre in quello che nell'ambito dell'apprendimento statistico è chiamato overfitting.

L'overfitting può essere definito come la situazione in cui il modello è sovra adattato agli esempi utilizzati in fase di addestramento e quindi non ha definito una regola che gli permetta di eseguire una buona classificazione in generale.

Riprendendo la nostra metafora l'overfitting è lo scenario in cui lo studente ha sostanzialmente memorizzato tutti gli esempi su cui si è esercitato ma non ha definito alcuna regola generale che gli permetta di risolvere un nuovo problema.

Va altresì notato che i risultati teorici ottenuti nei capitoli precedenti non vengono meno nel caso in cui l sia piccolo. Infatti le diseguaglianze fondamentali

$$R(\alpha_l) \leq R_{emp}(\alpha_l) + \Omega(h, l, \eta)$$

$$\Delta(\alpha_l) \leq \sqrt{\frac{-\ln \eta}{2l}} + \Omega(h, l, \eta)$$

sono vere indipendentemente da che valore assume l.

Quello che stiamo qui dicendo è che se l è contenuto la soluzione ottimale non è quella che rende minimo $R_{emp}(\alpha_l)$ perché rimane un fattore $\Omega(h, l, \eta)$ che altrimenti si azzererebbe per $l \to \infty$.

A conferma del fatto che tutto il lavoro fin'ora svolto non è stato inutile osserviamo che se le due diseguaglianze fondamentali sono vere queste ci suggeriscono implicitamente un metodo alternativo alla ricerca della soluzione di ottimo.

Infatti, se la dimensione campionaria è contenuta allora è ragionevole affermare che la funzione di ottimo sarà quella che all'interno dello spazio funzionale S scelto restituisce il minor valore della somma delle due componenti a destra della diseguaglianza che definisce il vincolo di $R(\alpha_l)$ ossia

$$\alpha^* = \arg\left[\min_{\alpha\in\Lambda}\left(R_{emp}(\alpha_l) + \Omega(h,l,\eta)\right)\right]$$

In questo modo dovremmo essere in grado di definire un vincolo al rischio effettivo che sia inferiore a quello ottenibile con l'ERM.

Dobbiamo quindi chiederci se esista un modo con cui possiamo controllare il parametro $\Omega(h,l,\eta)$ per poterlo considerare attivamente nella minimizzazione.

A tal proposito si riprenda l'esempio numerico proposto poc'anzi, si consideri di non modificare alcun parametro tranne la dimensione di VC. I risultati diventano

- $h = 3 \Rightarrow R(\alpha_l) \leq 0.93151571$ $con\ probabilità$ 0.998151
- $h = 2 \Rightarrow R(\alpha_l) \leq 0.68777550$ $con\ probabilità$ 0.989926
- $h = 1 \Rightarrow R(\alpha_l) \leq 0.40253950$ $con\ probabilità$ 0.907345

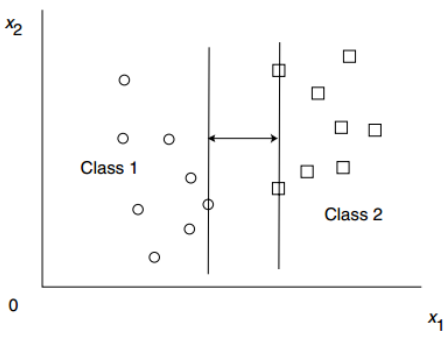
va notato che avendo posto il rischio empirico pari a zero il vincolo superiore di $R(\alpha_l)$ è interamente definito dalla grandezza $\Omega(h,l,\eta)$ che quindi ha una relazione inversa rispetto alla dimensione VC.

Più la dimensione di VC è bassa e minore è il valore ottenuto su $\Omega(h,l,\eta)$ a parità di ogni altra condizione quindi la soluzione che minimizza il vincolo superiore del rischio effettivo deve tener conto anche del grado di complessità del set di funzioni scelto.

Come possiamo però controllare quest'ultima componente?

Dall'esempio numerico di cui sopra ricordiamo che il set di ipotesi scelto era quello degli iperpiani di separazione la cui dimensione VC è pari a $n + 1$ ove n è la dimensione dello spazio di input.

Sempre nel nostro esempio avevamo assunto di avere dati campionari definiti sul piano cartesiano e da li abbiamo definito la h del nostro set di ipotesi pari a 3.



In realtà nulla ci vieta di poter utilizzare come classificatore un valore soglia (definito quindi in R^1) pur avendo dati definiti sulle due dimensioni. Semplicemente classificherò le osservazioni a seconda di come si

posizionano rispetto ad un valore soglia (che farà riferimento alla prima o seconda coordinata).

Generalizzando a spazi campionari di n dimensioni se stiamo adottando come set di ipotesi l'insieme degli iperpiani di separazione nulla mi costringe a dover scegliere la soluzione nel sottoinsieme degli iperpiani definiti su n dimensioni e che quindi hanno $h = n + 1$.

Potrei scegliere come soluzione un qualsiasi iperpiano la cui dimensione varia da 1 a n e che quindi ha una dimensione VC che varia tra 2 e $n + 1$.

L'esempio di cui sopra è utile per farci comprendere come certi set di ipotesi S abbiano al proprio interno una sorta struttura nidificata definita sulla base del grado di complessità a cui si fa riferimento.

Nel nostro esempio se S è l'insieme degli iperpiani di separazione su n dimensioni lo stesso S è scomponibile in una struttura nidificata del tipo

$$S_1 \subset S_2 \subset \cdots \subset S_n$$

che equivalgono a

$$soglia \subset retta \subset piano \subset iperpiano \subset \cdots$$

Ogni sotto insieme S_i è caratterizzato dall'avere un grado di complessità superiore a quelli precedenti ed inferiore a quello dei livelli che lo contengono.

Definendo con h_i la VC dimension dell'i-esimo livello della struttura per i motivi di cui sopra dovrà valere che

$$h_1 \leq h_2 \leq \cdots \leq h_n$$

Una struttura di questo tipo nel set d ipotesi permetterebbe di controllare il parametro $\Omega(h, l, \eta)$ tramite la scelta di un opportuno sottoinsieme S_i in cui individuare la soluzione di ottimo.

Il problema che però sorge da una tale frammentazione del set di ipotesi è che scegliendo un certo sottoinsieme S_i non si trovi una soluzione ottimale. E' quindi necessario garantire che per ogni sottoinsieme S_i di S sia possibile ottenere almeno asintoticamente una soluzione che sia ottimale limitatamente alle funzioni che appartengono a questo subset.

Ricordando che la consistenza dell'ERM risulta garantita solo a patto di avere una VC dimension finita una condizione necessaria perché ciò valga per tutti i sottoinsiemi S_i è che h_i sia finito per ogni i.

Infine considerando che con $Q(z,\alpha)$ s'intende una funzione di perdita è ragionevole pensare che questa non possa assumere qualsiasi valore ma che sia limitata come

$$0 \leq Q(z,\alpha) \leq B$$

ne segue che ad ogni sottoinsieme S_i corrisponderà un B_i che avranno una struttura simile a quella vista con le dimensioni VC

$$B_1 \leq B_2 \leq \cdots \leq B_n$$

L'ultima condizione è per motivi molto tecnici ma ci servirà per superare un passaggio in un una dimostrazione di un teorema fondamentale.

Definendo con

$$S^* = \bigcup_k S_k$$

assumiamo che questo sia denso su tutto S ossia che esista un $Q(z,\alpha^*) \in S^*$ tale per cui

$$\int |Q(z,\alpha) - Q(z,\alpha^*)|\, dF(z) < \delta$$

ove δ è un valore piccolo a piacere.

Abbiamo ora tutti gli elementi per poter definire il concetto di struttura ammissibile di un set di ipotesi che sarà fondamentale nel principio di ricerca della funzione ottimale nel caso in cui non si disponga di una dimensione campionaria sufficiente.

Definiamo struttura ammissibile di $S = \{Q(z,\alpha): \alpha \in \Lambda\}$ la

$$S_1 \subset S_2 \subset S_3 \subset \cdots$$

per cui sono vere le seguenti condizioni

1. $h_1 \leq h_2 \leq h_3 \leq \cdots$ ove ogni h_i è finito
2. $B_1 \leq B_2 \leq B_3 \leq \cdots$
3. S^* è denso su tutto S

Questa nozione ci servirà nei prossimi capitoli in cui dimostreremo due teoremi fondamentali i quali sostanzialmente affermano che:

1. se S ha una struttura ammissibile allora per ogni sottoinsieme il rischio effettivo ottenuto con la funzione che rende minimo il rischio empirico converge all'ottimo con un tasso di convergenza non infinito

2. l'SRM è consistente (come l'ERM) indipendentemente dalla distribuzione di Z e converge all'ottimo assoluto con una velocità elevata.

3.2.1 Alcune definizioni

Nel precedente capitolo abbiamo introdotto l'SRM come un principio che a differenza della mera minimizzazione del rischio empirico cerca la soluzione di ottimo nella funzione $Q(z, \alpha^*)$ tale per cui

$$\alpha^* = \arg\left[\min_{\alpha\in\Lambda}\left(R_{emp}(\alpha_l) + \Omega(h, l, \eta)\right)\right]$$

Questo cambio di approccio si giustificava dal fatto che l'ERM consente di ottenere buoni risultati out of sample solo quando la dimensione campionaria tende a più infinito.

Dopo aver visto che il secondo addendo $\Omega(h, l, \eta)$ è in parte governato dalla dimensione VC del set di ipotesi scelto si è intuito come uno stesso insieme S possa essere visto come la nidificazione di tanti sottoinsiemi definiti da una diversa dimensione VC.

Nella ricerca del α^* ottimale è quindi possibile governare il fattore h (ossia $\Omega(h, l, \eta)$) con un'opportuna scelta del subset di ipotesi in cui cercare la soluzione finale.

Per una serie di motivazioni tecniche abbiamo poi dato una definizione di struttura ottimale di un set di ipotesi.

Prima di passare ai due teoremi che ci giustificheranno l'utilizzo dell'SRM è bene dare una serie di definizioni che ci permetteranno di snellire la notazione che utilizzeremo in seguito.

Innanzitutto va specificato che nei capitoli riguardanti l'ERM abbiamo implicitamente assunto che la funzione di perdita $Q(z, \alpha)$ non avesse alcun vincolo sui valori che può assumere. Più in generale si dimostra che se $A \leq Q(z, \alpha) \leq B$ la diseguaglianza di Hoeffding diviene

$$P\{\Delta|\alpha_{worst}| > \varepsilon\} \leq e^{-2\frac{\varepsilon^2 l}{(B-A)^2}}$$

e quindi

$$P\{\Delta|\alpha_{worst}| > \varepsilon\} \leq N\, e^{-2\frac{\varepsilon^2 l}{(B-A)^2}}$$

Ponendo $A = 0$ come specificato nella definizione di struttura ammissibile otteniamo

- $P\{\Delta|\alpha_{worst}| > \varepsilon\} \leq N\, e^{-2\frac{\varepsilon^2 l}{B^2}}$ nel caso a cardinalità finita

- $P\{\Delta|\alpha_{worst}| > \varepsilon\} < 4e^{\left[\frac{H^{\Lambda}_{ann}(2l)}{l} - \left(\frac{\varepsilon_0}{B}\right)^2\right]l}$ nel caso a cardinalità infinita

Quindi la diseguaglianza fondamentale diviene

$$R(\alpha_l) \leq R_{emp}(\alpha_l) + B\frac{\varepsilon(l)}{2}\left(1 + \sqrt{1 + 4\frac{R_{emp}(\alpha_l)}{B\varepsilon(l)}}\right)$$

Definiamo inoltre le seguenti funzioni di riferimento

1. $Q(z, \alpha_l^k) = \arg\left[\min_{\alpha \in S_k} R_{emp}(\alpha)\right]$
   la funzione che minimizza l'errore empirico all'interno di uno specifico subset S_k

2. $Q(z, \alpha_0^k) = \arg\left[\min_{\alpha \in S_k} R(\alpha)\right]$
   la funzione che minimizza l'errore atteso all'interno di uno specifico subset S_k

3. $Q(z, \alpha_0) = \arg[\min_{\alpha \in S} R(\alpha)] = R(\alpha_0)$
   la funzione che minimizza l'errore atteso considerando l'intero set di ipotesi S

4. $n(l)$
   regola (che supponiamo esista) a priori e che ci dica quale S_k scegliere in base alla numerosità campionaria l

Infine affermiamo che una serie di variabili $\varphi_l, l = 1,2, \ldots$ converge ad un valore φ_0 con tasso di convergenza asintotica $V(l)$ se esiste una costante C tale per cui

$$V(l)|\varphi_l - \varphi_0| \xrightarrow{P} C \;\; per\; l \to \infty$$

E per ultimo diamo un veloce accenno al lemma di Borel-Cantelli nella sua versione relativa alla probabilità

Dato uno spazio di misura di probabilità (Ω, A, P) e assegnata una successione di eventi $\{E_n\}_{n \in N}$ se vale

$$\sum_{n=0}^{\infty} P(E_n) \in R \Rightarrow P\left(\limsup_{n \to \infty} E_n\right) = 0$$

ove $\mathrm{limsup}(E_n)$ è l'evento limite superiore per n che diverge ad infinito

3.2.2 Primo Teorema

Si consideri un set di ipotesi S che vanta una struttura ammissibile, considerando un generico sottoinsieme S_k possiamo definire la seconda diseguaglianza fondamentale come

$$\Delta(\alpha_l^k) = R(\alpha_l^k) - R(\alpha_0^k) \leq B_k \left(\sqrt{\frac{-\ln \eta}{2l}} + \sqrt{\frac{h_k \left(\ln \frac{2l}{h_k} + 1 \right) + 2\ln(2l)}{l}} \right) = \delta$$

ove $R(\alpha_l^k)$ è il valore di rischio atteso ottenibile dalla funzione che in S_k rende minimo il rischio empirico mentre $R(\alpha_0^k)$ è il minimo assoluto ottenibile su S_k.

Sommando e sottraendo a sinistra della diseguaglianza il minimo assoluto su tutto S otteniamo

$$R(\alpha_l^k) - R(\alpha_0^k) + [R(\alpha_0) - R(\alpha_0)] \leq \delta$$

$$R(\alpha_l^k) - R(\alpha_0) \leq [R(\alpha_0^k) - R(\alpha_0)] + \delta = r_k + \delta$$

ove r_k rappresenta il tasso di convergenza ossia quanto l'ottimo ottenibile sul sottoinsieme S_k è vicino all'ottimo su tutto S.

Avendo assunto che S^* è denso su tutto S sappiamo che esiste una funzione $Q(z, \alpha^*) \in S^*$ tale per cui

$$\int |Q(z, \alpha) - Q(z, \alpha^*)| \, dF(z) < \theta$$

nostro caso avremo che

$$r_k = R(\alpha_0^k) - R(\alpha_0) = \int Q(z, \alpha_0^k) dF(z) - \int Q(z, \alpha_0) dF(z) =$$

$$= \int \left| Q(z, \alpha_0) - Q(z, \alpha_0^k) \right| dF(z) < \theta$$

con θ piccolo a piacere.

Ne segue che

$$\lim_{l \to \infty} r_k = 0$$

quindi affinché $R(\alpha_l^k) - R(\alpha_0)$ tenda a zero al divergere di l è necessario che anche la δ vada ad azzerarsi.

Si osservi che

$$\delta = B_k \left(\sqrt{\frac{-\ln \eta}{2l}} + \sqrt{\frac{h_k \left(\ln \frac{2l}{h_k} + 1 \right) + 2 \ln(2l)}{l}} \right)$$

può essere riscritto come

$$\delta = B_k \sqrt{\frac{-\ln \eta}{2l}} + \sqrt{\frac{{B_k}^2 h_k \left(\ln \frac{2l}{h_k} + 1 \right)}{l} + \frac{2{B_k}^2 \ln(2l)}{l}}$$

e per i motivi visti nei capitoli sull'ERM sappiamo che

$$\lim_{l \to \infty} B_k \sqrt{\frac{-\ln \eta}{2l}} = 0 \ \ e \ \ \lim_{l \to \infty} B_k \frac{2{B_k}^2 \ln(2l)}{l} = 0$$

Quindi affinché δ converga a zero serve che

$$\lim_{l \to \infty} \frac{{B_k}^2 h_k \left(\ln \frac{2l}{h_k} + 1 \right)}{l} = 0$$

che possiamo semplificare in

$$\lim_{l \to \infty} \frac{{B_k}^2 h_k \ln(l)}{l} = 0$$

Chiamiamo ora con $V(l)$ tutto ciò che sta a destra della disequazione fondamentale definita sopra

$$R(\alpha_l^k) - R(\alpha_0) \leq r_k + \delta = V(l)$$

Invertendo la $V(l)$ otteniamo con probabilità strettamente inferiore a $1 - 2/l^2$

$$V^{-1}(l)[R(\alpha_l^k) - R(\alpha_0)] \leq 1$$

ossia

$$P(V^{-1}(l)[R(\alpha_l^k) - R(\alpha_0)] > 1) < \frac{2}{l^2}$$

La diseguaglianza di cui sopra (così come la sua probabilità) dipende dal valore assunto dalla l.

E' abbastanza intuitivo il fatto che una serie del tipo

$$\sum_{l=1}^{\infty} \frac{2}{l^2}$$

converga ad un valore finito e quindi

$$\sum_{l=1}^{\infty} \frac{2}{l^2} = l_0 + \sum_{l=l_0+1}^{\infty} \frac{2}{l^2} < \infty \; con \; l_0 < l$$

Riprendendo la diseguaglianza di cui sopra otteniamo

$$\sum_{l=1}^{\infty} P\{V^{-1}(l)[R(\alpha_l^k) - R(\alpha_0)] > 1\} < \sum_{l=1}^{\infty} \frac{2}{l^2} = l_0 + \sum_{l=l_0+1}^{\infty} \frac{2}{l^2} < \infty$$

sfruttando il Teorema di Borel Cantelli e definendo l'evento l-esimo come

$$E_l = \{V^{-1}(l)[R(\alpha_l^k) - R(\alpha_0)] > 1\}$$

otteniamo che

$$P\left(\lim_{l\to\infty} \sup E_l\right) = P\left\{\lim_{l\to\infty} \sup \; (V^{-1}(l)[R(\alpha_l^k) - R(\alpha_0)] > 1)\right\} = 0$$

da cui ricaviamo che

$$P\left\{\lim_{l\to\infty} \sup(V^{-1}(l)[R(\alpha_l^k) - R(\alpha_0)] \leq 1)\right\} = 1$$

Infine introducendo una regola $n(l)$ che ci dica quale subset S^k utilizzare in base alla numerosità campionaria disponibile da quello che abbiamo visto sull'ERM una condizione ragionevole è che

$$\mathrm{n(l)} \overset{\mathrm{l}}{\to} \infty$$

sappiamo infatti che per l che tende ad infinito la regola migliore resta quella fornita dall'ERM ossia minimizzare il rischio empirico. Tale minimizzazione è però ottimale solo quando cerchiamo la soluzione sull'intero insieme funzionale quindi la condizione di cui sopra ci assicura che se l va a più infinito il subset di ricerca sarà in pratica l'intero S.

Possiamo quindi riscrivere l'ultimo risultato come

$$P\left\{\lim_{l\to\infty} \sup\left(V^{-1}(l)\left[R\left(\alpha_l^{\mathrm{n(l)}}\right) - R(\alpha_0)\right] \leq 1\right)\right\} = 1$$

In sostanza stiamo dicendo che per una struttura di S ammissibile la funzione $\alpha_l^{n(l)}$ che rende minimo il rischio empirico sulla base delle sole funzioni di $S_{n(l)}$ determina un rischio atteso $R\left(\alpha_l^{n(l)}\right)$ che converge al minimo assoluto $R(\alpha_0)$ in un tempo certamente finito.

Possiamo quindi formulare il primo teorema del SRM come segue:

Se è vero che

- $\exists\, n(l): n(l) \overset{l}{\to} \infty$
- $\lim_{l\to\infty} \frac{B_{n(l)}{}^2 h_{n(l)} \ln(l)}{l} = 0$

allora l'SRM fornisce una funzione $Q\left(z, \alpha_l^{n(l)}\right)$ che minimizza il rischio empirico su $S_{n(l)}$ e che converge al minimo assoluto $R(\alpha_0)$ con un tasso di convergenza

$$V(l) = r_k + \delta = V(l)$$

certamente finito.

3.2.3 Secondo Teorema

Con questo secondo teorema vogliamo dimostrare che l'SRM converge alla miglior soluzione possibile con probabilità 1 indipendentemente dalla distribuzione di Z e con un tasso di convergenza elevato.

Per semplificare la dimostrazione assumiamo però di considerare solamente i primi l sottoinsiemi $S_1, \dots, S_l$ di S ove l è il numero di osservazioni campionarie. Questo perché un set di ipotesi può essere costituito da infiniti subset nidificati e quindi al fine di eseguire una ricerca dell'ottimo su un insieme finito (come facevamo nel primo capitolo in cui si è considerata la diseguaglianza di Hoeffding) facciamo questa ipotesi.

Visto che ora stiamo considerando l subset di ipotesi vi saranno l diverse funzioni che rendono minimo (all'interno del loro sottoinsieme) il rischio empirico. Quindi avremo

$$Q(z, \alpha_l^k) = \arg\left[\min_{\alpha \in S_k} R_{emp}(\alpha)\right] \; per\ ogni\ k = 1, \dots, l$$

e definiamo con $Q(z, \alpha_l^+)$ quella funzione che all'interno delle l $Q(z, \alpha_l^k)$ garantisce con probabilità $1 - 1/l$ che

$$R_{emp}^+(\alpha_l^+) = \min_{1 \le k \le l} \left(R_{emp}(\alpha_l^k) + B_k \sqrt{\frac{h_k\left(\ln\frac{2l}{h_k} + 1\right) + 2\ln(2l)}{l}} \right)$$

sul mio set di osservazioni.

Vogliamo capire quanto il rischio effettivo che otteniamo con α_l^+ (parametro che rende minimo il rischio empirico $R_{emp}^+(\alpha_l^+)$ sulle l osservazioni) sia distante dal minimo assoluto $R(\alpha_0)$ ottenibile sull'intero set di ipotesi.

$$R(\alpha_l^+) - R(\alpha_0)$$

Si consideri la seguente scomposizione

$$R(\alpha_l^+) - R(\alpha_0) = \left(R(\alpha_l^+) - R_{emp}^+(\alpha_l^+)\right) + \left(R_{emp}^+(\alpha_l^+) - R(\alpha_0)\right) = A + B$$

l'obiettivo è verificare quando le due componenti si azzerano.

Partendo dalla A e considerando l'evento

$$\{A > \varepsilon\} = \left\{R(\alpha_l^+) - R_{emp}^+(\alpha_l^+) > \varepsilon\right\}$$

utilizzando i soliti ragionamenti fatti con la diseguaglianza di Hoeffding otteniamo che

$$P(A > \varepsilon) = P\left(R(\alpha_l^+) - R_{emp}^+(\alpha_l^+) > \varepsilon\right) < \sum_{k=1}^{l} P\left(R(\alpha_l^k) - R_{emp}^+(\alpha_l^k) > \varepsilon\right)$$

Sostituendo in $R_{emp}^+(\alpha_l^k)$ la sua definizione

$$R_{emp}^+(\alpha_l^k) = R_{emp}(\alpha_l^k) + B_k\sqrt{\frac{h_k\left(\ln\frac{2l}{h_k} + 1\right) + 2\ln(2l)}{l}} = R_{emp}(\alpha_l^k) + B_k\sqrt{C_k}$$

otteniamo

$$P(A > \varepsilon) < \sum_{k=1}^{l} P\left(R(\alpha_l^+) - R_{emp}(\alpha_l^k) > \varepsilon + B_k\sqrt{C_k}\right)$$

E dalla diseguaglianza di Hoeffding otteniamo

$$\sum_{k=1}^{l} P\left(R(\alpha_l^+) - R_{emp}(\alpha_l^k) > \varepsilon + B_k\sqrt{C_k}\right) \leq \sum_{k=1}^{l} 2e^{-2l\left(\varepsilon + B_k\sqrt{C_k}\right)^2}$$

Notare infatti che in questo caso non i vincoli superiori non sono sempre uguali perché riferiscono ad un valore $\varepsilon + B_k\sqrt{C_k}$ che dipende dal subset S_k a cui si fa riferimento.

Sviluppando il quadrato avremo

$$\left(\varepsilon + B_k\sqrt{C_k}\right)^2 = \varepsilon^2 + 2\varepsilon B_k\sqrt{C_k} + {B_k}^2 C_k = {B_k}^2\left(\frac{\varepsilon^2}{{B_k}^2} + \frac{2\varepsilon\sqrt{C_k}}{B_k} + C_k\right)$$

che possiamo riscrivere come

$${B_k}^2\left(\frac{\varepsilon^2}{{B_k}^2} + \frac{2\varepsilon\sqrt{C_k}}{B_k} + C_k\right) = {B_k}^2\left(\frac{\varepsilon}{B_k} + \sqrt{C_k}\right)^2$$

e quindi l'argomento dell'esponenziale diventa

$$-2l\left(\varepsilon + B_k\sqrt{C_k}\right)^2 = -2l\,{B_k}^2\left(\frac{\varepsilon}{B_k} + \sqrt{C_k}\right)^2 = -2{B_k}^2 D_k$$

$$\sum_{k=1}^{l} 2e^{-2l\left(\varepsilon + B_k\sqrt{C_k}\right)^2} = \sum_{k=1}^{l} 2e^{-2{B_k}^2 D_k}$$

Se però noi ci focalizzassimo sulla ultima terz'ultima equazione avremmo che

$$\left(\frac{\varepsilon^2}{{B_k}^2} + \frac{2\varepsilon\sqrt{C_k}}{B_k} + C_k\right) \geq \frac{\varepsilon^2}{{B_k}^2}$$

Perché tutte le grandezze in gioco sono positive ma quindi moltiplicando ad entrambe $-l$ otterremmo

$$-l\left(\frac{\varepsilon^2}{{B_k}^2} + \frac{2\varepsilon\sqrt{C_k}}{B_k} + C_k\right) \leq -\frac{\varepsilon^2 l}{{B_k}^2}$$

ma visto che questi sono gli argomenti dell'esponenziale si ha

$$\sum_{k=1}^{l} 2\exp\left(-2l\left(\varepsilon + B_k\sqrt{C_k}\right)^2\right) = \sum_{k=1}^{l} 2\exp\left(-2{B_k}^2 D_k\right) \leq \sum_{k=1}^{l} 2\exp\left(-\frac{\varepsilon^2 l}{{B_k}^2}\right)$$

Eliminando la costante e dividendo per l si ha

$$\frac{1}{l}\sum_{k=1}^{l} \exp\left(-2{B_k}^2 D_k\right) \leq \frac{1}{l}\sum_{k=1}^{l} \exp\left(-\frac{\varepsilon^2 l}{{B_k}^2}\right) \leq \exp\left(-\frac{\varepsilon^2 l}{{B_l}^2}\right)$$

Perché l'elemento centrale è una media che sarà evidentemente inferiore al più alto valore della sommatoria. Infine considerando un δ: ${B_l}^2 \leq l^{1-\delta}$ otteniamo che

$$\exp\left(-\frac{\varepsilon^2 l}{{B_l}^2}\right) \leq \exp\left(-\varepsilon^2 l^{\delta}\right)$$

In definitiva abbiamo ottenuto che

$$P\left[R(\alpha_l^+) - R_{emp}^+(\alpha_l^+) > \varepsilon\right] \leq \sum_{k=1}^{l} P\left(R(\alpha_l^+) - R_{emp}(\alpha_l^k) > \varepsilon + B_k\sqrt{C_k}\right) \leq \exp\left(-\varepsilon^2 l^{\delta}\right)$$

e quindi

$$\lim_{l\to\infty} P\left[R(\alpha_l^+) - R_{emp}^+(\alpha_l^+) > \varepsilon\right] \leq \lim_{l\to\infty} \exp\left(-\varepsilon^2 l^{\delta}\right) = 0$$

non potendo essere una probabilità negativa si ottiene in definitiva

$$\lim_{l\to\infty} P\left[R(\alpha_l^+) - R_{emp}^+(\alpha_l^+) > \varepsilon\right] = 0$$

Non ci resta che verificare sotto quali condizioni la seconda componente B si azzera, ricordando che

$$B = R_{emp}^{+}(\alpha_l^{+}) - R(\alpha_0)$$

Dall'assunzione che S^* sia denso su tutto S ricaviamo che

1. $\forall \varepsilon \ \exists \ S_s : \ R(\alpha_l^s) - R(\alpha_0) < \varepsilon$
2. $\forall \varepsilon \ \exists \ l_0 < l : \ B_s \sqrt{\frac{h_s\left(\ln\frac{2l}{h_s}+1\right)+2\ln(2l)}{l}} = B_s\sqrt{C_s} < \frac{\varepsilon}{2}$

Quindi quando $l_0 < l$ abbiamo

$$P\left(\min_{1\le k\le l}\left\{R_{emp}^{+}(\alpha_l^k)\right\} - R(\alpha_0^s) > \varepsilon\right) \le P\left(R_{emp}^{+}(\alpha_l^s) - R(\alpha_0^s) > \varepsilon\right)$$

Sostituendo ad $R_{emp}^{+}(\alpha_l^s)$ la sua definizione come si era fatto con la componente A otteniamo che

$$P\left(R_{emp}^{+}(\alpha_l^s) - R(\alpha_0^s) > \varepsilon\right) = P\left(R_{emp}(\alpha_l^s) - R(\alpha_0^s) > \varepsilon - B_s\sqrt{C_s}\right)$$

$$P\left(R_{emp}(\alpha_l^s) - R(\alpha_0^s) > \varepsilon - B_s\sqrt{C_s}\right) \le P\left(R_{emp}(\alpha_l^s) - R(\alpha_0^s) > \frac{\varepsilon}{2}\right)$$

da cui

$$P\left(R_{emp}(\alpha_l^s) - R(\alpha_0^s) > \frac{\varepsilon}{2}\right) \le P\left(\sup_{\alpha\in\Lambda}\left|R(\alpha) - R_{emp}(\alpha)\right| > \frac{\varepsilon}{2}\right)$$

E sfruttando ancora la diseguaglianza di Hoeffding con ragionamenti analoghi a quelli visti per la componente A

$$P\left(\sup_{\alpha\in\Lambda}\left|R(\alpha) - R_{emp}(\alpha)\right| > \frac{\varepsilon}{2}\right) \le \left(\frac{2l}{h_s}\right)^{h_s} \exp\left(h_s - \frac{\varepsilon^2 l^{\delta}}{4}\right)$$

Ed essendo l'esponenziale più veloce a tendere a zero al divergere di l la parte a destra della diseguaglianza si azzera. Otteniamo quindi che

$$\lim_{l\to\infty} P\left[R_{emp}^{+}(\alpha_l^{+}) - R(\alpha_0) > \varepsilon\right] = 0$$

Dato che entrambe le componenti si azzerano per l che tende a più infinito ricaviamo che

$$\lim_{l\to\infty} P[R(\alpha_l^{+}) - R(\alpha_0) > \varepsilon] = 0$$

quindi

$$R(\alpha_l^{+}) \to R(\alpha_0)$$

Se questo è vero allora quando la soluzione ottimale appartiene ad un sottoinsieme S_s della struttura vale che

$$R(\alpha_0^s) = R(\alpha_0)$$

combinando i vincoli di entrambi i termini e considerando un l_0 per cui valga la condizione

$$B_s\sqrt{C_s} < \frac{\varepsilon}{2}$$

abbiamo

$$P[R(\alpha_l^+) - R(\alpha_0) > \varepsilon] \leq P\left(A \geq \frac{\varepsilon}{2}\right) + P\left(R(\alpha_l^+) - R(\alpha_0) > \frac{\varepsilon}{2}\right)$$
$$\leq \exp\left(-\frac{\varepsilon^2 l}{4\mu(l)}\right) + \left[\left(\frac{2l}{h_s}\right)^{h_s} \exp\left(h_s - \frac{\varepsilon^2 l}{16\mu(l)}\right)\right] = V(l)$$

otteniamo quindi il tasso di convergenza come

$$V(l) = O\left(\sqrt{\frac{\mu(l)\ln l}{l}}\right)$$

Possiamo quindi formalizzare i risultati dei ragionamenti di cui sopra nel seguente enunciato

Se S è una struttura ammissibile allora vale che

1. se ${B_l}^2 \leq l^{1-\delta} \Rightarrow R(\alpha_l^+) \to R(\alpha_0)\ \forall Z$
2. se $Q(z, \alpha_0) \in S_0$ e ${B_{n(l)}}^2 \leq \mu(l) \leq l^{1-\delta} \Rightarrow V(l) = O\left(\sqrt{\frac{\mu(l)\ln l}{l}}\right)$

in sostanza il teorema ci dice che per ogni Z l'SRM da una soluzione che converge alla migliore soluzione possibile con quel set di ipotesi con probabilità 1.

Conclusione della prima parte

Con l'SRM si conclude la prima parte del nostro percorso nella teoria dell'apprendimento statistico. Fin qui abbiamo definito cosa vuol dire apprendere nel nostro ambito e sotto quali condizioni ciò possa avvenire.

In estrema sintesi potremmo dire che con apprendere intendiamo la ricerca di una funzione di classificazione $f(x, \alpha)$ che sia in grado di ottenere un errore atteso generico (ossia non limitato ai soli valori del campione) sufficientemente basso in base ai nostri obiettivi conoscitivi.

Abbiamo poi visto come sia possibile definire una relazione in probabilità tra questo rischio generico (non calcolabile per varie ragioni) ed una grandezza definita dalla somma tra il rischio empirico e una misura di complessità del set di funzioni che utilizziamo per la classificazione.

$$R(\alpha_l) \leq R_{emp}(\alpha_l) + \Omega(h, l, \eta)$$

Quest'ultima relazione ci ha fatto intuire che nei casi reali (in cui la dimensione campionaria è limitata) la miglior performance non si ottiene minimizzando il solo rischio empirico ma la somma delle due componenti a destra della diseguaglianza.

Infine abbiamo dimostrato come effettivamente la soluzione che rende minima questa somma $R_{emp}(\alpha_l) + \Omega(h, l, \eta)$ converga alla miglior soluzione possibile quando la dimensione campionaria diverge a più infinito.

Sulla base di questo apparato teorico dovremo ora chiederci come costruire dei modelli di classificazione che si basino su questi risultati. Infatti quanto fin'ora visto non è altro che una "giustificazione teorica" ad un possibile approccio nella ricerca del classificatore ottimale ma non ci dice come effettivamente definirlo.

E' però evidente che grazie alla conoscenza ottenuta potremmo definire dei modelli di classificazione che per costruzione abbiano proprietà che li rendano dei buoni classificatori ossia che riescano effettivamente ad apprendere.

In particolare la teoria dell'apprendimento statistico ci permetterà di

1. definire un algoritmo di ricerca della soluzione ottimale che si basi sul principio di SRM evitando così di incorrere in problemi di overfitting
2. quantificare la capacità di generalizzazione della soluzione ottenuta sulla base delle equazioni fondamentali definite
3. confrontare diversi modelli di classificazione sulla base dei risultati teorici ottenuti

Nel prossimo capitolo verrà presentato il funzionamento della learning mahcine che più si ispira alla teoria dello statistical learning ossia le c.d. macchine a vettori di supporto (SVM).

A seguito tratteremo due modelli alternativi alle SVM che sono le reti neuronali e gli alberi decisionali. Le prime per vedere come il principio di SRM possa essere applicato in un modo opposto rispetto a quanto si fa nelle SVM, gli alberi decisionali verranno invece trattati per mostrare con maggior chiarezza il trade off che intercorre tra complessità del modello e precisione dello stesso (già intuita nel capitolo del SRM).

Seconda Parte

1.Introduzione alle SVM

In questo capitolo vedremo come sfruttare i risultati teorici visti nella prima parte per poter effettivamente costruire dei modelli di classificazione in grado di apprendere.

Essendo ancora una fase introduttiva partiremo con il più semplice modello di classificazione ossia quello lineare dicotomico (in cui esistono due sole classi).

Supponiamo quindi di avere a disposizione un training set definito come l'insieme delle l coppie $\{(x^1, y^1), (x^2, y^2), \dots, (x^l, y^l)\}$ tali che

- $x^i \in R^n$ informazione sull'osservazione
- $y^i \in \{-1, +1\}$ etichetta che indica la classe di appartenenza

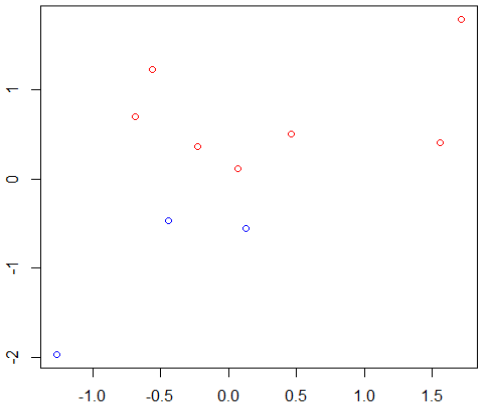
Ogni osservazione si definisce da un vettore di n informazioni (che per semplicità assumeremo siano tutti valori numerici) a cui si affianca una etichetta che indica a quale gruppo appartiene.

Il modo più semplice di eseguire una classificazione consiste nell'assegnare ogni osservazione ad un gruppo in base a come si posiziona rispetto ad una qualche funzione $f(x, \alpha)$ definita nello spazio campionario.

Di fatto $f(x, \alpha)$ non fa altro che partizionare lo spazio di input in due zone che fanno riferimento alle due possibili classi. In questo modo una osservazione viene assegnata ad una classe in base alla zona dello spazio che la contiene.

Per definire quale $f(x, \alpha)$ utilizzare evidentemente ci baseremo sull'informazione campionaria disponibile posto che il nostro obiettivo è individuare un classificatore che in generale commetta un errore sufficientemente basso.

Supponiamo per semplicità che il training set sia rappresentabile sul piano cartesiano, avremo quindi il seguente grafico.



Assumiamo inoltre che le osservazioni campionarie siano linearmente separabili ossia che sia possibile definire una retta in grado di classificare correttamente tutti punti.

Com'è evidente dal grafico è non esiste una sola retta in grado di separare correttamente tutti i punti ma infinite.

Infatti tracciando una qualsiasi retta che separi tutti i punti è possibile "spostare" questa funzione senza scontrarsi con nessuna delle osservazioni campionarie e quindi senza modificare la classificazione complessiva.

Sorge spontaneamente la seguente domanda:

$$quale\ retta\ scegliere?$$

Evidentemente se sono possibili infinite rette che danno una classificazione perfetta il problema è individuare quella che in generale da un rischio di errore $R(\alpha)$ minore secondo quanto abbiamo visto nella prima parte.

Intuitivamente la soluzione migliore è quella che partiziona lo spazio campionario garantendo la maggior distanza tra i due cluster di osservazioni. Questo perché un punto che cade vicino al separatore ha una maggior probabilità di non essere classificato correttamente rispetto ad una osservazione che cade in un zona più "interna". L'obiettivo dunque è quello di minimizzare il numero di punti vicini alla frontiera.

Vediamo quindi di impostare il problema in termini più formali: consideriamo due insiemi di punti $A, B \in R^n: A \cap B = \emptyset$ e assumiamo che A e B siano linearmente separabili.

Ciò implica che esiste un iperpiano $H = \{x \in R^n: w^T x + b = 0\}$ che separa lo spazio in modo tale per cui tutti i punti $x^i \in A$ appartengano ad un'area e tutti punti $x^i \in B$ all'altra. Assumiamo inoltre che attorno a questo iperpiano sia definito una sorta di margine di tolleranza ε ossia una distanza minima che tutti i punti devono avere rispetto all'iperpiano.

In termini formali ciò si traduce in

$$\exists(w, b): \begin{cases} w^T x^i + b \geq \varepsilon \ \ \forall x^i \in A \ con \ \varepsilon > 0 \\ w^T x^i + b \leq \varepsilon \ \ \forall x^i \in B \ con \ \varepsilon > 0 \end{cases}$$

infatti con $(w^T x + b)$ indichiamo l'equazione del generico iperpiano di separazione che costituisce il luogo di punti ove $(w^T x + b) = 0$ (si osservi che la notazione è in termini vettoriali quindi nella x si hanno tutte le coordinate che definiscono il punto)

La scrittura $(w^T x^i + b)$ restituisce il valore di un i −esimo punto rispetto all'iperpiano, questo sarà positivo se si trova "sopra" la superficie di classificazione e negativo altrimenti.

Semplifichiamo il tutto dividendo per ε (mantenendo per semplicità la simbologia di w e b pur non essendo identici a quelli di cui sopra)

$$\begin{cases} w^T x^i + b \geq 1 \ \ \forall x^i \in A \\ w^T x^i + b \leq 1 \ \ \forall x^i \in B \end{cases}$$

Come si è intuito l'iperpiano migliore è quello che garantisce la massima distanza fra i due gruppi, d'altro canto un cluster è definito da una pluralità di osservazioni quindi esistono l distanze diverse dall'iperpiano. Quale considerare?

Dato che il problema di ricerca dell'iperpiano di ottimo risiede nei punti di frontiera (ossia quelli più vicini al separatore) considereremo come distanza di riferimento quella del punto più vicino all'iperpiano.

Questa distanza minima è detta *margine di separazione* ed in termini formali si definisce come:

$$\rho(w,b) = \min_{x^i \in A \cup B} \frac{|w^T x^i + b|}{||w||}$$

ove $H(w,b)$ è l'iperpiano di separazione dei punti $x^i \in A \cup B$.

Possiamo quindi individuare l'iperpiano di separazione migliore risolvendo il seguente problema di ottimizzazione

$$\max_{w \in R^n, b \in R} \rho(w,b) = \max_{w \in R^n, b \in R} \left(\min_{x^i \in A \cup B} \frac{|w^T x^i + b|}{||w||} \right)$$

Dobbiamo quindi massimizzare la distanza tra l'iperpiano ed il punto più vicino, posto così il problema risulta un po' contorto e sicuramente difficilmente implementabile in un software. Cerchiamo quindi un modo per ritradurlo in un problema equivalente ma più diretto.

Innanzitutto definiamo con x^* il punto più vicino all'iperpiano, questo evidentemente avrà una distanza d^* minore rispetto alla distanza di un qualsiasi altro punto del training set

$$d^* = \frac{|w^T x^* + b|}{||w||} \leq \frac{|w^T x^i + b|}{||w||} \qquad \forall x^i \in A \cup B$$

Si osservi ora che l'iperpiano è una figura geometrica invariante alla scala di osservazione, ciò significa che se moltiplicassimo una costante a ad (w,b) otterremmo un iperpiano (aw, ab) che in realtà è lo stesso.

$$3y = 4x + 1 \text{ è } equivalente\ a\ 15y = 20x + 5$$

Per semplificare al notazione del problema consideriamo quindi la scrittura dell'iperpiano (w,b) per cui la distanza rispetto a x^* sia normalizzata ossia

$$|w^T x^* + b| = 1$$

quindi d^* diventa semplicemente

$$d^* = \frac{1}{||w||}$$

Il nostro problema

$$\max_{w \in R^n, b \in R} \rho(w, b) = \max_{w \in R^n, b \in R} \left(\min_{x^i \in A \cup B} \frac{|w^T x^i + b|}{||w||} \right)$$

è equivalente a

$$\max_{w \in R^n, b \in R} \frac{1}{||w||}$$

$$sub \min_{x^i \in A \cup B} |w^T x^i + b| = 1$$

La scrittura è sicuramente già più semplice rispetto alla precedente ma avere un vincolo con un minimo potrebbe porre dei problemi. Dobbiamo cercare una nuova formulazione che semplifichi ulteriormente quanto scritto sopra senza modificare il problema.

Innanzi tutto osserviamo che la condizione di minimo di cui sopra può essere ritradotta in l condizioni

$$\left| w^T x^i + b \right| \geq 1 \ \ \forall x^i \in A \cup B$$

evidentemente non potremo avere una soluzione x^* per cui $|w^T x^i + b| > 1$ perché sarebbe sempre possibile riscalare l'iperpiano fino ad ottenere $|w^T x^i + b| = 1$. D'altro canto queste l condizioni garantiscono che l'iperpiano classifichi correttamente tutti i punti del training set.

In secondo luogo osserviamo che massimizzare una frazione con numeratore costante equivale a minimizzarne il denominatore. Moltiplicando il tutto per una costante $1/2$ non stiamo modificando il punto di ottimo e contemporaneamente ciò ci semplificherà il calcolo delle derivate.

In definitiva il problema

$$\max_{w \in R^n, b \in R} \frac{1}{||w||}$$

$$sub \min_{x^i \in A \cup B} |w^T x^i + b| = 1$$

equivale al più diretto

$$\min_{w \in R^n, b \in R} \frac{1}{2} w^T w$$

$$sub\ \left|w^T x^i + b\right| \geq 1\ \ \forall x^i \in A \cup B$$

Definito il problema dobbiamo ora cercare di dimostrare se questo ha di fatto una soluzione e se tale soluzione è anche unica.

Per il primo punto dovremmo considerare l'insieme di tutti i possibili iperpiani capaci di classificare correttamente il training set e verificare che esista un particolare iperpiano appartenente a questo gruppo che vanta il minor $w^T w$.

Definiamo con F l'insieme degli iperpiani orientati ammissibili ossia che classificano correttamente l'intero training set

$$F = \{(w, b): w^T x^i + b \geq 1\ \forall x^i \in A\ \wedge w^T x^i + b \leq -1\ \forall x^i \in B\}$$

fissato un iperpiano $(w_0, b_0) \in F$ con $\|w_0\|$ definiamo con L_0 l'insieme degli iperpiani ammissibili tali che $\|w\|^2 \leq \|w_0\|^2$

$$L_0 = \{(w, b) \in F: \|w\|^2 \leq \|w_0\|^2\}$$

Se questo insieme di livello L_0 fosse chiuso e limitato vorrebbe dire che esiste un numero limitato di iperpiano $(w, b) \in F$ con $\|w\|^2 \leq \|w_0\|^2$ e che quindi l'iperpiano con il più basso $\|w\|^2$ appartenente a L_0 sarebbe anche il minimo dell'insieme F.

Infatti se si suppone per assurdo che esiste una sequenza illimitata $\{w_k, b_k\} \in L_0$ si nota che:

- $\begin{cases} \|w_k\| \leq \|w_0\| \\ \|\ .\ \| \in [0 + \infty) \end{cases}$

  i valori di $\|w_k\|$ oscillano tra 0 e $\|w_0\|$ allora perché si abbia una sequenza illimitata di $\{w_k, b_k\} \in L_0$ deve variare b_k e ciò al limite comporta che $|b_k| \to +\infty$

- $\forall k$ deve comunque valere che

  $$\begin{cases} {w_k}^T x^i + b_k \geq +1\ \forall x^i \in A \\ {w_k}^T x^i + b_k \leq -1\ \forall x^i \in B \end{cases}$$

  per k sufficientemente grandi si arriverebbe alla situazione in cui (essendo $|b_k| \to +\infty$) $\|w_k\|^2 > \|w_0\|^2$ che contraddice l'ipotesi di appartenenza ad L_0.

Allora L_0 è compatto (chiuso e limitato) quindi per il teorema di Weistrass la funzione $\|w\|^2$ ammette un minimo assoluto (w^*, b^*) su L_0 e quindi anche su F (perché L_0 è una sorta di *insieme inferiore* di F e l'obiettivo è appunto la sua minimizzazione).

Dunque (w^*, b^*) è la soluzione del problema di $\{\min\|w\|^2\}$ quello che ci manca è dimostrare che questa soluzione è anche unica e che quindi (w^*, b^*) è un punto di minimo globale.

Affrontiamo questa problematica con un ragionamento molto intuitivo: dovremo dimostrare che se due iperpiani hanno lo stesso valore di $\|w\|^2$ allora coincidono, ne segue che l'iperpiano di ottimo (w^*, b^*) è unico perché non esistono iperpiani diversi con lo stesso valore di $\|w^*\|^2$.

Assumiamo per assurdo che esista una altro iperpiano di separazione $(\bar{w}, \bar{b})$ diverso da (w^*, b^*) che però ha lo stesso valore di $\|w^*\|$

$$\exists(\bar{w}, \bar{b}) \in F: (\bar{w}, \bar{b}) \neq (w^*, b^*) \ \wedge \ \|\bar{w}\|^2 = \|w^*\|^2$$

Essendo F un insieme convesso deve valere che per ogni coppia di punti che vi appartengono il tratto che li unisce appartiene anch'esso a F.

$$\lambda(w^*, b^*) + (1-\lambda)(\bar{w}, \bar{b}) \in F \ \forall\lambda[0,1]$$

Ponendo $\lambda = 1/2$ si definisce un iperpiano $(\tilde{w}, \tilde{b}) = \left(\frac{1}{2}w^* + \frac{1}{2}\bar{w}, \frac{1}{2}b^* + \frac{1}{2}\bar{b}\right)$ che appartiene a F (per la sua convessità) e che a fronte della convessità della funzione $\|\,.\,\|^2$ verifica la seguente disequazione

$$\|\lambda w^* + (1-\lambda)\bar{w}\|^2 = \left\|\frac{1}{2}w^* + \frac{1}{2}\bar{w}\right\|^2 = \|\tilde{w}\|^2 < \frac{1}{2}\|w^*\|^2 + \frac{1}{2}\|\bar{w}\|^2$$

Ma se $\|\bar{w}\|^2 = \|w^*\|^2$ allora

$$\frac{1}{2}\|w^*\|^2 + \frac{1}{2}\|\bar{w}\|^2 = \|w^*\|^2 = \|\bar{w}\|^2 \implies \|\tilde{w}\|^2 < \|\bar{w}\|^2 = \|w^*\|^2$$

il che contraddice il fatto che (w^*, b^*) sia un minimo globale dunque è necessario che $w^* \equiv \bar{w}$.

Assumiamo ora che $b^* < \bar{b}$ e prendendo il punto più prossimo dell'insieme A ($\hat{x}^i \in A: {w^*}^T\hat{x}^i + b^* = 1$) si ha che

$${w^*}^T\hat{x}^i + b^* = 1 = \bar{w}^T\hat{x}^i + b^* > \bar{w}^T\hat{x}^i + \bar{b}$$

che contraddice la condizione necessaria affinché si abbia un iperpiano di separazione ossia che

$$\bar{w}^T x^i + \bar{b} \geq 1 \ \forall x^i \in A$$

quindi è necessario che $b^* \equiv \bar{b}$.

Con questo ragionamento abbiamo è dimostrato che se due iperpiani hanno lo stesso valore di $\|w\|^2$ necessariamente coincidono quindi una volta che si è trovato l'ottimo (w^*, b^*) è garantito che non esistono altri iperpiani con $\|w\|^2 < \|w^*\|^2$.

Con questo primo capitolo abbiamo introdotto la logica fondamentale su cui si basano i classificatori lineari che massimizzano il margine di separazione. Sotto l'ipotesi che il data set sia linearmente separabile abbiamo dimostrato che effettivamente esiste un iperpiano in grado di massimizzare il margine di separazione e che questo è unico.

Nel capitolo successivo vedremo come riscrivere nuovamente il problema di ottimizzazione al fine di semplificarne al massimo l'implementazione ma prima è bene però fare un piccolo ragionamento ricollegandoci a quanto visto nella prima parte.

Dato il problema di ottimo fin'ora considerato

$$\min_{w \in R^n, b \in R} \frac{1}{2} w^T w$$

$$sub \left| w^T x^i + b \right| \geq 1 \ \ \forall x^i \in A \cup B$$

osserviamo che se esiste una soluzione (e abbiamo dimostrato che ciò accade quando i punti sono linearmente separabili) allora tutti gli l vincoli sono stati soddisfatti.

Questo significa che l'iperpiano di ottimo (w, b) classifica correttamente tutti i punti del training set quindi l'errore empirico è nullo. Riprendendo la diseguaglianza fondamentale otteniamo quindi

$$R(\alpha_l) \leq R_{emp}(\alpha_l) + \Omega(h, l, \eta) = \Omega(h, l, \eta)$$

E' noto che se la dimensione campionaria non è elevata la componente $\Omega(h, l, \eta)$ non si azzera e quindi il vincolo superiore del rischio atteso risulta elevato.

Sappiamo che $\Omega(h, l, \eta)$ è inversamente legata alla VC dimension del set di funzioni utilizzato nella ricerca della soluzione e che quest'ultima nel caso di funzioni lineari è pari ad $n + 1$ ove n è la dimensione dello spazio su cui si definisce il classificatore.

Ciò che ora introduciamo è un teorema fondamentale nella logica della costruzione dei SVM il quale giustifica quanto abbiamo intuito sul fatto che l'iperpiano migliore sia quello che massimizza il margine di separazione.

Si dimostra che l'insieme di iperpiani con margine ρ ha $VC\ dimension\ h$ tale che

$$h \leq min\left[\left(\frac{D}{\rho}\right)^2, n\right] + 1$$

ove D è il raggio della più piccola sfera in grado di contenere tutti i punti del training set e ρ l'ampiezza del margine di separazione ottenuto.

Posto che D sia un valore costante osserviamo che se fossimo in grado di ottenere un $(D/\rho)^2 < n$ potremmo effettivamente definire un limite superiore di h minore a quello che fin'ora pensavamo invalicabile definito da n + 1.

Non solo, rispetto a quando abbiamo trattato la logica del SRM notiamo che un iperpiano definito su n dimensioni ma con un margine molto elevato potrebbe ottenere una VC dimension minore di un altro iperpiano definito su $m < n$ dimensioni ma con un margine più contenuto. In sostanza non siamo costretti a dover utilizzare un set di funzioni più semplice quando è possibile ottenere una margine molto elevato.

Quindi nel caso di punti linearmente separabili *l'iperpiano che massimizza il margine è quello che garantisce una miglior performance di generalizzazione perché è in grado di minimizzare* $\Omega(h, l, \eta)$ *posto che il rischio empirico sia zero*.

Come vedremo sarà quindi possibile allargare ulteriormente questo margine ρ accettando il fatto che l'iperpiano non separi correttamente tutti i punti. In sostanza sopportiamo l'avere un rischio empirico diverso da zero per poter ulteriormente abbassare $\Omega(h, l, \eta)$ secondo la logica a noi ben nota del SRM che viene attivamente sfruttata nella risoluzione del nostro problema.

Nota: Condizione perché un set di punti sia linearmente separabile

Un insieme di punti $\{x^1, \dots, x^m\} \subset R^n$ è linearmente separabile se e solo se $x^1, \dots, x^m$ sono affinamente indipendenti.

Un insieme di punti $\{x^1, \dots, x^m\} \subset R^n$ è affinamente dipendente se esistono scalari $w_1, \dots w_m$ non tutti nulli tali che

$$\sum_{i=1}^{m} w_i x^i = 0 \; con \; \sum_{i=1}^{m} w_i = 0$$

inoltre si dimostra che un insieme di punti $\{x^1, \dots, x^m\}$ è affinamente indipendente se e solo se i vettori scarto $x^i - x^j$ con $i = 1, \dots m$ e $i \neq j$ sono linearmente indipendenti

Assumendo che il primo punto sia l'origine ($x^1 = 0$) e che $x^1, \dots, x^m$ siano affinamente indipendenti si ha che i vettori scarto

$$x^2 - x^1, \dots, x^m - x^1 = x^2, \dots, x^m$$

sono linearmente indipendenti.

Quello che si vuole dimostrare è che in tale situazione comunque si scelgano le etichette $y^1, \dots, y^m$ $\exists$ un $\overline{w} \in R^n$ (vettore di pesi) e un $\overline{b} \in R$ (soglia) in grado di classificare correttamente tutti i punti

$$y^i\left(\overline{w}^T x^i + \overline{b}\right) > 0 \quad \forall i = 1, \dots, m$$

Ponendo

$$\overline{b} = \begin{cases} +\frac{1}{2} \; se \; y^i = +1 \\ -\frac{1}{2} \; se \; y^i = -1 \end{cases}$$

per $x^1 = 0$ si ottiene che $y^1\left(w^T x^1 + \overline{b}\right) = y^1 \overline{b} > 0 \; \forall \, w \in R^n$ dunque il primo punto è sempre classificato correttamente data tale definizione di $\overline{b}$.

Data l'indipendenza lineare tra $x^2, \dots, x^m$ si ha che

$$w^T x^2 + \cdots + w^T x^m = 0 \Leftrightarrow w = 0$$

Dato che $y^i = \pm 1$ e non 0 il sistema lineare con incognita w

$$\begin{pmatrix} (x^2)^T \\ \dots \\ (x^m)^T \end{pmatrix} w = \begin{pmatrix} y^2 \\ \dots \\ y^m \end{pmatrix}$$

ammette almeno una soluzione $\overline{w}$ diversa da zero, sommando nel sistema $\overline{b}$ avremo

$$\left(\overline{w}^T x^i + \overline{b}\right) = (y^i + \overline{b}) = \begin{cases} +1 + \overline{b} = +\frac{3}{2} > 0 \ \forall i = 2, \dots, m : y^i = +1 \\ -1 + \overline{b} = -\frac{3}{2} < 0 \ \forall i = 2, \dots, m : y^i = -1 \end{cases}$$

e quindi

$$y^i\left(\overline{w}^T x^i + \overline{b}\right) = (y^i)^2 + \overline{b} > 0 \ \forall i = 2, \dots, m$$

Tutti i punti $x^2, \dots, x^m$ sono correttamente classificati. Ricordando che anche il punto x^1 è stato correttamente classificato ne segue che l'intero training set è stato separato dall'iperpiano $\left(\overline{w}, \overline{b}\right)$ in maniera corretta.

2.Il Duale di Wolfe

In questo capitolo apriremo una parentesi su uno strumento che risulterà essere fondamentale per la risoluzione del nostro problema di ottimizzazione

Questo strumento si basa sul concetto matematico di dualità di un problema di ottimizzazione che in sostanza consiste nell'individuare un problema equivalente al primo (cioè con la stessa soluzione) ma più semplice da risolvere.

L'intuizione è costruire un problema di massimo partendo dal problema di minimo assegnato. Definiamo dunque:

1. Problema Primale $\min_{x\in S} f(x)$
2. Problema Duale $\max_{u\in U} \Psi(u)$

perché sia possibile eseguire tale passaggio è necessario che sia verificata una delle due condizioni di dualità

1. Condizione Forte $inf_{x\in S} f(x) = sup_{u\in U}\Psi(u)$
2. Condizione Debole $inf_{x\in S} f(x) \geq sup_{u\in U}\Psi(u)$

Intuitivamente se il punto di minimo di $f(x)$ equivale al massimo di $\Psi(u)$ trovando quest'ultimo otteniamo anche l'ottimo della prima equazione. Fortunatamente la condizione forte è spesso presente nei problemi di programmazione convessa (come il nostro).

Specifichiamo il discorso introducendo anche un set di vincoli

$$\min f(x)$$

$$sub \begin{cases} g_i(x) \leq 0 & i = 1, \dots, m \\ c_j^T x - d_j = 0 & j = 1, \dots p \end{cases}$$

con $f(x): R^n \to R$ e $g_i(x): R^n \to R \ \forall i = 1, \dots, m$ assunte convesse e continuamente differenziabili.

Possiamo quindi definire il lagrangiano del problema come

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{p} \mu_j (c_j^T x - d_j)$$

Il quale a sua volta definisce un problema di ottimizzazione del tipo

$$\max_{x,\lambda,\mu} L(x,\lambda,\mu)$$

$$sub \begin{cases} \nabla_x L(x,\lambda,\mu) = 0 \\ \lambda \geq 0 \end{cases}$$

L'obiettivo è dimostrare che la soluzione del problema di minimo x^* associata alla rispettiva coppia di moltiplicatori di lagrange (λ^*, μ^*) determina un punto (x^*, λ^*, μ^*) che è a sua volta la soluzione del problema di massimizzazione del lagrangiano e che il valore che le due funzioni ($f(x)$ e $L(x,\lambda,\mu)$) assumono nei rispettivi ottimi è lo stesso.

In questo modo sappiamo che basta risolvere il più semplice dei due problemi per poter ricavare la soluzione dell'altro.

Assumiamo quindi che (x^*, λ^*, μ^*) sia il punto di ottimo del problema di massimo, il lagrangiano in quel punto assumerà un valore pari a

$$L(x^*,\lambda^*,\mu^*) = f(x^*) + \sum_{i=1}^{m} \lambda^*{}_i g_i(x^*) + \sum_{j=1}^{p} \mu^*{}_j (c_j^T x - d_j)$$

dalle condizioni di Karush Kuhn Tucker (KKT da ora in poi) ricaviamo che:

$$\begin{cases} \nabla_x L(x^*,\lambda^*,\mu^*) = 0 \\ (\lambda^*)^T g(x^*) = 0 \\ \lambda^* \geq 0 \end{cases}$$

ma si osservi che

$$(\lambda^*)^T g(x^*) = \sum_{i=1}^{m} \lambda^*{}_i g_i(x^*) = 0$$

$$c_j^T x - d_j = 0 \; \forall j \Rightarrow \sum_{j=1}^{p} \mu_j (c_j^T x - d_j) = 0$$

quindi

$$L(x^*,\lambda^*,\mu^*) = f(x^*) + \sum_{i=1}^{m} \lambda^*{}_i g_i(x^*) + \sum_{j=1}^{p} \mu^*{}_j (c_j^T x - d_j) = f(x^*)$$

Abbiamo agilmente dimostrato che $L(x^*,\lambda^*,\mu^*) = f(x^*)$, ora dobbiamo dimostrare che effettivamente (x^*, λ^*, μ^*) è la soluzione del problema di massimo. In questo modo risolvendo il più semplice dei due problemi avremo automaticamente la soluzione di quello più complesso.

Per dimostrare che (x^*, λ^*, μ^*) è il massimo assoluto ci basta dimostrare che la funzione di lagrange assumerebbe un valore inferiore in un qualsiasi altro punto che considera la stessa x^* ma una diversa coppia di moltiplicatori di lagrange.

Sia (x, λ, μ) un punto tale per cui $\nabla_x L(x, \lambda, \mu) = 0$ e $\lambda \geq 0$, si osservi che la funzione

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{p} \mu_j (c_j^T x - d_j)$$

è convessa perché:

1. $f(x)$ è convessa per ipotesi
2. $g_i(x)$ è per ipotesi convessa e si dimostra che una combinazione lineare con coefficienti non negativi di funzioni convesse è ancora convessa
3. il terzo termine è una funzione affine e quindi convessa
4. la somma di funzioni convesse è ancora convessa

quindi per la definizione di convessità

$$L(y) - L(x) \geq (y - x)^T \nabla_x \mathrm{L}(\mathrm{x})$$

Dato che $\lambda \geq 0$ e $g_i(x) \leq 0 \; \forall i$ la grandezza $\lambda^T g(x^*)$ è negativa (è pari a zero se e solo se uso sia x^* e λ^*)

$$L(x^*, \lambda^*, \mu^*) = f(x^*) \geq f(x^*) + \sum_{i=1}^{m} \lambda_i g_i(x^*)$$

Essendo poi che $c_j^T x - d_j = 0 \; \forall j$ vale che $\sum_{j=1}^{p} \mu_j (c_j^T x - d_j) = 0$ e quindi

$$L(x^*, \lambda^*, \mu^*) = f(x^*) \geq f(x^*) + \sum_{i=1}^{m} \lambda_i g_i(x^*) = f(x^*) + \sum_{i=1}^{m} \lambda_i g_i(x^*) + \sum_{j=1}^{p} \mu_j (c_j^T x - d_j)$$
$$= L(x^*, \lambda, \mu)$$

$$L(x^*, \lambda^*, \mu^*) \geq L(x^*, \lambda, \mu)$$

Facendo riferimento al punto $L(x^*, \lambda, \mu)$ osserviamo che data la convessità di $L(\,.\,)$ vale che

$$L(x^*, \lambda, \mu) \geq L(x, \lambda, \mu) + (x^* - x)^T \nabla_x \mathrm{L}(\mathrm{x})$$

ma affinché x sia un punto critico è necessario che $\nabla_x \mathrm{L}(\mathrm{x}) = 0$ per ogni $\lambda \geq 0$ e per ogni μ, quindi

$$L(x, \lambda, \mu) + (x^* - x)^T \nabla_x \mathrm{L}(\mathrm{x}) = L(x, \lambda, \mu) + (x^* - x)^T (0) = L(x, \lambda, \mu)$$

Riassumendo quello che si è ottenuto dal ragionamento di cui sopra è che

$$f(x^*) = L(x^*, \lambda^*, \mu^*) \geq L(x^*, \lambda, \mu) \geq L(x, \lambda, \mu)$$

quindi effettivamente il punto (x^*, λ^*, μ^*) è quello che rende massima la funzione di lagrange e in questo il valore ottenuto è pari a quello ottenuto nel punto di ottimo dal problema di partenza.

Il problema di massimo (del lagrangiano) duale al problema di minimo sopra indicato è il così detto duale di Wolfe.

Visto il problema definito nel capitolo precedente estendiamo il ragionamento fatto può al caso di programmazione quadratica

Si consideri il problema

$$\begin{cases} \min_x f(x) = \frac{1}{2} x'Qx + c'x \\ Ax - b \leq 0 \end{cases}$$

con $Q \in R^{n \times n}, c \in R^n,\ A \in R^{m \times n}, b \in R^m$.

Posto il lagrangiano $L(x, \lambda) = f(x) + \lambda'(Ax - b)$ con $\lambda \in R^m$.il duale di Wolfe è definito come

$$\begin{cases} \max_{x,\lambda} L(x, \lambda) \\ \nabla_x L(x, \lambda) = 0 \\ \lambda \geq 0 \end{cases}$$

Anche in questo caso dobbiamo verificare che esista una equivalenza tra la soluzione del problema di minimo x^* con quella di massimo (x^*, λ^*).

Si consideri la prima condizione del problema duale

$$\nabla_x L(x, \lambda) = Qx + A'\lambda + c = 0$$

eseguendo il trasposto e post-moltiplicando per x

$$x'Q + \lambda'A + c' = 0 \Longrightarrow x'Qx + \lambda'Ax + c'x = 0 \Longrightarrow \lambda'Ax + c'x = -x'Qx$$

sostituendo nel duale si ottiene

$$\max_{x,\lambda} \frac{1}{2} x'Qx + c'x + \lambda'Ax - \lambda'b$$
$$= \max \frac{1}{2} x'Qx - x'Qx - \lambda'b = \max -\frac{1}{2} x'Qx - \lambda'b = \min \frac{1}{2} x'Qx + \lambda'b$$

Il problema duale (di massimo) è stato tradotto in un problema di minimo dunque su questo è possibile riapplicare ad un secondo stadio il duale di Wolfe

$$\begin{cases} \max W(x,\lambda,v,z) \\ \nabla_x W(x,\lambda,v,z) = 0 \\ \nabla_\lambda W(x,\lambda,v,z) = 0 \\ v,z \geq 0 \end{cases}$$

con $W(x,\lambda,v,z) = \frac{1}{2}x'Qx + \lambda'b - v'(Qx + A'\lambda + c) - z'\lambda$.

Sia $(\bar{x},\bar{\lambda})$ una soluzione del problema $\{\max_{x,\lambda} L(x,\lambda)\}$, per quanto detto nel paragrafo precedente sappiamo che il valore che assume la funzione $L(.)$ nel punto di ottimo equivale al valore che il suo lagrangiano $W(.)$ assume nel punto $(\bar{x},\bar{\lambda},\bar{z},\bar{v})$.

$$L(\bar{x},\bar{\lambda}) = W(\bar{x},\bar{\lambda},\bar{z},\bar{v})$$

Abbiamo già dimostrato che il punto $(\bar{x},\bar{\lambda},\bar{z},\bar{v})$ è proprio la soluzione del problema di massimizzazione di $W(\bar{x},\bar{\lambda},\bar{z},\bar{v})$ e quindi vale che

1. $\nabla_x W = Q\bar{x} - Q\bar{v} = 0$
2. $\nabla_\lambda W = b - A\bar{v} - \bar{z} = 0$

Se il punto $(\bar{x},\bar{\lambda})$ è soluzione del problema di minimizzazione di $L(.)$ vale che

3. $\nabla_x L = Q\bar{x} + A'\bar{\lambda} + c = 0$

Infine le rimanenti condizioni di KKT implicano che la i moltiplicatori di lagrange $\bar{v} \in R^n$ e $\bar{z} \in R^r$ verifichino

4. $\bar{z}'\bar{\lambda} = 0$
5. $\bar{z} \geq 0$
6. $\bar{\lambda} \geq 0$

Dalla seconda e quinta condizione si ricava che $b - A\bar{v} \geq 0$ mentre sottraendo dalla prima la terza si ha che $Q\bar{v} + A'\bar{\lambda} + c = 0$, infine sostituendo il risultato della seconda nella quarta si ottiene $\bar{\lambda}'(A\bar{v} - b) = 0$ e quindi le sei condizioni di cui sopra si sintetizzano in

1. $A\bar{v} - b \leq 0$
2. $Q\bar{v} + A'\bar{\lambda} + c = \nabla_x L(\bar{v},\bar{\lambda}) = 0$
3. $\bar{\lambda}'(A\bar{v} - b) = 0$
4. $\bar{\lambda} \geq 0$

Osserviamo che il vettore $\bar{v}$ (che è un insieme di moltiplicatori di lagrange del duale W) rispetta la condizione del problema $\min f(x)$ e che la coppia $(\bar{v}, \bar{\lambda})$ verifica anche le condizioni di KKT affinché sia una soluzione del problema duale $\max L(x, \lambda)$.

Dato quanto abbiamo visto nel caso non quadratico si deduce che $\bar{v}$ è soluzione del problema $\min f(x)$ allora : $\bar{v} = x^*$e quindi il punto $(\bar{v}, \bar{\lambda}) = (x^*, \bar{\lambda})$ è una coppia: minimo globale, moltiplicatore di lagrange.

Per lo stesso ragionamento anche il vettore $\bar{x}$ (che insieme a $\bar{\lambda}$ definiva la soluzione di $\max_{x,\lambda} L(x, \lambda)$) risulta essere soluzione di $\min f(x)$ e se teniamo conto della condizione $\nabla_x W = Q\bar{x} - Q\bar{v} = 0$ ricaviamo

$$\begin{cases} Q\bar{x} = Q\bar{v} \\ x^* = \bar{v} \end{cases} \Longrightarrow Q(x^* - \bar{x}) = 0$$

I risultati ottenuti ci permettono di definire al seguente proposizione:

Assumendo Q semidefinita positiva $(x'Qx \geq 0)$ sia $(\bar{x}, \bar{\lambda})$ una soluzione del duale di Wolfe, allora esiste x^* non necessariamente uguale a $\bar{x}$ tale che

1. $Q(x^* - \bar{x}) = 0$
2. x^* è soluzione del problema primale
3. $(x^*, \bar{\lambda})$ è una coppia: (minimo globale, vettore di moltiplicatori di Lagrange)

3.Programmazione quadratica per SVM lineari

Nel primo capitolo di questa seconda parta abbiamo visto come nel caso di punti linearmente separabili il classificatore migliore fosse quello in grado di massimizzare il c.d. margine di separazione.

Per individuarlo è quindi necessario risolvere il seguente problema di ottimizzazione.

$$\begin{cases} \max \rho(w,b) \\ t.c. \left| w^T x^i + b \right| \geq 1 \ \ \forall x^i \in A \cup B \end{cases}$$

con $A \cap B = \emptyset$ insieme di punti (contenuti in R^n) assunti linearmente separabili.

Si è poi visto che la prima scrittura è equivalente al seguente problema di programmazione quadratica convessa

$$\begin{cases} \min \frac{1}{2} w^T w \\ t.c. \left| w^T x^i + b \right| \geq 1 \ \ \forall x^i \in A \cup B \end{cases}$$

Chiamando la funzione obiettivo $F(w) = \frac{1}{2} w^T w$ e ricordando che il classificatore deve dare una etichetta $y^i = +1 \ \forall x^i \in A$ e $y^i = -1 \ \forall x^i \in B$ possiamo sintetizzare il problema come

$$\begin{cases} \min \ F(w) \\ t.c. \ y^i (w^T x^i + b) - 1 \geq 0 \ \forall \ i = 1, \dots, l \end{cases}$$

ove l è la numerosità dei punti del training set.

Riprendendo quanto appena visto si può definire il duale del problema come

$$\max L(w,b,\lambda) = \frac{1}{2} w^T w - \sum_{i=1}^{l} \lambda_i (\ y^i w^T x^i + \ y^i b - 1)$$

$$sub \begin{cases} \nabla_w L(w,b,\lambda) = w - \sum_{i=1}^{l} \lambda_i \ y^i x^i = 0 \\ \frac{\partial L(w,b,\lambda)}{\partial b} = \sum_{i=1}^{l} \lambda_i \ y^i = 0 \\ \lambda \geq 0 \end{cases}$$

si ricava quindi che il vettore w si definisce come

$$w = \sum_{i=1}^{l} \lambda_i \; y^i x^i$$

sostituendo in $L(w, b, \lambda)$ la definizione di w

$$L(w, b, \lambda) = \frac{1}{2}\left(\sum_{i=1}^{l} \lambda_i \; y^i x^i\right)^T \sum_{j=1}^{l} \lambda_j \; y^j x^j - \sum_{i=1}^{l} \lambda_i (\; y^i \left(\sum_{j=1}^{l} \lambda_j \; y^j x^j\right)^T x^i + \; y^i b - 1)$$

e quindi

$$L(w, b, \lambda) = \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y^i \, y^j \, \lambda_i \lambda_j (x^i)^T x^j - \sum_{i=1}^{l}\sum_{j=1}^{l} y^i \, y^j \, \lambda_i \lambda_j (x^i)^T x^j + b \sum_{i=1}^{l} \lambda_i \, y^i + \sum_{i=1}^{l} \lambda_i$$

ma dato che $\sum_{i=1}^{l} \lambda_i \; y^i = 0$ si ricava

$$L(w, b, \lambda) = -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y^i \, y^j \, \lambda_i \lambda_j (x^i)^T x^j + \sum_{i=1}^{l} \lambda_i = S(\lambda)$$

La funzione obiettivo è dunque diventata una funzione nella sola variabile λ, il problema di massimizzazione diventa

$$\max S(\lambda) = -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y^i y^j (x^i)^T x^j \lambda_i \lambda_j + \sum_{i=1}^{l} \lambda_i$$

$$sub \begin{cases} \sum_{i=1}^{l} \lambda_i \; y^i = 0 \\ \lambda_i \geq 0 \;\; \forall i = 1, \dots, l \end{cases}$$

che equivalente a minimizzare $\Gamma(\lambda) = -S(\lambda)$

$$\min \Gamma(\lambda) = \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y^i y^j (x^i)^T x^j \lambda_i \lambda_j - \sum_{i=1}^{l} \lambda_i$$

$$sub \begin{cases} \sum_{i=1}^{l} \lambda_i \; y^i = 0 \\ \lambda_i \geq 0 \;\; \forall i = 1, \dots, l \end{cases}$$

Possiamo alleggerire la notazione utilizzando la simbologia vettoriale e definendo la matrice $Q = \{(y^i y^j (x^i)^T x^j)\}_{i,j=1,\dots l}$

$$\min \Gamma(\lambda) = \frac{1}{2}\lambda^T Q\lambda - 1^T\lambda$$

$$sub\begin{cases}\lambda^T y = 0\\ \lambda \geq 0\end{cases}$$

Nel primo capitolo abbiamo dimostrato che l'iperpiano ottimale esiste ed è unico (quindi esiste un solo w^*) mentre nel capitolo sul duale di Wolfe abbiamo visto che la soluzione (w^*, b^*, λ^*) del problema duale esiste ed è tale per cui $L(w^*, b^*, \lambda^*) = \frac{1}{2}\|w^*\|^2$ quindi risolvendo il problema di cui sopra otteniamo sia w^* che il margine di separazione.

Con un qualsiasi software di programmazione quadratica convessa si ricava il vettore di ottimo λ^*, questo è tipicamente caratterizzato da un gran numero di elementi nulli perché per le condizione di KKT deve valere che

$$\lambda^*_i(y^i(w^T x^i + b) - 1) = 0 \quad \forall i \in A \cup B$$

ciò accade quando $\lambda^*_i = 0$ e/o $y^i(w^T x^i + b) = 1$ ma abbiamo già visto nei capitoli precedenti che il secondo caso si verifica solo quando il punto (x^i, y^i) ha una distanza dall'iperpiano di $1/\|w\|$.

Quindi per i punti più prossimi all'iperpiano il moltiplicatore di lagrange λ^*_i sarà strettamente maggiore di zero mentre per tutti gli altri è nullo.

Ottenuto il vettore λ^* si ricava immediatamente il vettore w^* come

$$w^* = \sum_{i=1}^{l} \lambda_i^* \; y^i x^i$$

Ma visto che moltissimi punti avranno un moltiplicatore λ^*_i nullo la sommatoria di cui sopra si definisce unicamente dalle coppie (x^i, y^i) che hanno un moltiplicatore di lagrange strettamente positivo.

$$w^* = \sum_{i \in S} \lambda_i^* \; y^i x^i$$

ove S è l'insieme di moltiplicatori di lagrange non nulli.

Per ricavare b^* si osservi che per tutti quei punti che hanno un moltiplicatore di lagrange non nullo vale che

$$\lambda^*_i > 0 \Rightarrow y^i((w^*)^T x^i + b) = 1 \Rightarrow b^* = \frac{1}{y^i} - (w^*)^T x^i \quad \forall i \in S$$

La coppia (w^*, b^*) definisce interamente l'iperpiano che massimizza il margine di separazione. Va però notato che w^* e b^* sono stati ricavati utilizzando i soli (x^i, y^i) a cui si affiancava un $\lambda^*_i > 0$.

Questi punti sono quelli che soddisfano la condizione

$$y^i(w^T x^i + b) = 1$$

e quindi sono in sostanza le coppie più prossime all'iperpiano stesso (i c.d. punti di frontiera).

Ne segue che tutti i vettori a cui si associa un moltiplicatore di lagrange nullo sono più lontani rispetto all'iperpiano di separazione (i c.d. punti interni) e **non influenzano in alcun modo la definizione dell'iperpiano di ottimo** (w^*, b^*).

I vettori appartenenti ad S sono chiamati vettori di supporto perché graficamente sono quelli più vicini alla superficie di separazione e sembra quasi che la sorreggano.

In particolare il numero di vettori di supporto è tipicamente molto più contenuto rispetto a quello della dimensione del campione di addestramento (anche se per ora questo non ci dice nulla vedremo più avanti come al diminuire del numero dei vettori di supporto migliori la performance di generalizzazione della macchina).

Questa caratteristica delle SVM è potenzialmente sia un punto debole che un pregio del metodo perché anche a fronte di forti variazione nella disposizione di tutti i punti interni l'iperpiano di separazione comunque non si modificherebbe. D'altro canta basta una lieve variazione ad uno qualsiasi dei vettori di supporto per provocare una forte variazione della soluzione.

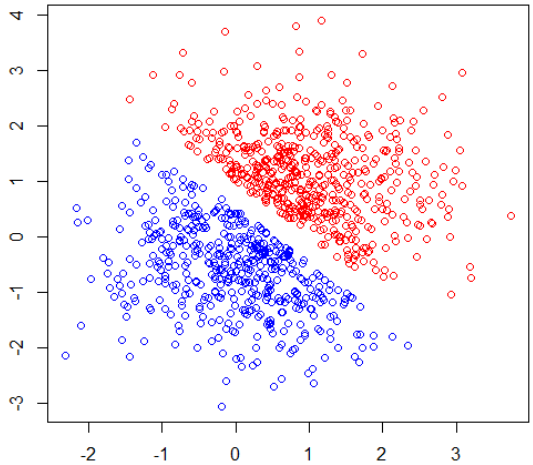
In questo capitolo abbiamo visto come calcolare l'iperpiano di classificazione ottimale (che sappiamo massimizza la performance out of sample posto di avere un errore empirico nullo) e come quest'ultimo sia interamente definito da un subset molto contenuto del training set.

Dobbiamo però ricordare che questo risultato vale unicamente nel caso in cui i punti del training set siano linearmente separabili. Evidentemente dovremo generalizzare il discorso al caso di un training set non linearmente separabile ma prima verrà proposto un breve esempio in R che ripercorra quanto abbiamo fin'ora trattato.

4.Esempio di SVM lineare

Si consideri un data set di 1000 osservazioni $\{(x_i, y_i)\}_{i=1,\dots,1000}$ tali per cui:

- $x \in R^2$: vettore bidimensionale che contiene le caratteristiche dell'osservazione
- $y \in R$: scalare che indica il gruppo di appartenenza dell'osservazione



Per il nostro esempio generiamo le osservazioni da una normale standard e con un semplice accorgimento suddividiamo i punti in due insiemi che siano linearmente separabili.

Come è evidente tracciando una retta è possibile separare correttamente tutti i punti del campione nelle rispettive aree di appartenenza.

Il problema è che ruotando e/o traslando di poco la retta di cui sopra non ci scontreremmo con nessuna osservazione e quindi riusciremmo ancora a separare correttamente l'intero data set.

Se dunque esistono infinite soluzioni al problema di cui sopra, qual è la migliore?

Intuitivamente le nuove osservazioni che cadranno molto vicine al bordo della retta tracciata avranno un maggior rischio di essere classificate in maniera errata quindi una possibile soluzione sta nel ricercare la retta che *massimizza la distanza tra i due gruppi.*

Osserviamo però che ogni gruppo è composto da un certo numero di elementi ed ognuno di questi ha una distanza con la retta quindi è necessario definire una distanza di riferimento da utilizzare nel problema di massimo.

La distanza che andremo a massimizzare sarà quella del punto più vicino alla retta di separazione, questa nella letteratura viene appunto definita *margine di separazione* che formalmente definiamo come

$$\rho(w, b) = \min_{x^i \in A \cup B} \frac{|w^T x^i + b|}{||w||}$$

ove A e B sono i due gruppi di elementi in cui si ripartiscono le osservazioni.

La soluzione del nostro problema sta quindi nella retta in grado di classificare correttamente l'intero data set e che massimizzi il margine di separazione $\rho(w, b)$.

Dopo una serie di passaggi matematici (che non verranno qui indicati perché già trattati in altri capitoli) si può dimostrare che il problema di cui sopra equivale al seguente

$$\min \Gamma(\lambda) = \frac{1}{2}\lambda^T Q\lambda - 1^T\lambda$$

$$sub\begin{cases} y^T\lambda = 0 \\ \lambda \geq 0 \end{cases}$$

ove λ rappresenta il vettore dei moltiplicatori di Lagrange (uno per ogni osservazione disponibile) e Q è una matrice lxl con elemento generico $\{(y_i y_j x_i^T x_j)\}_{i,j=1,\dots,l}$.

Osserviamo che la matrice Q è interamente definita da valori noti perché facenti parte del data set iniziale, l'obiettivo è quindi definire un vettore di ottimo λ che minimizzi l'espressione di cui sopra sotto le due condizioni.

Il primo passo sta quindi nel calcolare la matrice Q, per farlo ci avvaliamo di un semplice ciclo for nidificato che per ogni coppia di osservazioni dovrà fare il prodotto tra le etichette y_i e y_j e moltiplicare il risultato al prodotto interno tra i due vettori x_i e x_j.

```
Q<-matrix(0,m,m)

for (j in 1:m){
        for (i in 1:m) {
                Q[i,j]<- (y[i]*y[j])*(sum(x[i,]*x[j,]))
        }}
```

Definita la matrice Q è possibile calcolare il vettore di ottimo λ avvalendosi di un qualche programma atto alla risoluzione di problemi di ottimizzazione quadratica convessa.

Nel nostro esempio ci avvarremo del pacchetto LowRankQP che permette la risoluzione di problemi del tipo

$$min_\alpha \frac{1}{2}\alpha^T H\alpha + d^T\alpha$$

$$sub\begin{cases} A^T\alpha = b \\ \alpha \in [0, u] \end{cases}$$

Sono evidenti le similitudini con il nostro problema, per poter calibrare la funzione dovremo quindi porre:

- $H = Q$ ove Q è la matrice dei prodotti interni precedentemente calcolata
- $d = (-1)$ vettore di 1000 elementi tutti pari a -1

- $A = y$ la matrice dei vincoli A si riduce al solo vettore delle etichette y
- $b = 0$ il vettore b degenera in uno scalare pari a zero
- $u = 100$ rappresenta il valore massimo che possono assumere i moltiplicatori di lagrange

Notare che nel nostro problema non esiste un limite superiore per i valori assumibili dai moltiplicatori di lagrange e per questo poniamo che u sia pari ad un valore particolarmente alto.

Una volta calibrata la funzione ed eseguita la routine otteniamo il vettore λ^* che minimizza la funzione obiettivo sopra definita. La dimensione del vettore è data dalla numerosità delle osservazioni del training set quindi nel nostro esempio abbiamo a che fare con un vettore di 1000 elementi.

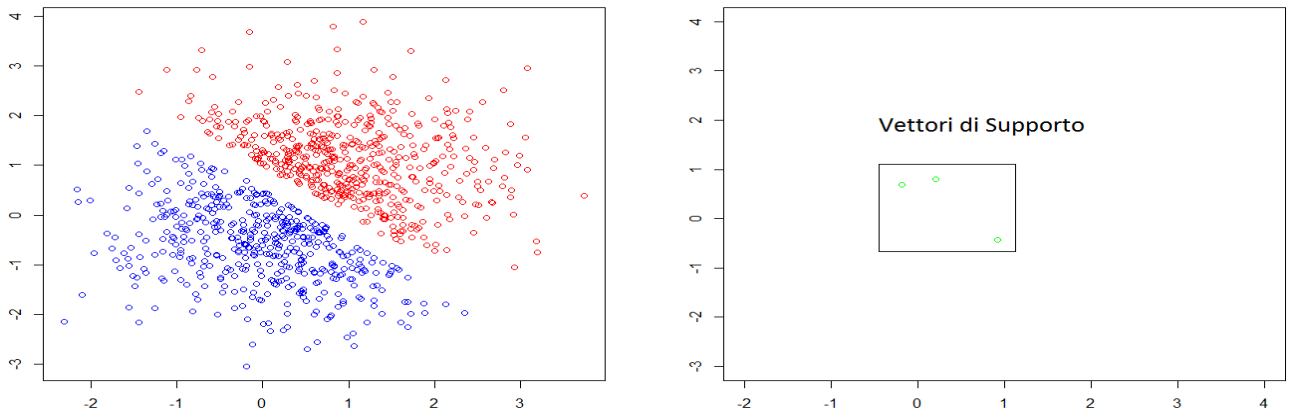
Si può però osservare che la stragrande maggioranza degli elementi di λ^* sono costituiti da valori estremamente piccoli (il valore minimo è nell'ordine di 10^{-16}) quindi inseriamo nello script una semplice funzione volta a pulire il vettore λ^* ponendo pari a zero ogni elemento il cui valore risulti essere minore di 10^{-5}.

Dopo questa operazione possiamo osservare che il vettore di ottimo è quasi interamente costituito da zeri, infatti su 1000 elementi solo 3 risultano non nulli.

Questi sono le osservazioni numero 9, 32 e 238 ed i loro moltiplicatori di lagrange hanno rispettivamente valori 13,621952 15,566550 e 1,944597.

Per essere sicuri di non aver commesso errori utilizziamo gli stessi dati in un diverso risolutore di problemi di programmazione quadratica. Nel nostro esempio ci avvaliamo della funzione ipop(.) presente nel pacchetto kernlab. Il risultato ottenuto è identico a quello calcolato con la funzione LowRankQP.

Graficamente si ottiene



Notare come le tre osservazioni che vantano un moltiplicatore di lagrange non nullo si trovino sulla frontiera dei due gruppi.

Queste tre osservazioni rappresentano i c.d. vettori di supporto ossia gli elementi che di fatto determinano la retta di separazione ottimale secondo il principio di massimizzazione del margine di separazione.

Abbiamo quindi ottenuto che dei 1000 vettori osservati solo 3 sono davvero rilevanti ai nostri fini, infatti se modificassimo (cambiando le loro coordinate) le osservazioni che hanno un moltiplicatore nullo la retta di ottimo non cambierebbe (a patto che la modifica di coordinate non li tramuti in vettori di supporto). D'altro canto modificare anche un solo vettore di supporto provocherebbe rilevanti cambiamenti nella retta di ottimo ottenuta.

Grazie al vettore di ottimo pulito possiamo ricavare l'equazione della retta di separazione con i seguenti risultati

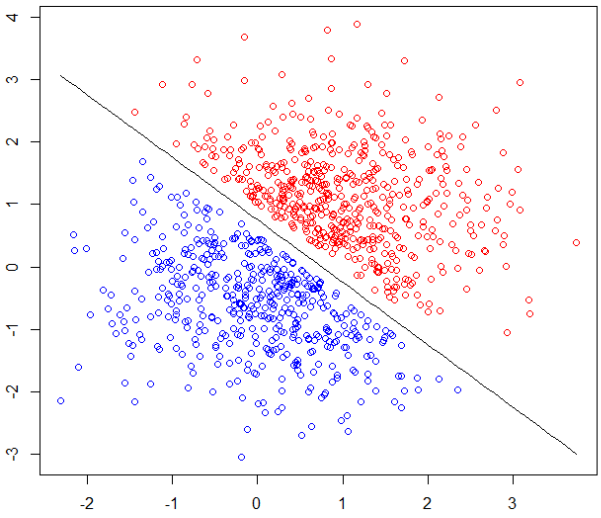
$$w^* = \sum_{i=1}^{l} \lambda_i^* \; y^i x^i = w^* = \sum_{i \in S} \lambda_i^* \; y^i x^i$$

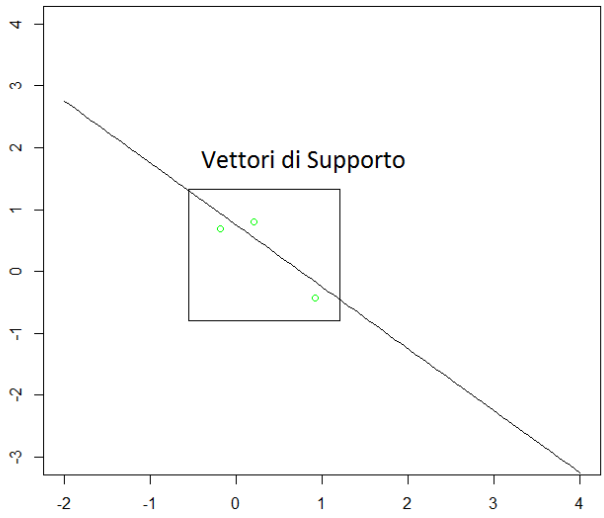
ove S è l'insieme degli elementi di λ^* non nulli.

Per ricavare b^* si osservi che per tutti quei punti che hanno un moltiplicatore di lagrange non nullo vale che

$$\lambda^*_i = 0 \Rightarrow y^i((w^*)^T x^i + b) = 1 \Rightarrow b^* = \frac{1}{y^i} - (w^*)^T x^i \quad \forall i \in S$$

Graficamente otteniamo





Va però ricordato come l'obiettivo del modello sia classificare correttamente le *nuove* osservazioni per le quali non abbiamo informazioni sulla classe di appartenenza.

Al fine di valutare il comportamento del modello appena ottenuto con delle nuove osservazioni è possibile eseguire un test out of sample.

Per farlo si generano altre 200 osservazioni seguendo gli stessi passaggi visti all'inizio e si esegue il confronto con la superficie di separazione ottenuta ricordando che una osservazione viene correttamente classificata quando si verifica

$$y^i(w^T x^i + b) \geq 1$$

ove i parametri w e b sono quelli della retta di separazione sopra indicata.

Al fine di calcolare la percentuale di errore affianchiamo al database un vettore z che abbia argomento pari a 1 in caso di errore e 0 altrimenti.

Sommando tutti gli elementi di z e rapportando per la lunghezza del vettore otteniamo una percentuale di errore del 5%.

Va comunque considerato che i dati considerati nell'esempio sono stati costruiti appositamente per mostrare il funzionamento di una SVM lineare e quindi non deve sorprendere una percentuale di errore così bassa.

5.SVM non lineari

Fin'ora abbiamo sempre assunto che il training set fosse linearmente separabile e quindi che fosse possibile definire un iperpiano in grado di classificare correttamente tutti i punti ottenendo un $R_{emp}(\alpha) = 0$. Riprendendo l'equazione fondamentale si ha quindi

$$R(\alpha_l) \leq R_{emp}(\alpha_l) + \Omega(h, l, \eta) = \Omega(h, l, \eta) \; con \; \alpha_l = (w^*, b^*)$$

Data la relazione inversa che intercorre tra il margine di separazione e la VC dimension l'iperpiano ottimale è quello che rende massimo $\rho(w, b)$.

Il problema è che i dati reali difficilmente sono linearmente separabili e quindi il sistema

$$\begin{cases} w^T x^i + b \geq +1 \;\; \forall x^i \in A \\ w^T x^i + b \leq -1 \;\; \forall x^i \in B \end{cases}$$

nella stragrande maggioranza dei casi non ha soluzione.

La non linearità con cui i punti del training set si posizionano è però osservabile in due diversi livelli

1. bassa non linearità: il data set non è interamente classificabile con un iperpiano a causa di un limitato numero di punti "scomodi" ma se fosse possibile ignorarli si ritornerebbe al caso linearmente separabile
2. alta non linearità: la disposizione dei due gruppi A e B è fortemente non lineare e quindi anche se si ignorassero certe osservazioni non si ritornerebbe al caso linearmente separabile

Queste due casistiche non si escludono a vicenda ed è molto probabile che in una applicazione reale si debbano affrontare entrambi i problemi.

Nei capitoli successivi si affronteranno queste due problematiche separatamente ma per rendere più intuitivo il discorso introduciamo subito l'idea alla base delle due soluzioni.

Rifacendosi alla diseguaglianza fondamentale il caso di bassa non linearità richiama evidentemente il concetto di SRM. In sostanza potremmo scegliere tra

1. utilizzare un set di funzioni non lineare (più complesse e quindi con una maggior VC dimension) così da ottenere un rischio empirico nullo oppure
2. accettare di avere qualche errore in fase di training al fine di utilizzare un set di funzioni lineari che massimizzano il margine

la scelta migliore come sempre sarà quella che ci permetterà di minimizzare la grandezza $R_{emp}(\alpha_l) + \Omega(h, l, \eta)$ nel suo complesso.

Nel caso fortemente non lineare le possibili strategie sono di nuovo due:

1. utilizzare classificatori non lineari cosi da ottenere un rischio empirico nullo oppure
2. tradurre i punti del data set in un nuovo spazio all'interno dei quali risultano essere linearmente separabili.

Come vedremo nella seconda strategia il nuovo spazio che solitamente ha un numero di dimensioni di molto superiore a quelle dello spazio di partenza.

Ne segue che il classificatore lineare avrà una VC dimension molto elevata (essendo in genere pari a $n + 1$) e quindi tale soluzione non sembra offrire una buona performance di generalizzazione.

Come vedremo in questi casi non lineari la VC dimension del classificatore lineare (nel secondo spazio) è in realtà legata al **numero di support vector** che otterremo.

Ne segue che potremmo definire classificatori su spazi con moltissime dimensioni (anche infinite) pur mantenendo una VC dimension contenuta e quindi una buona capacità di performance.

In particolare il classificatore lineare nel secondo spazio vedremo risulterà essere una funzione fortemente non lineare nello spazio di partenza.

5.1 Bassa non linearità

Si assuma che il data set $\{(x^i, y^i)\}_{i=1,\dots,l}$ contenga dei punti che impediscono di classificare correttamente tutte le osservazioni con una funzione lineare.

Come si è già anticipato una strategia risolutiva consiste nel permettere all'iperpiano di classificare in maniera errata alcuni vettori del campione.

Sapendo che l'iperpiano (w, b) classifica correttamente una osservazione (x^i, y^i) quando

$$y^i(w^T x^i + b) \geq 1$$

una osservazione ha due modi per essere classificata in maniera errata:

1. $0 = y^i(w^T x^i + b) < 1$ il punto sta nel corretto iperspazio ma viola il margine
2. $y^i(w^T x^i + b) < 0$ il punto è nel iperspazio sbagliato

Per ogni osservazione introduciamo una variabile scalare di scarto $\xi_i \geq 0$ che rappresenta il possibile tipo di errore che l'iperpiano commette su quel vettore. Il sistema che rappresenta come l'iperpiano classifica il data set diventa quindi

$$\begin{cases} w^T x^i + b \geq +1 - \xi_i \ \ \forall x^i \in A \\ w^T x^i + b \leq -1 + \xi_i \ \ \forall x^i \in B \end{cases}$$

Notare che nel caso A se una osservazione (x^i, y^i) non è correttamente classificata la relativa ξ_i sarà strettamente maggiore di 1, mentre se il vettore viola il margine la variabile scarto assumerà un valore tra zero (escluso) ed uno. Se invece $\xi_i = 0$ il vettore è correttamente classificato.

Il valore delle singole ξ_i da una idea dell'errore (come quantità) che viene ammesso nella classificazione. D'altro canto nella diseguaglianza fondamentale abbiamo

$$R(\alpha_l) \leq R_{emp}(\alpha_l) + \Omega(h, l, \eta)$$

ove $R_{emp}(\alpha_l)$ è una sorta di valore complessivo dell'errore in sample.

Minimizzando la componente a destra della diseguaglianza ottengo dunque un valore $R_{emp}(\alpha_l)$ ottimale ossia una quantità complessiva di errore in sample da ottenere.

Posso quindi definire una soglia massima del numero di errori ammessi nell'addestramento come

$$R^*_{emp}(\alpha_l) = \sum_{i=1}^{l} \xi_i$$

Aggiungendo un parametro moltiplicativo $C > 0$ possiamo pesare l'errore del training set e quindi avere una idea dell'importanza che $\left(\sum_{i=1}^{l} \xi_i\right)$ ha su $\left(\frac{1}{2} w^T w\right)$. Infatti all'aumentare di C è aumenta l'errore che viene ammesso sul training set e viceversa.

Il problema di ottimo diviene

$$\min F(w, \xi) = \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i$$

$$t.c. \begin{cases} y^i (w^T x^i + b) - 1 + \xi_i \geq 0 & \forall i = 1, \dots, l \\ \xi_i \geq 0 & \forall i = 1, \dots l \end{cases}$$

il cui lagrangiano è

$$L(w, b, \xi, \lambda, \mu) = \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i - \sum_{i=1}^{l} \lambda_i (y^i (w^T x^i + b) - 1 + \xi_i) - \sum_{i=1}^{l} \mu_i \xi_i$$

Sfruttando i risultati già visti otteniamo il duale di Wolfe come

$$\max L(w, b, \xi, \lambda, \mu)$$

$$sub \begin{cases} \nabla_w L(w, b, \xi, \lambda, \mu) = w - \sum_{i=1}^{l} \lambda_i y^i x^i = 0 \\ \frac{\partial L(w, b, \xi, \lambda, \mu)}{\partial b} = \sum_{i=1}^{l} \lambda_i \ y^i = 0 \\ \frac{\partial L(w, b, \xi, \lambda, \mu)}{\partial \xi_i} = C - \lambda_i - \mu_i = 0 \ \forall i = 1, \dots, l \\ \lambda \geq 0 \\ \mu \geq 0 \end{cases}$$

sostituendo il risultato della prima condizione nella funzione obiettivo si ricava

$$\min \Gamma(\lambda) = \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y^i y^j (x^i)^T x^j \lambda_i \lambda_j - \sum_{i=1}^{l} \lambda_i$$

$$sub \begin{cases} \sum_{i=1}^{l} \lambda_i \ y^i = 0 \\ 0 \leq \lambda_i \leq C \ \ \forall i = 1, \dots, l \end{cases}$$

che è esattamente identico a quello visto nel caso linearmente separabile salvo per la seconda condizione che impone un upper bound a λ_i che deriva da

$$\begin{cases} C - \lambda_i - \mu_i = 0 \ \forall i = 1, \dots, l \\ \lambda \geq 0 \\ \mu \geq 0 \end{cases}$$

Si osserva che:

1. il vettore w^* è determinato da

$$w^* = \sum_{i=1}^{l} \lambda_i{}^* y^i x^i$$

2. per le condizioni di KKT vale che

$$\begin{cases} \lambda_i{}^* \left[y^i \left(w^{*T} x^i + b \right) - 1 + \xi_i \right] \geq 0 \ \forall i = 1, \dots, l \\ \mu_i{}^* \xi_i{}^* = 0 \qquad \forall i = 1, \dots, l \end{cases}$$

da cui si ricava b^*.

Notare come l'impostazione del problema e la sua risoluzione risultino estremamente simili a quelli visti nel caso lineare "puro". Effettivamente in letteratura questi tipi di classificatori vengono chiamati soft margin per il fatto che si definiscono quasi interamente come un normale classificatore lineare a massimo margine ma quest'ultimo può essere di fatto violato da un numero limitato di osservazioni.

Per questo motivo il primo classificatore che abbiamo visto (che non ammetteva errori dato che era in grado di classificare correttamente l'intero training set) viene anche chiamato come classificatore hard margin.

5.2 Forte non linearità

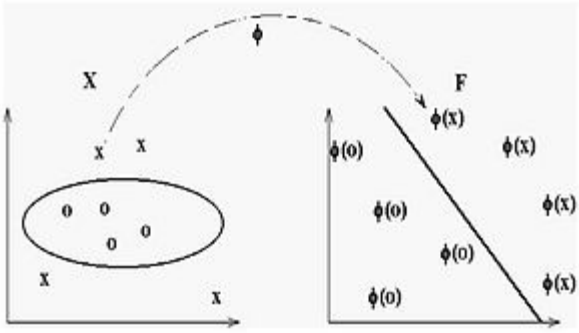
Quando i vettori del data set sono disposti nei due gruppi A e B in maniera fortemente non lineare non è sufficiente utilizzare un iperpiano che ammetta degli errori nella classificazione.

Se fosse però possibile tradurre le nostre osservazioni in un nuovo spazio in cui queste sono linearmente separabili allora potremmo applicare in questo secondo "ambiente" quanto abbiamo visto nei due capitoli precedenti per poi ritradurre il risultato nello spazio di partenza.

Questi due spazi in cui sono definiti i punti vengono rispettivamente chiamati

1. Input space: spazio in cui sono definite le osservazioni campionarie
2. Feature space: spazio in cui sono "tradotte" le osservazioni campionarie di partenza

questa operazione è ben rappresentata dal seguente (celebre) grafico



L'operazione di codifica (mappatura) dei vettori x^i avviene tramite la funzione $\phi: R^n \to R^N$ dove tipicamente la dimensione N del feature space è maggiore di quella dello spazio di partenza.

Le coordinate dei vettori verranno definite sinteticamente come $z^i = \phi(x^i)$ quindi i punti del data set diverranno $\{(z^i, y^i)\}_{i=1,\dots l}$.

Il nostro problema di ottimizzazione si traduce in

$$\min \Gamma(\lambda) = \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y^i y^j \lambda_i \lambda_j (z^i)^T z^j - \sum_{i=1}^{l} \lambda_i$$

$$sub \begin{cases} \sum_{i=1}^{l} \lambda_i \ y^i = 0 \\ \lambda_i \geq 0 \ \forall i = 1, \dots, l \end{cases}$$

Fin'ora non sono state poste condizioni particolari su come eseguire la mappatura dall'input al feature space. E' però abbastanza intuitivo osservare che se ci trasferiamo in uno spazio con un numero di dimensioni sufficientemente alto qualsiasi problema di partenza non lineare è traducibile in un equivalente lineare.

Questa però introduce una nuova problematica sul piano pratico data dal fatto che all'aumentare della complessità del feature space aumenta anche quella computazionale nella risoluzione del nostro problema di massimo.

Ciò è dato dal fatto di avere un elevato numero di coordinate con cui si definisce ogni punto z^i. Al limite potrebbe capitare di utilizzare spazi Z con infinite dimensioni e quindi z^i risulta essere definito come un vettore di infinite coordinate rendendo impossibile ogni approccio di calcolo.

Dalla notazione della funzione $\Gamma(\lambda)$ osserviamo però che per poter risolvere il nostro problema di fatto **non serve tradurre i vettori x^i negli equivalenti vettori z^i** del feature space ma basterebbe conoscere i prodotti interni per tutte le coppie $i \neq j$ per $i = 1, \dots, l$.

Ciò è confermato dal fatto che la funzione decisionale $g(z) = sgn(w^{*T} z + b^*)$ dipende solo dai prodotti interni tra i vettori tradotti nel feature space (per verificare ciò basta sostituire la definizione di w^* e b^*).

Il prodotto interno tra le immagini dei vettori $x^i \in R^n$ nel feature space definisce la così detta funzione kernel

$$k(x^i, x^j) = \langle \phi(x^i), \phi(x^j) \rangle = \langle z^i, z^j \rangle$$

quindi la funzione $\Gamma(\lambda)$ diventa

$$\Gamma(\lambda) = \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y^i y^j \lambda_i \lambda_j k(x^i, x^j) - \sum_{i=1}^{l} \lambda_i$$

Nel ragionamento di cui sopra vi è una però un problema: come è possibile "conoscere" solamente il risultato di un prodotto interno tra vettori senza conoscerne le coordinate?

In sostanza:

$$come\ si\ fa\ a\ conoscere\ \langle z^i, z^j \rangle\ senza\ sapere\ cosa\ sono\ z^i\ e\ z^j?$$

per rispondere a questo punto in maniera intuitiva consideriamo un semplice esempio.

Dato uno spazio di input bidimensionale si considerino due vettori $x = (x_1, x_2)$ e $y = (y_1, y_2)$ ed una trasformazione nello spazio R^6 definita come

$$\phi(x) = \begin{pmatrix} 1 \\ x_1^2 \\ x_2^2 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \\ \sqrt{2}x_1x_2 \end{pmatrix} \quad e \quad \phi(y) = \begin{pmatrix} 1 \\ y_1^2 \\ y_2^2 \\ \sqrt{2}y_1 \\ \sqrt{2}y_2 \\ \sqrt{2}y_1y_2 \end{pmatrix}$$

Il prodotto interno tra i due vettori trasformati risulta essere

$$\langle \phi(x^i), \phi(x^j) \rangle = 1 + x_1^2y_1^2 + x_2^2y_2^2 + 2x_1y_1 + 2x_2y_2 + 2x_1x_2y_1y_2$$

Si consideri ora la funzione $f(x, y) = (1 + x^Ty)^2$ si osserva che tale funzione *da lo stesso risultato del prodotto interno tra i due punti mappati nello spazio R^6 senza tradurre le coordinate dei vettori in quelle del feature space*.

$$k(x, y) = \langle \phi(x^i), \phi(x^j) \rangle = (1 + x^Ty)^2$$

La funzione nei vettori x e y corrisponde ad un prodotto interno tra le immagini dei vettori in un "qualche" feature space.

E' quindi possibile calcolare i prodotti interni tra i vettori z^i e z^j senza dover necessariamente calcolare le coordinate dei punti nel feature space.

Dal punto di vista computazionale questa strategia ha il vantaggio di dover richiedere uno "sforzo" di calcolo identico quale che sia la dimensione dello spazio caratteristica perché l'input informativo che sarà dato all'algoritmo di apprendimento è costituito dalla matrice $l \times l$ dei prodotti interni cioè da l^2 valori scalari.

Evidentemente una nuova problematica sorge a questo punto: data una qualche funzione in x^i e x^j come si fa a sapere che questa corrisponde al prodotto interno tra i vettori mappati in un qualche feature space?

Intuitivamente sarà necessario verificare una serie di proprietà che rendono la funzione di mappatura "valida" a rappresentare un kernel. Vista la complessità del problema dedicheremo il prossimo capitolo interamente a tale problema.

5.3 Validità di un kernel

In questo capito dobbiamo capire quali sono le proprietà fondamentali affinché una determinata funzione dei vettori definiti nell'input space corrisponda ad un prodotto interno in un qualche spazio delle caratteristiche.

Ricordando che il nostro problema ora è definito come

$$\min \Gamma(\lambda) = \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y^i y^j \lambda_i \lambda_j (z^i)^T z^j - \sum_{i=1}^{l} \lambda_i$$

$$sub \begin{cases} \sum_{i=1}^{l} \lambda_i \ y^i = 0 \\ \lambda_i \geq 0 \ \ \forall i = 1, \dots, l \end{cases}$$

dovremo concentrarci sulla prima doppia sommatoria in cui effettivamente compaiono le z, di questa in particolare dovremo studiare $(z^i)^T z^j$ che può essere sinteticamente riscritta in notazione vettoriale.

Sia quindi G la matrice quadrata $l \times l$ che contiene i valori della funzione kernel per ogni possibile coppia di vettori x^i e x^j con $i, j = 1, \dots, l$.

$$G = \{k(x^i, x^j)\} = \begin{bmatrix} k(x^1, x^1) & \dots & k(x^1, x^l) \\ \dots & \dots & \dots \\ k(x^l, x^1) & \dots & k(x^l, x^l) \end{bmatrix} = \begin{bmatrix} \langle \phi(x^1), \phi(x^1) \rangle & \dots & \langle \phi(x^1), \phi(x^l) \rangle \\ \dots & \dots & \dots \\ \langle \phi(x^l), \phi(x^1) \rangle & \dots & \langle \phi(x^l), \phi(x^l) \rangle \end{bmatrix}$$

Dato che la funzione kernel si definisce come un prodotto interno ne deve vantare le proprietà:

1. $\langle x, y \rangle = \langle y, x \rangle$
2. $\langle x_1 + x_2, y \rangle = \langle x_1, y \rangle + \langle x_2, y \rangle$
3. $\langle \lambda x, y \rangle = \lambda \langle x, y \rangle$
4. $\langle x, x \rangle \geq 0 \ con \ \langle x, x \rangle = 0 \ \Leftrightarrow x = 0$

Dalla definizione di kernel come prodotto interno possiamo quindi ricavare due caratteristiche che deve avere la matrice G

1. se il kernel è una funzione simmetrica allora anche la matrice G risulta simmetrica
2. G deve anche essere semidefinita positiva

Infatti si osservi che

$$v^T G v = \sum_{i=1}^{l}\sum_{j=1}^{l} v_i v_j k(x_i, x_j) = \sum_{i=1}^{l}\sum_{j=1}^{l} v_i v_j \langle \phi(x_i), \phi(x_j)\rangle = \sum_{i=1}^{l}\sum_{j=1}^{l} \langle v_i \phi(x_i), v_j \phi(x_j)\rangle$$

$$= \langle \sum_{i=1}^{l} v_i \phi(x_i), \sum_{j=1}^{l} v_j \phi(x_j)\rangle = \langle z, z\rangle = \|z\|^2 \geq 0$$

$$v^T G v \geq 0 \; \forall v \in R^l.$$

quindi la matrice G risulta essere simmetrica e semi definita positiva.

Queste due caratteristiche della G derivano essenzialmente dal fatto che il kernel sia un prodotto interno. Questo prodotto interno è però fatto tra due trasformate dei punti dell'input space quindi ora dobbiamo capire quali proprietà deve vantare la funzione di mappatura ϕ.

Innanzi tutto osserviamo che se $k(x^i, x^j) = \langle \phi(x^i), \phi(x^j)\rangle$ il feature space in cui si definiscono i punti $\phi(x)$ dovrà essere almeno uno spazio metrico in cui è definita l'operazione di prodotto interno (altrimenti non potremmo calcolare il kernel).

Supponiamo che il feature space sia definito come lo spazio di funzioni $k(*, x)$ allora la mappatura ϕ non farà altro che assegnare ad ogni punto del input space una funzione $k(*, x)$ ove $*$ ne indica l'argomento.

Per prima cosa dobbiamo verificare che in questo spazio sia definita l'operazione di prodotto interno, per farlo "allarghiamo" lo spazio di sovra definito assumendo che questo contenga anche funzioni che sono trasformazioni lineari di $k(*, x)$. Definiamo quindi lo spazio F come

$$F = \left\{ \sum_{i=1}^{l} \alpha_i k(x_i, *) : l \in N, x_i \in X, \alpha_i \in R \; per \; i = 1, \dots l \right\}$$

Prendiamo ora due elementi generici di F che chiamiamo f e g

$$f = \sum_{i=1}^{n} \alpha_i k(x_i, *)$$

$$g = \sum_{j=1}^{m} \beta_j k(x_j, *)$$

E' immediato verificare che la somma $f + g$ ed il prodotto $f * g$ definiscono un elemento ancora riconducibile allo spazio F quindi possiamo affermare che quest'ultimo è lineare (ogni elemento dello spazio è ottenibile come combinazione lineare dei suoi punti).

Dobbiamo ora verificare che in F sia anche definito il prodotto interno, per farlo innanzitutto "proviamo" ad eseguire il conto con in nostri due punti generici

$$< f, g > = < \sum_{i=1}^{n} \alpha_i k(x_i, *), \sum_{j=1}^{m} \beta_j k(x_j, *) > = \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j k(x_i, x_j)$$

se riusciamo a dimostrare che quanto scritto sopra è effettivamente un prodotto interno abbiamo raggiunto lo scopo. Innanzitutto dobbiamo verificare che l'operazione di cui sopra sia ben definita cioè che valga per ogni generica coppia di punti in F.

$$< f, g > = \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j k(x_i, x_j) = \sum_{j=1}^{m} \beta_j f(x_j) = \sum_{i=1}^{n} \alpha_i g(x_i)$$

La relazione

$$< f, g > = \sum_{j=1}^{m} \beta_j f(x_j)$$

ci dice in sostanza che il questa funzione di f e g non dipende da α_i e x_i mentre la seconda relazione

$$< f, g > = \sum_{i=1}^{n} \alpha_i g(x_i)$$

ci conferma che $< f, g >$ non dipende neanche da β_j e x_j quindi la funzione è ben definita su tutto F.

Dobbiamo verificare che la funzione $< f, g >$ abbia tutte le proprietà del prodotto interno

$$< f, g > = < g, f > \ in\ quanto\ k(x_i, x_j) = k(x_j, x_i)$$

$$< af, g >= \sum_{i=1}^{n}\sum_{j=1}^{m} a\,\alpha_i\,\beta_j k(x_i, x_j) = a\sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i\,\beta_j k(x_i, x_j) = a < f, g >$$

$$< f + a, g >= \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i\,\beta_j k(x_i, x_j) + \sum_{k=1}^{n}\sum_{j=1}^{m} a\,\beta_j k(x_k, x_j) =< f, g > +< a, g >$$

$$< f, f >= \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i\,\alpha_j k(x_i, x_j) = \alpha^T G\alpha \geq 0\ per\ le\ riflessioni\ fatte\ su\ G$$

Visto che tutte le proprietà del prodotto interno sono state verificate possiamo concludere che lo spazio delle feature definito come

$$F = \left\{\sum_{i=1}^{l} \alpha_i k(x_i, *) : l \in N, x_i \in X, \alpha_i \in R\ per\ i = 1, \dots l\right\}$$

definisce al suo interno un prodotto interno.

L'ultima cosa da definire è la mappatura $\phi: R^n \to R^N$ che traduce i punti dall'input al feature space appena definito.

Si consideri l'elemento ϕ in F definito come $\phi(x_i) = k(x_i, *)$, comunque si scelgano due vettori $x_i, x_j \in X$ dalla definizione della funzione $< f, g >$ si ottiene che

$$< \phi(x_i), \phi(x_j) >=< k(x_i, *), k(x_j, *) >$$

e ricordando il risultato

$$< f, g >= \sum_{i=1}^{n} \alpha_i g(x_i) = \sum_{j=1}^{m} \beta_j f(x_j)$$

che equivale a dire

$$< f, g >= f(g(x_i)) = g(f(x_j))$$

otteniamo

$$< \phi(x_i), \phi(x_j) >=< k(x_i, *), k(x_j, *) >= k(x_i, x_j) = k(x_j, x_i)$$

dunque si è dimostrato che k è un kernel.

Grazie alla teoria sviluppata in questo capitolo possiamo dire con certezza quando una funzione definita unicamente nello spazio di input corrisponde ad un prodotto interno in un qualche feature space che non dobbiamo necessariamente conoscere.

Ritornando all'esempio con cui abbiamo introdotto la problematica

$$k(x, y) = \langle \phi(x), \phi(y) \rangle = (1 + x^T y)^2$$

in sostanza ora possiamo sapere se $(1 + x^T y)^2$ corrisponde ad un qualche prodotto interno senza conoscere lo spazio R^6 in cui era definito e la mappatura $\phi(x)$.

E' bene osservare che questa possibilità esiste solo sul piano teorico perché nella pratica la effettiva verifica della validità di una funzione come kernel è un compito assai complesso. Fortunatamente in letteratura esistono una serie di funzioni per le quali si è già dimostrata la validità come kernel.

Tra questi abbiamo kernel

1. Lineare $k(x, y) = x^T y$
2. Polinomiale $k(x, y) = (ax^T y + b)^p$
3. Gaussiano $k(x, y) = \exp\left(\frac{\|x-y\|^2}{2\sigma^2}\right)$

come il lettore avrà intuito la questione qui è che non esiste una regola generale che mi dica quale kernel utilizzare a fronte del problema che devo affrontare.

Per tale motivo a volte è preferibile cercare di implementare una qualche tecnica di feature reduction così da ottenere un nuovo input space in cui tutti i punti sono linearmente separabili. Su questo si applica quindi una SVM lineare evitando la scelta soggettiva di quale kernel scegliere.

5.4 Formula risolutiva nel caso non lineare

Fatti questi ragionamenti possiamo ora ritornare al nostro problema di partenza e vedere come risolverlo nel caso in cui non si considerino le informazioni x ma delle loro trasformate $\phi(x)$

$$\min F(w,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i$$

$$t.c.\begin{cases} y^i(w^T\phi(x^i) + b) - 1 + \xi_i \geq 0 & \forall i = 1,\dots,l \\ \xi_i \geq 0 & \forall i = 1,\dots l \end{cases}$$

Questo avrà come duale di Wolfe il problema

$$\min \Gamma(\lambda) = \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y^i y^j (\phi(x^i))^T\phi(x^j)\lambda_i\lambda_j - \sum_{i=1}^{l}\lambda_i$$

$$sub\begin{cases} \sum_{i=1}^{l}\lambda_i\ y^i = 0 \\ 0 \leq \lambda_i \leq C\ \ \forall i = 1,\dots,l \end{cases}$$

Con passaggi analoghi a quelli visti nel caso lineare otteniamo

$$w^* = \sum_{i=1}^{l}\lambda_i{}^*\, y^i\, \phi(x^i)$$

noto w^* lo scalare b^* è ottenuto come

$$y^i\left(\sum_{j=1}^{l}\lambda_j{}^*\, y^j(\phi(x^i))^T\phi(x^j) + b^*\right) - 1 = y^i\left(\sum_{j=1}^{l}\lambda_j{}^*\, y^j k(x^i,x^j) + b^*\right) - 1 = 0$$

e quindi la funzione decisionale nel processo di classificazione risulterà

$$f(x) = sgn(\,(w^*)^T\phi(x) + b) = sgn\left(\sum_{j=1}^{l}\lambda_j{}^*\, y^j k(x^i,x^j) + b^*\right)$$

È interessante notare quest'ultimo risultato perché evidenzia come la superficie di separazione sia comunque lineare nello spazio delle caratteristiche mentre nell'input space risulta essere non lineare.

E’ bene inoltre evidenziare che i vettori di supporto ottenuti così come il margine di separazione sono stati ricavati nello spazio F quindi la distanza tra i vettori di supporto ed il classificatore nello spazio di input di fatto non rappresenta il margine di separazione.

Quest’ultimo infatti è stato calcolato in uno spazio in cui le osservazioni erano linearmente separabili e quindi è errato pensare di ritradurlo in termini dell’input space di partenza.

5.5 Capacità di generalizzazione di una SVM

Alla fine del primo capitolo di questa seconda parte ci siamo chiesti come fosse possibile sfruttare la teoria dell'apprendimento statistico per valutare la capacità di generalizzazione di un classificatore che massimizza il margine di separazione

Per questo motivo abbiamo ripreso la nostra diseguaglianza fondamentale

$$R(\alpha_l) \leq R_{emp}(\alpha_l) + \Omega(h, l, \eta)$$

ed abbiamo notato che se l'iperpiano ottimale classifica correttamente l'intero data set allora il rischio empirico sarà nullo. Ne segue che il limite massimo del rischio di generalizzazione è interamente definito da $\Omega(h, l, \eta)$ che si dimostra inversamente proporzionale alla VC dimension.

Si dimostra che la VC dimension di un classificatore lineare è pari ad n + 1 dove n è il numero di dimensioni dello spazio in cui è definito l'iperpiano di separazione.

Intuiamo quindi il problema che si pone in ambito non lineare: abbiamo visto che nel caso di problemi non linearmente separabili il "trucco" sta nel tradurre le osservazioni campionarie in un nuovo spazio Z ove queste sono classificabili da un iperpiano di separazione.

Questo è sicuramente un pregio della metodologia perché evita di utilizzare un set di ipotesi Λ più complesso di quello lineare per il quale potrebbe essere non agevolo calcolare h.

D'altro lo spazio Z ha tipicamente un numero di dimensioni di molto superiore a quello campionario di partenza (addirittura grazie al kernel trick è possibile utilizzare spazi con infinite dimensioni) e ciò significa che il classificatore lineare definito in Z avrà una h molto elevata.

Se anche avessimo un iperpiano di separazione che in Z classifica correttamente tutti i punti (ottenendo un rischio empirico nullo) la VC dimension della soluzione sarebbe comunque così elevata che alla fine la performance di generalizzazione sarebbe poco soddisfacente.

In realtà non dobbiamo disperare troppo di questo perché è possibile dimostrare con semplici passaggi come il valore atteso dell'errore $R(\alpha_l)$ sia superiormente limitato da una grandezza funzione del numero di support vector ottenuti che sappiamo essere una frazione molto contenuta delle l osservazioni campionarie.

Per dimostrare ciò abbiamo prima bisogno di presentare una tecnica di stima dell'errore out of sample conosciuta nell'ambito del machine learning come la procedura "leave one out".

Si supponga di avere l osservazioni campionarie definite sinteticamente come $(z_1, \dots, z_l)$ ove l è il numero complessivo di osservazioni campionarie.

Definiamo con $Q(z, \alpha_l)$ l'iperpiano di separazione calcolato sull'intera informazione campionaria che rende minimo il rischio empirico

$$R_{emp}(\alpha) = \sum_{i=1}^{l} Q(z_i, \alpha)$$

Supponiamo ora di eseguire la seguente procedura

1. Dall'informazione campionaria escludiamo la prima osservazione z_1 e ricalcoliamo il classificatore lineare che minimizza il rischio empirico calcolato solo sugli $l-1$ vettori rimanenti
2. Definiamo con $Q(z, \alpha_{l-1}|z_1)$ il classificatore ottenuto utilizzando le sole $l-1$ osservazioni campionarie rimanenti
3. Sulla base di $Q(z, \alpha_{l-1}|z_1)$ calcoliamo l'errore di classificazione che questo commette sull'osservazione z_1 **che non è stata utilizzata nell'addestramento** ottenendo un errore che indichiamo con $Q(z_1, \alpha_{l-1}|z_1)$
4. Reinseriamo la z_1 nel database e togliamo la z_2 e rieseguiamo quanto indicato sopra

Questa procedura viene fatta per tutti gli l punti del campione, in questo modo possiamo utilizzare tutti gli l vettori dell'informazione campionaria sia per l'addestramento che per il test out of sample.

L'errore complessivo out of sample che otteniamo nelle l iterazioni viene definito come

$$L(z_1, \dots, z_l) = \sum_{i=1}^{l} Q(z_i, \alpha_{l-1}|z_i)$$

possiamo quindi calcolare l'expected di L considerando la probabilità congiunta di aver osservato il campione come

$$E[L(z_1, \dots, z_l)] = \int \sum_{i=1}^{l+1} Q(z_i, \alpha_{l-1}|z_i)\, dP(z_1) \dots dP(z_{l+1})$$

"selezionando" l'i −esimo elemento avremo

$$\int \left[\sum_{i=1}^{l+1} \left(\int Q(z_i, \alpha_{l-1} | z_i) dP(z_i) \right) \right] dP(z_1) \dots dP(z_{i-1}) dP(z_{i+1}) \dots dP(z_l)$$

ma essendo

$$\int Q(z_i, \alpha_{l-1} | z_i) dP(z_i) = R(\alpha_l | z_i)$$

otteniamo

$$\int \left[\sum_{i=1}^{l+1} R(\alpha_l | z_i) \right] dP(z_1) \dots dP(z_{i-1}) dP(z_{i+1}) \dots dP(z_l) = E\left[\sum_{i=1}^{l+1} R(\alpha_l | z_i) \right] = (l+1) E[R(\alpha_l)]$$

In definitiva otteniamo

$$E[L(z_1, \dots, z_l)] = (l+1) E[R(\alpha_l)]$$

e quindi

$$E\left[\frac{L(z_1, \dots, z_l)}{l+1} \right] = E[R(\alpha_l)]$$

Quindi l'expected dell'errore leave one out è pari all'expected dell'errore out of sample che si otterrebbe con la funzione α_l che rende minimo l'errore di classificazione.

Si ricordi ora un passaggio fondamentale visto nel capitolo sui SVM lineari: l'iperpiano di classificazione che massimizza il margine è interamente definito dai vettori di supporto.

Dunque tutte le osservazioni che non sono dei support vector non influenzano in alcun modo la soluzione finale, in pratica se sposto o **elimino** un punto che non è un SV il classificatore non cambia.

Cosa significa questo alla luce della metodologia del leave one out? Supponiamo che x_i sia un'osservazione campionaria ma non un SV allora togliendo quel vettore dal training set l'iperpiano di ottimo non cambia. Se l'iperpiano non cambia allora classificherà correttamente quell'osservazione.

In generale una procedura leave one out riconosce correttamente tutti i punti che non sono vettori di supporto perché il classificatore sarà addestrato sulle rimanenti $l-1$ osservazioni che contengono tutti i SV.

L'errore complessivo $L(z_1, \dots, z_l)$ sarà necessariamente minore del numero di vettori di supporto che definiscono la soluzione finale

$$L(z_1, \dots, z_l) \leq \#SV$$

ne segue che

$$E[R(\alpha_l)] = E\left[\frac{L(z_1, \dots, z_l)}{l+1}\right] \leq E\left[\frac{\#SV}{l+1}\right]$$

Con questi semplici passaggi abbiamo dimostrato come la capacità di generalizzazione sia superiormente limitata da una grandezza funzione del numero di vettori di supporto e della dimensione campionaria.

Come ci aspettavamo al divergere della dimensione campionaria l'upper bound dell'expected $R(\alpha_l)$ si azzera, d'altro canto un contenuto numero di SV garantisce comunque un buon vincolo di generalizzazione.

Come vedremo nel capitolo sulla gestione delle feature tipicamente il numero di support vector tende a salire all'aumentare della complessità dello spazio Z e quindi della funzione kernel utilizzata.

Per questo motivo a volte è preferibile applicare opportune tecniche di feature selection e featur extraction al database **prima** di addestrare il classificatore. In questo modo ci si augura di ottenere un problema che nello spazio descritto dalle feature "filtrare" sia linearmente separabile o al più con un contenuto grado di non linearità.

Conclusione della seconda parte

Con questa seconda parte abbiamo visto una applicazione della teoria dello statistical learning per definire un classificatore che non solo possa ottenere una buona performance in fase di addestramento ma che effettivamente riesca a generalizzare il fenomeno.

In estrema sintesi potremmo dire che la logica di fondo dei SVM è quella di individuare un classificatore lineare che sia in grado di massimizzare il cd margine di separazione ossia la distanza tra il separatore e i vettori più vicini a quest'ultimo (detti di frontiera).

Il fatto di massimizzare il margine di separazione ha senso sia ad un livello intuitivo perché a causa del disturbo presente nel campione i punti di frontiera sono quelli che più facilmente sono esposti ad un rischio di classificazione errata.

D'altro canto abbiamo visto come il margine di separazione sia inversamente legato alla VC dimension del classificatore e che quindi all'aumentare del margine è possibile ottenere un minor upper bound del rischio di errore in generale.

Questo classificatore ottimale si dimostra essere definito unicamente da un subset degli l punti utilizzati in fase di addestramento chiamati appunto vettori di supporto.

Passando poi al caso non lineare si è visto come i SVM utilizzino comunque un classificatore di tipo lineare previa la mappatura dell'informazione campionaria in uno spazio Z (chiamato feature space) nel quale le osservazioni risultano linearmente separabili. Il classificatore addestrato in Z risulta essere una equazione fortemente non lineare una volta ritradotto nello spazio di partenza X.

In particolare abbiamo visto che per poter sfruttare la logica appena descritta non è necessario definire effettivamente lo spazio Z perché ai nostri fini basterebbe poter conoscere i prodotti interni tra tutte le coppie (x_i, x_j) tradotte nel loro equivalente (z_i, z_j).

Questo viene ottenuto tramite le c.d. funzioni kernel ossia funzioni definite nei soli valori dell'input space che però equivalgono ad un prodotto interno in un qualche spazio Z. Tale "trucco" consentirebbe addirittura di utilizzare spazi Z con infinite dimensioni sopportando lo stesso sforzo computazionale dato dal fatto che alla fine i conti sono fatti unicamente nello spazio X.

Infine abbiamo dimostrato come l'errore atteso out of sample di un classificatore definito con la logica SVM sia superiormente limitato da una grandezza funzione del numero di SV (che sappiamo essere tipicamente molto contenuto rispetto alla dimensione campionaria) ed il numero di osservazioni disponibili.

Evidentemente queste poche pagine non possono descrivere in maniera esaustiva una metodologia complessa come quella delle SVM e che recentemente ha avuto un cosi forte sviluppo in moltissimi campi applicativi.

Ciò nonostante quanto scritto da una idea della logica fondamentale alla base del metodo e ci permetterà di utilizzarla in una futura applicazione con un adeguato grado di coscienza.

Nella prossima parte saranno presentati due metodi alternativi alle SVM ma facenti comunque parte della famiglia del machine learning: le reti neuronali e gli alberi decisionali.

Le prime saranno interessanti perché si basano su una logica di addestramento che è esattamente l'opposto di quella presentata nelle SVM, i secondi per presentare un metodo di mahcine learning meno potente rispetto a questi due ma che vanta la caratteristica di essere una "white box".

Terza Parte

1.Reti Neuronali

In come anticipato nella conclusione della seconda parte in questo capitolo verrà trattata un'altra metodologia di apprendimento automatico chiamata appunto rete neurale perché come vedremo si ispira direttamente al funzionamento del cervello umano.

Questa metodologia è storicamente nata prima delle SVM ed infatti la sua modellizzazione non si riferisce direttamente alla teoria del'apprendimento automatico sviluppata dal matematico sovietico Vladimir Vapnik (non a caso padre delle stesse SVM).

Ciò nonostante sarà interessante fare un confronto tra le due metodologie sulla base dell'equazione fondamentale che definisce un vincolo superiore in probabilità del così detto errore out of sample.

Infatti una cosa che accomuna SVM e reti neuronali è l'essere una black box e la tendenza a sovra-adattarsi ai dati campionari rendendo quindi fondamentale la possibilità di relazionare l'errore in sample con quello generico definito sull'intero spazio campionario.

Per rendere la presentazione dell'argomento semplice ed immediata tratteremo l'argomento su tre punti

- Il primo darà un'idea generica del funzionamento delle RN in maniera discorsiva
- Nel secondo si presenterà un esempio step by step in cui si mostrerà il funzionamento di una semplice RN ed una delle metodologie più semplici per il suo addestramento
- Nel terzo si faranno alcune considerazioni sul confronto tra RN e SVM

1.1 Introduzione ai concetti chiave

Le reti neurali artificiali sono uno strumento di calcolo volto all'apprendimento di regole (pattern) del tutto svincolato da ipotesi di tipo semiparametriche o parametriche.

Ispirandosi al funzionamento delle reti neurali naturali questo modello matematico si basa su un elemento fondamentale (il neurone) caratterizzato da un comportamento binario estremamente semplice di tipo acceso/spento.

Un singolo neuroni è osservabile come un nodo che riceve informazione da uno o più neuroni. In base all'informazione che riceve il neurone può o meno attivarsi e quindi mandare a sua volta un segnale allo strato successivo.

Dato che non tutte le informazioni che riceve dallo strato precedente in input hanno la stessa importanza ogni segnale che viene inviato è associato ad un peso w.

Quindi l'informazione complessiva ricevuta in input è data da una combinazione lineare tra il segnale dato dal neurone (ossia il fatto che questi si sia acceso o meno) e i pesi (ossia la rilevanza del fatto che un particolare neurone si sia o meno attivato).

Per rappresentare matematicamente l'attivazione del neurone in base alla rilevanza dell'informazione ricevuta il segnale complessivo ottenuto in input viene confrontato con un valore soglia tipico dello specifico neurone che stiamo analizzando. Se la somma dei segnali pesati supera il valore sogli il neurone si attiva e trasferisce questa informazione allo strato successivo, altrimenti rimarrà "spento".

Notare che anche il caso di neurone "spento" è una informazione perché significa che quanto ricevuto dagli strati precedenti non è sufficiente affinché il nodo si attivi.

L'aggregazione di queste unita fondamentali permette alla rete di svolgere funzioni altamente complesse.

Schematicamente le reti neuronali si costituiscono tipicamente su tre livelli:

1. input: costituito da uno strato di neuroni il cui compito è assimilare l'informazione dall'esterno e trasferirla agli strati successivi
2. calcolo: costituito da uno o più strati di neuroni (detti nascosti) all'interno dei quali avviene la trattazione dell'informazione ricevuta dallo strato precedente
3. output: strato di neuroni che fornisce la soluzione

I collegamenti tra i neuroni possono essere di natura più o meno complessa, il caso più semplice è quello di tipo feed forward in cui i neuroni di uno strato i possono comunicare unicamente con i neuroni dello strato $i + 1$. Nei modelli più complessi è possibile

impostare la rete in modo tale che i neuroni possano avere collegamenti che bypassino certi strati o che facciano riferimento a strati precedenti o che comunichino con neuroni dello stesso strato in uno schema circolare.

Il pregio delle reti neuronali sta nella capacità di utilizzare informazioni non precedentemente trattate e di poter ricreare qualsiasi tipo di funzione (sia per problemi di regressione che di classificazione). Un problema particolarmente avvertito nelle applicazioni è però quello dell'overfitting ossia della tendenza del modello neuronale a specializzarsi troppo sui dati usati in fase di addestramento con una conseguente bassa performance nella successiva fase di generalizzazione.

Ciò viene confermato dal teorema di Kolmogorov il quale afferma che una rete neurale con sufficiente numero di neuroni nel livello nascosto e/o iterazione di addestramento può arrivare alla situazione in cui non commette alcun errore nel training set. Un'altra pecca del metodo è che i risultati ottenuti da una rete neurale non sono interpretabili perché di questo modello si può conoscere la struttura (determinata dal ricercatore) ma non la logica che avviene al suo interno una volta addestrata rendendo cosi lo strumento una black box.

Si consideri uno dei metodi più semplici di addestramento delle reti neuronali, la back propagation. Questo metodo di apprendimento supervisionato inizializza l'algoritmo assegnando un valore casuale a tutti i parametri del modello (ossia i pesi e le soglie descritte precedentemente) ed inserendo le varie osservazioni di addestramento al modello così definito. Essendo i parametri definiti casualmente le previsioni che si otterranno dal modello saranno quasi certamente sbagliate.

L'addestramento consiste nel "tornare indietro" modificando vari parametri e reinserendo l'informazione in input fino ad arrivare alla situazione in cui la rete restituisce un risultato corretto a fronte di un dato esempio.

Intuitivamente una volta addestrata la rete si hanno a disposizione una serie di valori che rappresentano i diversi parametri di peso e soglia, questi non danno alcuna informazione all'utilizzatore sulla funzione che la rete utilizza nei problemi di regressione o classificazione.

Un ulteriore problematica sta nella regola di apprendimento utilizzata, prima abbiamo detto che la rete viene addestrata modificando i parametri fino a che non classifica correttamente l'esempio che si sta trattando, la direzione verso cui vengono modificati i parametri viene eseguita in base all'opposto del gradiente di una determinata funzione di errore rispetto ai parametri.

Si dimostra che questa funzione potrebbe vantare diversi minimi relativi e che per come si struttura l'algoritmo di apprendimento la rete potrebbe convergere ad una soluzione non ottimale intrappolandosi in un minimo relativo.

1.2 Un esempio per capire

Al fine di capire il funzionamento di una rete neurale verrà qui proposto un semplice esempio in cui si mostrerà passo per passo il processo di apprendimento supervisionato di una rete a due strati in cui sono disponibili campioni che hanno già un valore "noto" di output.

In questo scenario dunque si suppone di avere delle informazioni (ad esempio: sinistri denunciati in una compagnia) per i quali è già noto quali campioni appartengono ad un tipo (ad esempio: sinistri fraudolenti) e quali ad un altro (ad esempio: sinistri effettivi).

Lo scopo è addestrare la rete su questi dati (detto appunto *training set*) al fine di poter determinare una qualche regola di calcolo che possa in futuro classificare *nuove* osservazioni per le quali non è noto il gruppo di appartenenza.

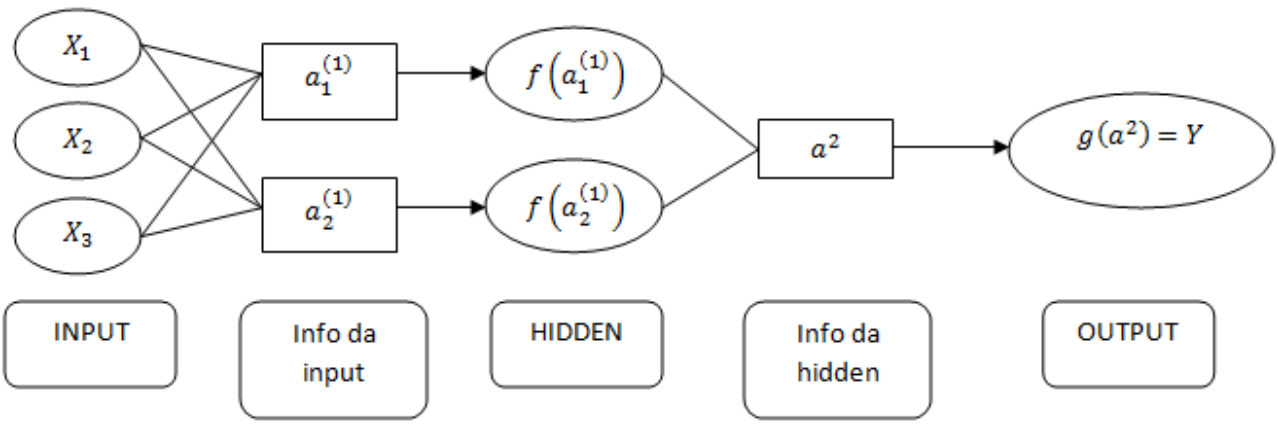
STEP 1: definire l'architettura della rete

Un primo passo da compiere è determinare la struttura della rete neuronale che si intende utilizzare. Il caso più semplice (che verrà qui trattato) prevede una struttura con almeno 2 strati (input, calcolo ed output) *unidirezionale* ove quindi il neurone di uno strato riceve informazione dai soli neuroni dello strato precedente e comunica l'informazione con i soli neuroni dello strato successivo.

Ai nostri fini utilizzeremo una struttura di questo tipo:

- livello di input: 3 neuroni indicati col pedice s
- livello di calcolo: 2 neuroni indicati col pedice j
- livello di output: 1 neurone

che possiamo rappresentare come



Notare che il numero di neuroni nel livello di input dipende dalla struttura del campione disponibile, se ogni osservazione si costituisce da un vettore di n valori lo strato di input avrà n neuroni ognuno dei quali riceverà una specifica informazione.

Nel nostro semplice caso stiamo sostanzialmente eseguendo un problema di classificazione quindi il risultato finale può essere schematicamente rappresentato da una variabile binaria unidimensionale che assuma valori 0 e 1.

STEP 2: addestramento della rete

Per capire come funziona il processo di addestramento è necessario fare un breve accenno sul funzionamento del singolo neurone. Una rete neurale non è nient'altro che una aggregazione di unità elementari che interagiscono tra di loro il cui funzionamento si può sintetizzare in un comportamento binario: acceso spento.

Ogni neurone (salvo quelli di input) riceve una informazione dallo strato precedente, se l'informazione ricevuta supera una determinata soglia il neurone si attiva e trasferisce a sua volta il segnale allo strato successivo.

Dato che non tutte le informazioni che il nodo riceve hanno lo stesso peso il segnale complessivo che viene ricevuto è definito come una media ponderata del tipo

$$a_j^{(1)} = \sum_{s=1}^{3} x_{is} w_{sj}^{(1)} + w_{0j}^{(1)} = \sum_{s=0}^{3} x_{is} w_{sj}^{(1)} \qquad con\ x_{i0} = 1$$

ove il $w_{0j}^{(1)}$ rappresenta il valore della soglia che può essere inserito nella sommatoria assumendo un neurone fittizio x_{i0} che assume sempre valore 1.

Notare l'apice di $a_j^{(1)}$ che indica che il j-esimo neurone dello strato di calcolo ricevere informazioni dal primo strato (ossia quello di input), logicamente indicheremo con un $a^{(2)}$ l'informazione che il nodo di output riceve dal quello intermedio.

Come si è prima accennato l'informazione che un neurone riceve deve superare un determinato valore soglia affinché questo si attivi, formalmente ciò viene rappresentato con una funzione detta "di trasferimento".

$$valore\ assunto\ dal\ neurone\ intermedio\ j-esimo = f\left(a_j^{(1)}\right)$$

La forma di tale funzione (insieme all'architettura della rete) è la seconda ipotesi che viene richiesta dal modello, di per se non esistono vincoli sul tipo di funzione da utilizzare ed il

caso più semplice potrebbe essere quello di una funzione a gradino che ben rappresenta la logica del "superare il valore soglia".

Ai nostri fini utilizzeremo una funzione di tipo sigmoide perché vanta una comoda proprietà che ci permetterà di semplificare i calcoli più avanti infatti

$$y = \frac{1}{1 + e^{-x}} \rightarrow \frac{\partial y}{\partial x} = y(1 - y)$$

STEP 2.1: inserimento dei dati

Definita la struttura e le funzioni di trasferimento andremo a definire in maniera casuale i diversi pesi e valori soglia della rete. Visto che il risultato del modello è definito da i valori che assumono tali parametri il problema di addestramento consiste nel "ben definire" i valori che questi devono avere.

Per farlo si segue una logica di minimizzazione dell'errore commesso. Infatti avendo definito i diversi parametri in via casuale si passerà all'inserimento nella rete dei diversi campioni x_i.

Ragionevolmente i risultati y_i che la rete cosi calibrata restituirà saranno diversi da quelli reali ma proprio perché sono noti i valori che la rete *dovrebbe* restituire (apprendimento supervisionato) possiamo calcolare l'errore commesso dalla rete come

$$E = \sum_{i=1}^{n} E_i \quad con\ E_i = (y_i - t_i)^2$$

ove la variabile t_i rappresenta il valore target ossia quello che dovrebbe restituire la rete.

Il nostro obiettivo è modificare i pesi w e le soglie in modo che sia minimizzato E, per capire verso che direzione muoversi nel modificare questi pesi si può osservare la nostra E come funzione dei pesi e quindi ci dovremo spostare verso la direzione che minimizza il valore di E.

STEP 2.2: calcolo delle derivate parziali

Avendo definito con

$$a_j^{(1)} = \sum_{s=0}^{3} x_{is} w_{sj}^{(1)}$$

$$a^{(2)} = \sum_{j=1}^{2} f\left(a_j^{(1)}\right) w_j^{(2)} + w_0^{(2)} = \sum_{j=0}^{2} f\left(a_j^{(1)}\right) w_j^{(2)} = \sum_{j=0}^{2} f\left(\sum_{s=0}^{3} x_{is} w_{sj}^{(1)}\right) w_j^{(2)}$$

rispettivamente l'informazione che il j-esimo neurone intermedio riceve dallo strato di input e l'informazione che il neurone di output riceve dallo strato nascosto ed avendo assunto che le funzioni di trasferimento siano definite come

$$g\left(a^{(2)}\right) = \frac{1}{1 + e^{-a^{(2)}}} = y_i$$

$$f\left(a_j^{(1)}\right) = \frac{1}{1 + e^{-a_j^{(1)}}}$$

possiamo modificare i nostri pesi w al fine di minimizzare l'errore. Nel metodo qui proposto (detto forward/backward) la modalità con cui ciò viene fatto parte aggiustando i pesi tra lo strato intermedio e l'output (ossia i $w_j^{(2)}$) per poi passare a quelli tra l'input e l'intermedio (i $w_{sj}^{(1)}$).

Avremo quindi

$$\frac{\partial E_i}{\partial w_j^{(2)}} = \frac{\partial E_i}{\partial y_i} \frac{\partial y_i}{\partial a^{(2)}} \frac{\partial a^{(2)}}{\partial w_j^{(2)}} = (y_i - t_i) y_i (1 - y_i) f\left(a_j^{(1)}\right)$$

per i pesi tra l'hidden e l'output e

$$\frac{\partial E_i}{\partial w_{sj}^{(1)}} = \frac{\partial E_i}{\partial y_i} \frac{\partial y_i}{\partial a^{(2)}} \frac{\partial a^{(2)}}{\partial f\left(a_j^{(1)}\right)} \frac{\partial f\left(a_j^{(1)}\right)}{\partial a_j^{(1)}} \frac{\partial a_j^{(1)}}{\partial w_{sj}^{(1)}}$$

$$= (y_i - t_i) y_i (1 - y_i) w_j^{(2)} f\left(a_j^{(1)}\right) \left(1 - f\left(a_j^{(1)}\right)\right) x_{is}$$

STEP 2.3: minimizzazione dell'errore

Il calcolo delle derivate parziali permette di capire se effettivamente esiste un punto di minimo assoluto nella funzione dell'errore e come si modifica la grandezza E al variare dei pesi e soglie di ogni neurone.

L'obiettivo è minimizzare E, quindi è necessario definire una regola di apprendimento ben precisa che ci dica come modificare pesi. Nella versione classica del back propagation la regola di apprendimento più semplice è quella dello *steepest descent* che determina la variazione dei pesi di ogni neurone in base alla direzione in cui l'errore si riduce più rapidamente (ossia il gradiente delle derivate parziali di E con ogni peso).

STEP 3: test out of sample

Uno dei problemi delle reti neurali è la loro capacità di iper-adattarsi ai dati del training set, questo comporta una problematica nel modello in quanto l'obiettivo è poter classificare correttamente i nuovi campioni che verranno dati alla rete per i quali non è noto il valore di output.

Per poter valutare la capacità di generalizzazione del modello una soluzione consiste nel non utilizzare tutti i dati disponibili del training set ma di usarne solo una parte. I dati non utilizzati nell'addestramento serviranno a valutare la capacità di classificazione della rete perché permetteranno di vedere come il modello si comporta con delle nuove osservazioni (non utilizzate nella fase di addestramento) per le quali è noto l'output.

1.3 Reti Neurali ed SVM

Presentato il funzionamento di una rete neurale elementare possiamo fare alcuni confronti con quanto abbiamo visto nelle SVM.

Innanzi tutto richiamiamo la nostra diseguaglianza fondamentale

$$R(\alpha_l) \leq R_{emp}(\alpha_l) + \Omega(h, l, \eta)$$

ove sappiamo che il secondo membro della parte a destra della diseguaglianza è legato al grado di complessità del set di funzioni utilizzato per la classificazione.

Il rischio empirico non ci da problemi perché è semplicemente l'errore commesso dalla learning machine una volta addestrata sui dati utilizzati in fase di training ma come si definisce la $\Omega(h, l, \eta)$ nelle reti neurali.

Negli esempi fin'ora proposti si è sempre considerata una rete elementare costituita da un solo strato nascosto con andamento strettamente feed forward. Nella pratica però la struttura di una rete può essere estremamente complessa prevedendo una pluralità di strati di calcolo e flussi informativi che possono anche "tornare indietro" agli strati precedenti.

La misura di complessità del set di funzioni scelto $\Omega(h, l, \eta)$ è quindi definito dal grado di complessità della rete neurale. Infatti all'aumentare del grado di complessità aumenta anche la capacità della rete di definire classificatori fortemente non lineari e quindi di azzerare il rischio empirico (cosi come accade nel considerare set di ipotesi Λ più complessi di quello lineare).

La struttura della rete neurale è essenzialmente definita dallo sperimentatore in una fase antecedente a quella di addestramento. Quindi la logica di fondo di una RN consiste nel

1. Definire la struttura di rete ossia $\Omega(h, l, \eta)$
2. Addestrare la rete affinché ottenga il più basso $R_{emp}(\alpha_l)$

se nella fase di costruzione della struttura si è definito un "intreccio" troppo complesso anche azzerando il rischio empirico non si otterrebbe una buona performance out of sample incorrendo nella situazione di overfitting.

Nelle SVM la metodologia di addestramento è diametralmente opposta

1. Si definisce un livello di errore in sample definito accettabile C
2. Si addestra la SVM affinché ottenga il maggior margine possibile minimizzando $\Omega(h, l, \eta)$

Il problema non si riduce ad un mero approccio invertito perché se nelle SVM sia il rischio empirico che il grado di complessità sono facilmente controllabili dallo sperimentatore sulla base del principio di SRM nelle RN la definizione del grado di complessità della rete non è così immediata e la cosa si risolve con qualche euristica.

Un'altra considerazione da fare sulla differenza tra SVM e RN riguarda l'addestramento della learning machine.

Nell'approccio back propagation adottato dalle RN la superficie di perdita su cui dobbiamo cercare il minimo è tipicamente molto irregolare ossia con diversi minimi relativi. Non è' quindi certo che l'algoritmo di apprendimento della rete converga al punto di minimo globale (sempre che esista).

Nelle SVM la definizione dell'iperpiano ottimale è ottenuta risolvendo un problema di ottimizzazione quadratico convessa per cui è certa l'esistenza minimo globale che può essere trovato senza l'utilizzo di metodologie ricorsive. Il rischio di intrappolamento in minimi relativi viene quindi scongiurato.

Come vedremo la necessità di metodologie iterative entra nelle SVM solo in fase di gestione delle feature quando si vuole determinare quella combinazione di variabili che permetta di ottenere la miglior classificazione

2.Alberi Decisionali

Visti le SVM e le reti neuronali in questa sezione verrà presentato sinteticamente un terzo (ed ultimo) metodo di classificazione appartenente alla famiglia dei metodi di apprendimento automatico.

Come nel capitolo delle reti neurali la presentazione dell'argomento sarà articolata su tre parti

1. Presentazione discorsiva del metodo
2. Presentazione di un esempio pratico
3. Confronto con le SVM in base alla teoria dello statistical learning

In particolare verrà inizialmente presentata la logica di funzionamento di un singolo albero decisionale per poi estendere il ragionamento alle tecniche di bagging ed infine alle c.d. random forest.

2.1 Introduzione

Gli alberi decisionali (sinteticamente AD) sono una tecnica di machine learning utilizzabile in entrambi gli ambiti della classificazione e regressione che ha il pregio di vantare una struttura logica molto semplice e facilmente interpretabile al punto tale da essere definita come una white box a differenza di altri metodi come le reti neuronali e SVM.

Anche se la logica di fondo non varia sensibilmente tra la regressione e classificazione in questa breve introduzione ci si focalizzerà su quest'ultimo ambito essendo probabilmente il più intuitivo e quindi utile nell'affacciarsi per la prima volta agli AD.

Il metodo prevede di partizionare il predictor space (da ora in poi chiamato input space per coerenza con la nomenclatura utilizzata nelle SVM) in una serie di regioni che definiscono le varie classi esistenti nel problema. Una volta trovata la miglior partizione dell'input space una nuova osservazione viene assegnata ad una classe in base a dove "cade" rispetto alle partizioni dello spazio sovra definite.

Dato che la logica dell'insieme di regole utilizzate nel partizionare l'input space è di tipo gerarchico (o "top/down") possiamo rappresentare il processo di classificazione di una osservazione con uno schema ad albero (tipicamente rappresentato al rovescio).

Tra gli altri pregi che vanta il modello si ha la possibilità di lavorare naturalmente su problemi in cui esistono più di due classi e la capacità di operare con osservazioni ove le coordinate sono definite in termini di variabili qualitative e quantitative. Infatti non è richiesta alcuna transcodifica di informazioni puramente qualitative come avviene in altre metodologie di apprendimento automatico (quali ad esempio gli algoritmi genetici).

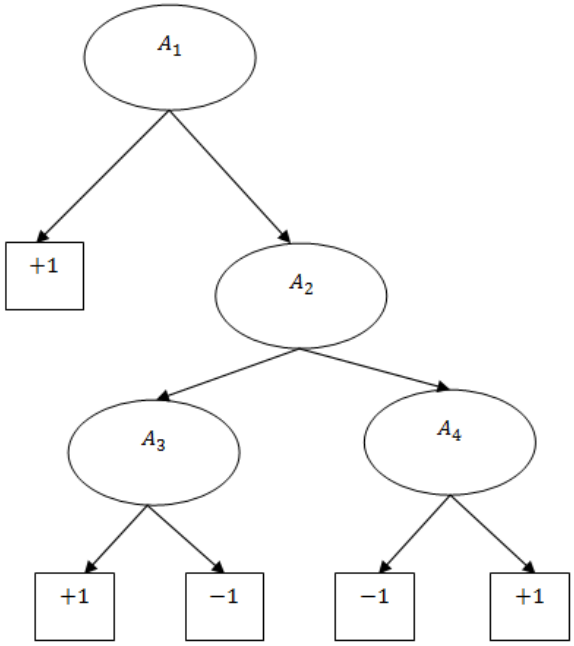
Nonostante tutti questi pregi gli AD tipicamente non vantano delle performance predittive comparabili con quelle di metodi più sofisticati quali le reti neurali o le SVM. Come vedremo è però possibile aumentare considerevolmente la performance di queste tecniche tramite una loro generalizzazione (chiamata random forest) che però sacrifica parte dell'interpretabilità dei risultati.

2.2 Un esempio per capire

Si assuma di operare in ambito supervisionato e di avere a disposizione m osservazioni nel training set, ogni osservazione è definita come una coppia $(x_i, y_i)_{i=1,\dots,m}$ ove x rappresenta un vettore di dimensione h che contiene le diverse "caratteristiche" che vengono rilevate per ogni osservazione mentre y è una etichetta che indica la classe di appartenenza della rispettiva osservazione.

In questa sede chiameremo "attributi" le diverse caratteristiche (o meglio variabili) che definiscono i vettori x. Nel nostro caso quindi ogni punto è costituito da h diversi attributi i quali possono essere contemporaneamente valori numerici continui o discreti, valori binari, categorie ecc.

La stessa y non ha alcun vincolo e quindi non necessariamente deve rappresentare un set di classi dicotomiche ma nel nostro caso assumeremo per semplicità che sia comunque definito come $y = \pm 1$.



Come costruire l'albero decisionale? Nella sostanza la struttura di un albero decisionale è definita da una serie di "nodi" in ognuno dei quali viene eseguito un certo test volto a ripartire le osservazioni giunte in quel nodo in diversi sottonodi. Ogni test riguarda un particolare attributo che definisce le osservazioni x e si esegue confrontando il valore assunto dall'osservazione i-esima per quella variabile ed un certo valore soglia.

Da ogni nodo partono quindi diversi sottorami e per ognuno di questi si dovrà definire un test analogo a quello del livello precedente.

Il processo si ripete quindi in maniera ricorsiva fino ad arrivare ad un punto in cui la osservazione viene assegnata ad una certa classe di y. In coerenza con la metafora dell'albero questo livello dell'AD è definito foglia.

Come il lettore avrà già intuito questo meccanismo pone un problema di ordinamento infatti se ogni nodo rappresenta un test che riguarda un particolare attributo di x e se la struttura decisionale è di tipo gerarchico allora qual è il primo attributo da utilizzare per "iniziare l'albero"? Esistono di fatto differenze nella capacità predittiva del modello se si inizializza l'AD con un diverso attributo o è indifferente da dove si comincia?

Il metodo che è stato proposto per la costruzione degli AD si basa sul concetto di entropia applicato con logica ricorsiva *greedy* al training set. In seguito vedremo nel dettaglio cosa significhi quanto appena scritto ma per meglio comprendere la logica di fondo del metodo si consideri il seguente ragionamento: se dovessi costruire un classificatore che si basa su un unico attributo del vettore x quale variabile permetterebbe la miglior classificazione possibile del training set? Intuitivamente andremo a scegliere quella che ci permette di guadagnare la maggior informazione possibile sulla relazione che intercorre tra le classi e le osservazioni di addestramento.

Quindi l'ideale sarebbe una variabile in grado di ripartire tutti i record nelle rispettive classi perché ciò vorrebbe dire che quel particolare attributo permette assegnare in maniera deterministica le varie osservazioni. Nella sostanza vogliamo che delle nostre n osservazioni tutte quelle classificate come $+1$ siano mandate in una foglia e tutte quelle classificate come -1 in un'altra.

Il caso peggiore invece è dato da un attributo che ripartisce i records su due foglie mantenendo le proporzioni delle osservazioni del training set. Se ad esempio il 60% dei records appartiene alla classe $+1$ ed il restante 40% alla classe -1 nelle due foglie avremo per ognuna un 60% di osservazioni $+1$ ed un 40% di osservazioni -1; il nostro livello di informazione sul fenomeno è rimasto esattamente lo stesso!

Una prima soluzione per individuare la "capacità informativa" di un attributo che fu proposta si basa sulla misurazione di un tasso di errore definito dalla frazione delle osservazioni di addestramento che finiscono in una foglia senza appartenere alla classe predominante.

Il problema di questo metodo è che non permette di avere una misura sufficientemente rappresentativa del livello di "purezza" del nodo e quindi si sono proposte le seguenti alternative:

Gini Index

$$G = \sum_{i=1}^{K} p_{mi}(1 - p_{mi})$$

questo indice misura la varianza totale sulle K classi ed è pari a zero (varianza nulla) per p_{mk} = 0 o 1.Quindi se $G \to 0$ nella regione i-esima dell'input space sono presenti solo punti appartenenti ad una particolare classe.

Un altro indice utilizzato per misurare il guadagno informativo dato da un certo attributo nel ripartire il training set è l'entropia attesa.

Il concetto di entropia qui utilizzato richiama l'interpretazione che viene adottata nell'ambito dalla meccanica statistica e fa riferimento ad una misura del livello di ordine nello spazio dei records considerati nell'addestramento dell'albero. Maggiore è il livello di entropia e maggiore è la difficoltà che si ha nell'assegnare le diverse osservazioni alle classi di appartenenza

Considerando per semplicità una classificazione binaria (come nel nostro esempio) il concetto di entropia viene definito dalla seguente espressione

$$H\left(\frac{p}{m},\frac{n}{m}\right) = -\frac{p}{m}log\left(\frac{p}{m}\right) - \frac{n}{m}log\left(\frac{n}{m}\right)$$

ove p rappresenta il numero di vettori classificati $+1$ ed n il numero di punti classificati come -1.

Si osservi come l'entropia sia un valore che oscilla tra 0 quando $\frac{p}{m}$ = 0 o 1 ed 1 quando $\frac{p}{m} = \frac{n}{m} = \frac{1}{2}$ che nel caso di una classificazione binaria risulta essere lo scenario più difficile da trattare.

Se A è un attributo che può assumere K valori (quindi separerebbe il training set E in K sottoinsiemi $E_1, \dots, E_K$) l'entropia attesa (EH) rimanente dopo aver utilizzato l'attributo A sarebbe pari a

$$EH(A) = -\sum_{i=1}^{K}\frac{p_i + n_i}{m}H\left(\frac{p_i}{p_i + n_i},\frac{n_i}{p_i + n_i}\right)$$

ove $\frac{p_i+n_i}{m}$ è la probabilità di rientrare nella classe i-esima.

Viene quindi definito information gain $I(A)$ la differenza tra il livello di entropia di partenza e quello ottenuto utilizzando l'attributo A

$$I(A) = H\left(\frac{p}{m},\frac{n}{m}\right) - EH(A)$$

Visto che all'aumentare dell'entropia aumenta la nostra ignoranza sul fenomeno oggetto di studio si sceglie come attributo per il primo nodo quello che comporta la più alta diminuzione di entropia (ossia il massimo information gain).

Definito il primo nodo si riesegue il processo in via ricorsiva sui K nodi precedentemente determinati fino alla definizione dell'intero albero.

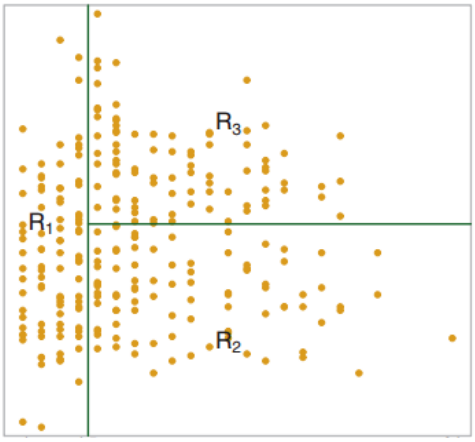
La definizione dell'addestramento "ricorsivo greedy" nasce dal fatto che la costruzione dei vari nodi si basa sulla selezione dell'attributo che permette in quel momento il più alto

information gain ma nulla ci assicura che utilizzando un diverso ordine gerarchico lo stesso attributo possa dare una maggior diminuzione di entropia ad uno stesso livello dell'albero (un po' come avviene nella feature selection quando si utilizzano metodi di ranking sui singoli attributi).

E' inoltre evidente che questo metodo dipende molto da come è strutturato il training set e che se si andasse a togliere o aggiungere una osservazione di una determinata classe cambierebbe l'information gain di un certo attributo e da li anche la struttura dell'AD. Come vedremo questo sarà uno dei motivi che sta alla base di una generalizzazione degli AD chiamata random forest e bagging.

Prima di concludere questo breve esempio sull'addestramento degli AD intendo presentare al lettore una breve nota su come avviene la classificazione di una nuova osservazione una volta addestrato l'albero.

Si consideri un data set in cui ogni osservazione è definita da un vettore x_i di due sole coordinate numeriche e da una etichetta $y_i = \pm 1$.



Dato che ogni punto si definisce da due sole coordinate entrambe numeriche possiamo rappresentare il training set su un piano cartesiano.

Il meccanismo gerarchico di classificazione che abbiamo rappresentato con uno schema ad albero può quindi essere anche osservato direttamente nello spazio delle variabili di input nel seguente modo

In questo caso l'input space viene partizionato in tre regioni che sono definite dai test eseguiti sui due attributi. Definendo con x_1 l'asse delle ascisse e con x_2 quello delle ordinate dal grafico a sinistra comprendiamo che il primo test sarà eseguito su x_1 che quindi è la variabile che permette il massimo information gain.

Tutti i punti che si trovano a sinistra del valore soglia di x_1 andranno in una certa classe, gli altri passano attraverso un secondo test definito per l'attributo x_2.

E' bene evidenziare che non vi è alcun vincolo su come debbano essere geometricamente definite le linee di confine tra le diverse regioni e di fatto potremmo avere regioni definite da curve e/o rette non necessariamente parallele agli assi. Ovviamente in questa sede non è nostro obiettivo andare a trattare casi così complessi.

Eseguita la partizione dell'input space la classificazione avviene semplicemente assegnando alla regione la classe predominante. Infatti non tutte le osservazioni di

addestramento che rientrano in una stessa regione sono necessariamente appartenenti ad una sola classe quindi si va a calcolare al tipologia di classe più frequente in quella partizione. Ritroviamo quindi la logica vista precedentemente quando ci si chiedeva quale criterio dovesse governare la scelta di un attributo nella costruzione dell'albero.

Evidentemente la situazione ideale è quella in cui per ogni regione si hanno solo punti appartenenti ad una stessa classe (che avrà frequenza pari a 1 quindi) anche se ciò farebbe sorgere il sospetto di un sovra adattamento del modello ai dati.

Dato una osservazione di test la classificazione avviene semplicemente osservando in quale regione il punto cade e assegnandoli quindi la classe che fa riferimento alla partizione.

Nel caso in cui si utilizzasse un AD per eseguire una regressione la logica di fondo è quasi immutata infatti il metodo prevede di calcolare per ogni regione il valor medio (o mediano) dei punti del training set che vi rientrano. La previsione per una nuova osservazione che ricade nella regione è semplicemente pari alla media della regione. La maggior differenza sta quindi nella modalità con cui si ripartisce l'input space nelle diverse regioni (che qui non tratteremo).

2.3 Oltre il singolo albero: introduzione della casualità

2.3.1 Bagging

Come accennato alla fine del precedente paragrafo il punto critico che si ha nella fase di addestramento di un AD è costituito dalla forte dipendenza tra la struttura dell'albero addestrato e la distribuzione tra le diverse classi dei dati utilizzati per l'addestramento.

Per poter meglio capire questo punto è però necessario prima introdurre il concetto di varianza e distorsione. Uno dei classici problemi nell'ambito del machine e statistical learning è quello del trade off tra la capacità di generalizzazione e di adattamento ai dati del modello.

Queste sono due fonti di errore che molto raramente possono essere contemporaneamente minimizzate e che formalmente definiamo come:

$$Bias\left[\hat{f}(x)\right] = E\left[\hat{f}(x)\right] - f(x)$$

$$Var\left[\hat{f}(x)\right] = E\left[\left(\hat{f}(x) - E\left[\hat{f}(x)\right]\right)^2\right]$$

Il Bias (o distorsione) quantifica la capacità di generalizzazione del metodo ossia la performance nel classificare punti non utilizzati nella fase di addestramento. Ciò viene misurato verificando se in media le previsioni ottenute dal modello per quell'osservazione convergono alla sua effettiva classe.

La varianza riguarda invece la capacità di adattamento del modello ai dati usati nell'addestramento e si misura con il livello delle previsioni fatte dai vari classificatori nell'introno della previsione media.

Gli AD sono tecniche di machine learning caratterizzate da una bassa distorsione ed una elevata varianza.

La bassa distorsione dell'AD fa si che in media le previsioni (ottenute con AD addestrati su diversi data set) che si ottengono per uno valore test x tendano a convergere alla classe reale dell'osservazione. D'altro canto l'elevata varianza del metodo fa si che AD addestrati su differenti data set diano classificazioni anche molto diverse per una stessa osservazione test x (che però convergono al valore reale data la bassa distorsione).

Il primo metodo che fu proposto per far fronte al problema dell'elevata varianza negli alberi decisionali è il così detto bagging. Esso consiste nell'utilizzare non uno ma B diversi data set e su ognuno di questi addestrare un albero T_b.

Così facendo si ottengono B modelli $\hat{f}_b(x)$ ad alta varianza i quali vengono aggregati in modo tale da ottenere un singolo modello di classificazione a bassa varianza

$$\hat{f}_{avg}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}_b(x)$$

Nel caso di alberi decisionali per problemi di classificazione questa media viene calcolata con il principio del "voto di maggioranza" ossia data una particolare foglia dell'albero si osserva a quale classe i diversi metodi $\hat{f}_b(x)$ assegnano l'osservazione e si determina la classe per la corrispondente foglia di $\hat{f}_{avg}(x)$ prendendo quella osservata più frequentemente sui B alberi decisionali.

Ovviamente nella pratica non si ha accesso a data set multipli quindi per poter eseguire il bagging si ricorre a metodologie quali ad esempio il bootstrapping che permette di generare B pseudo data set tramite una estrazione casuale con remissione degli elementi del training set di partenza.

2.3.2 Random Forest

Il bagging introduce un elemento di casualità nel metodo non presente nella versione plain vanilla degli alberi decisionali. Il fatto di definire un AD come media di una serie di AD addestrati su diversi data set estratti casualmente dai dati originali comporta una diminuzione del problema della varianza sovra indicato con il costo di una corrispondente diminuzione dell'interpretabilità dei risultati del metodo.

Il fatto di addestrare un insieme di AD su data set estratti casualmente spesso però non risulta sufficiente ad ottenere un classificatore che vanti una sensibile diminuzione di variabilità.

Il motivo di questo problema può essere intuito con il seguente ragionamento: si supponga di avere delle osservazioni x definite da h attributi A_i. Assumiamo che di questi h attributi ve ne sia uno che sia in grado di diminuire il livello di entropia in maniera sostanziale e che questo $I(A)$ sia di molto superiore ad ogni altro information gain ottenibile con gli altri attributi.

Cosa è ragionevole aspettarsi nella fase di addestramento dei diversi alberi? Ricordando il funzionamento dell'approccio ricorsivo greedy top/down quello che verosimilmente accadrà sarà che tutti gli alberi T_b che addestreremo sui diversi subset di dati avranno come nodo primario proprio questo particolare attributo.

Ciò comporta il fatto che il semplice metodo di bagging restituisca degli alberi T_b la cui struttura decisionale è sempre molto simile quindi quando eseguo la media dei diversi risultati difficilmente riuscirò ad ottenere un AD che vanta un sensibile calo della varianza.

Quello che è necessario fare è cercare di de-correlare i diversi T_b in modo tale da ottenere alberi sensibilmente diversi tra loro nella struttura decisionale e poter ricavare un classificatore medio che presenti una sensibile diminuzione della varianza.

Per de-correlare i diversi T_b si è proposto di addestrare B alberi decisionali su dei subset del data base estratti casualmente e che considerano anche un subset estratto casualmente degli attributi presenti nei dati originali.

Se quindi abbiamo un data set di m osservazioni $\{x_i, y_i\}$ ove ogni x_i è definito da un vettore di n elementi questo metodo (chiamato random forest) prevede di estrarre casualmente B subset di dati ove ogni osservazione x_i è a sua volta definito da un sottoinsieme di coordinate estratte casualmente.

Ciò permette di evitare il problema sovra evidenziato visto che per certi alberi decisionali l'attributo A che vantava il più alto $I(A)$ non sarebbe considerato.

Una volta addestrati i B alberi T_b si calcola il classificatore medio con la tecnica del voto di maggioranza sopra illustrata e si dimostra che la media dei T_b tra loro incorrelati converga ad un classificatore di ottimo.

2.3.4 Confronto con le SVM

A differenza di quanto fatto nel capitolo sulle reti neuronali il confronto tra le SVM e gli alberi decisionali è molto più complesso per il semplice fatto che non risulta così immediato quantificare la VC dimension di una random forest.

Da un lato la VC dimension di un singolo albero è abbastanza semplice da quantificare perché sappiamo che ogni nodo definisce un livello soglia che partiziona lo spazio campionario.

Un singolo nodo divide lo spazio in due partizioni quindi il massimo numero di punti frantumabili è due. Passando ad un albero con due nodi questi generano due valori soglia che possono ripartire lo spazio in tre o al più quattro aree (a seconda di come si posiziona rispetto al precedente valore soglia)quindi la sua VC dimension sarà tre.

In generale quando trattiamo un singolo albero la VC dimension è pari a n + 1 ove n è il numero di nodi che costituisce l'albero.

Il problema sorge quando si vuole misurare il grado di complessità di una random forest per la quale non il calcolo della VC dimension non è così agevolo. Per questo motivo il confronto con le SVM risulta essere problematico dato dal fatto che solo l'errore empirico è facilmente misurabile su entrambi i modelli ma spesso le SVM vantano performance più elevate.

Non deve però essere sottovalutata la semplicità della logica dietro gli alberi decisionali nonché la loro naturale capacità di operare con variabili sia quantitative che qualitative senza dover ricorrere all'utilizzo di variabili dummy.

Inoltre gli alberi decisionali sono direttamente applicabili anche in problemi di classificazione non binaria senza dover ricorrere a tecniche come la "uno a molti" che devono essere utilizzate nelle SVM data la loro originaria incapacità di lavorare oltre il semplice caso di classificazione a due soli stati.

Appendice: Teoria dell'Informazione ed Entropia

Secondo la Teoria dell'Informazione (sinteticamente TdL) l'informazione che una variabile aleatoria $\tilde{X}$ fornisce per ogni sua manifestazione x è inversamente proporzionale alla probabilità di assumere quel valore.

$$info(\tilde{X} = x) = \frac{1}{P(\tilde{X} = x)} = \frac{1}{P_x}$$

Tipicamente questo valore viene riscritto su scala logaritmica in base 2 ottenendo quindi che l'informazione fornita da $\tilde{X}$ quando assume valore x è

$$info(\tilde{X} = x) = \log_2 \frac{1}{P_x}$$

Indichiamo con $H(\tilde{X})$ l'entropia di $\tilde{X}$ ossia l'informazione fornita in media dalla variabile aleatoria oggetto di studio

$$H(\tilde{X}) = \sum_X P(x) \log_2 \frac{1}{P_x}$$

Maggiore è l'entropia di una variabile aleatoria e minore è la conoscenza che abbiamo su di essa. Infatti l'entropia può essere vista come una *"misura del numero di stati che hanno una probabilità di accadimento significativa"* (Leonard Susskind).

Si consideri ad esempio una variabile $\tilde{X} = \{x_1, x_2\}$ e si assuma che le probabilità di accadimento dei due stati siano

x_1	x_2	$H(\tilde{X})$
0,99	0,01	0,024321
0,75	0,25	0,244219
0,5	0,5	0,30103
0,01	0,99	0,024321

Si osservi come nel caso in cui $x_1 = 0{,}99$ la variabile aleatoria oggetto di studio è "quasi degenere". In tal caso il livello di entropia risulta essere molto basso perché evidentemente la conoscenza di x_1 da un alto grado di informazione sull'intera variabile aleatoria. Il caso in cui le probabilità di accadimento dei due stadi si equivalgono il livello di entropia raggiunte il massimo essendo il caso in cui si ha la minor conoscenza di $\tilde{X}$.

Possiamo estendere questo concetto anche a due variabili aleatorie nelle nozioni di entropia complessiva e condizionata.

Chiamiamo entropia complessiva di due variabili aleatorie $\tilde{X}$ ed $\tilde{Y}$ la media delle informazioni fornite dalla conoscenza congiunta dei valori che possono assumere

$$H(\tilde{X}, \tilde{Y}) = \sum_{X,Y} P(x, y) \log_2 \frac{1}{P(x, y)}$$

Chiamiamo invece entropia condizionata di $\tilde{X}$ dato $\tilde{Y}$ la seguente quantità

$$H(\tilde{X}|\tilde{Y}) = \sum_{X,Y} P(x|y) \log_2 \frac{1}{P(x|y)}$$

che indica l'incertezza su $\tilde{X}$ noto $\tilde{Y}$. Può essere anche interpretato come l'ammontare di informazioni necessarie per conoscere completamente $\tilde{X}$ posto di conoscere interamente di $\tilde{Y}$.

Si dimostra che

$$H(\tilde{X}|\tilde{Y}) \leq H(\tilde{X})$$

quindi se è noto $\tilde{Y}$ possiamo sfruttare questa informazione nella fase di costruzione dell'albero per ridurre l'entropia sfruttando la seguente relazione

$$I(X, Y) = H(X) - H(X, Y) = I(Y, X)$$

ove $I(X, Y)$ è l'informazione fornita da una variabile sull'altra e si dimostra che questa quantità è sempre la stessa (ossia $I(X, Y) = I(Y, X)$).

Definiamo infine con gain ratio il rapporto $I(X, Y)/H(Y)$ che è la percentuale di informazione fornita da Y utile nel caratterizzare la X.

3 Confronto tra classificatori: un esempio con dati simulati

3.1 Introduzione

Dopo aver presentato la teoria che sta alla base del funzionamento di una pluralità di classificatori quali SVM, reti neuronali, alberi decisionali e random forest potrebbe essere interessante andare a confrontare i diversi modelli in un caso applicato.

Va precisato che un confronto più approfondito nelle performance di classificazione tra i quattro modelli sopra elencati sarà presentato nella quinta parte di questo elaborato in cui si utilizzerà un data base reale riguardante il fenomeno della frode assicurativa.

In questo capitolo ci limitiamo ad offrire un'esercitazione con dati simulati con l'obiettivo di esplorare due situazioni che possono presentarsi nell'ambito dei problemi di classificazione:

1. confusione tra i due gruppi
2. asimmetria nella frequenza di osservazione dei due gruppi

Riprendendo quanto descritto nel capitolo delle SVM è noto che un classificatore non è altro che una funzione che va a partizionare lo spazio di input in due zone, una nuova osservazione è quindi classificata in base a dove questa si posiziona all'interno di questo spazio. Nell'esempio con dati simulati presentato nel capitolo delle SVM i due gruppi di punti erano perfettamente separabili ma all'aumentare del grado di confusione dei due insiemi l'individuazione di una funzione in grado di separare correttamente tutte le osservazioni risulta sempre più problematica. Nell'esercitazione che andremo a proporre in questo capitolo si confronterà la performance out of sample dei quattro classificatori all'aumentare del grado di confusione tra i gruppi che sarà misurato da un apposito parametro α definito tra 0 ed 1.

Un altro problema che può sorgere nell'ambito della classificazione (e come vedremo interesserà il caso della frode assicurativa nella quinta parte) si ha quando le osservazioni delle due classi non sono equamente distribuite nel training set. Intuitivamente: nel caso in cui vi sia una percentuale $\delta \to 0$ di osservazioni relative ad un certo gruppo (di minoranza) un classificatore tenderà a riconoscere ogni nuova osservazione come appartenente al gruppo di maggioranza. Questo perché l'errore che viene commesso dal modello in questo modo è pari ad una percentuale δ tendente a zero. Quindi il risultato è una macchina addestrata che di fatto non esegue alcuna classificazione (in quanto incapace di distinguere tra i due gruppi) e che vanta una "performance" (evidentemente distorta) tendente ad 1.

3.2 Presentazione della simulazione

In questa sessione andiamo a confrontare la performance di classificazione dei seguenti modelli:

1. SVM lineare
2. SVM Sigmoide
3. SVM Polinomiale
4. SVM Radiale
5. Albero Decisionale
6. Random Forest
7. Rete Neurale con un hidden layer

focalizzando in particolare sulla problematica del grado di confusione tra i gruppi.

Per poter misurare questo livello di confusione si è fatto il seguente ragionamento: si generano da una normale standard m di punti le cui coordinate (x_1, x_2) possono variare unicamente tra 0 e 1. Abbiamo quindi un insieme di punti racchiuso in un quadrato di lato unitario che va a rappresentare il gruppo di riferimento.

Andremo poi a generare un numero $m * n$ di punti da una normale standard le cui coordinate (x_1, x_2) sono tali per cui

$$x_1 \in [1 - a, 2 - a]$$

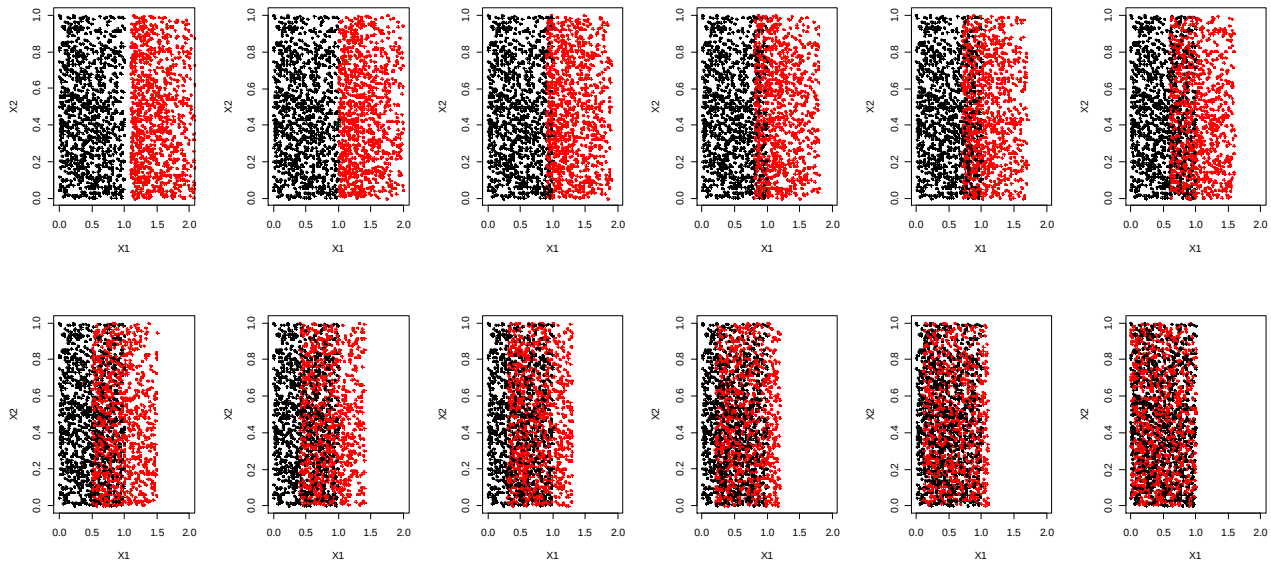
$$a, x_2 \; ed \; n \in [0,1]$$

otteniamo quindi un insieme di punti racchiusi anch'essi in un quadrato di lato unitario che al variare di a si sovrappone sempre di più al quadrato di partenza.

In particolare se a = 0 i due insiemi di punti sono perfettamente separati e quindi ci aspettiamo che tutti i classificatori siano in grado di separare correttamente le due classi. Quando a = 1 i due insiemi di punti sono perfettamente sovrapposti andando quindi a rappresentare il grado di confusione massima. Il parametro a è quindi pari all'area di sovrapposizione dei due quadrati e ci permette di misurare tale grado di confusione tra i gruppi.

In questo primo esempio poniamo n = 1 quindi la frequenza di osservazione tra le due classi è perfettamente bilanciata. Nel successivo andremo a porre n = 0,06 per vedere come i classificatori si comportano nel riconoscere la classe di minoranza.

Quanto descritto sopra è facilmente comprensibile dal seguente grafico:



L'insieme di punti in nero rappresenta il gruppo di riferimento mentre quello in rosso si sovrappone sempre di più a quello di partenza man mano che il parametro a tende ad 1. Per eseguire il confronto tra i modelli questi vengono addestrati e testati sugli stessi data base di training e test, tali data base si differenziano tra loro in base al grado di confusione tra gruppi.

Il confronto avviene quindi in base al complessivo numero di punti classificati erroneamente nella fase di test. Questa simulazione è quindi rieseguita 100 volte e per ogni macchina addestrata e grado di confusione si va a registrare il minimo, massimo e valor medio dell'errore di classificazione complessivamente osservato.

I risultati ottenuti sono riassunti nella seguente tabella:

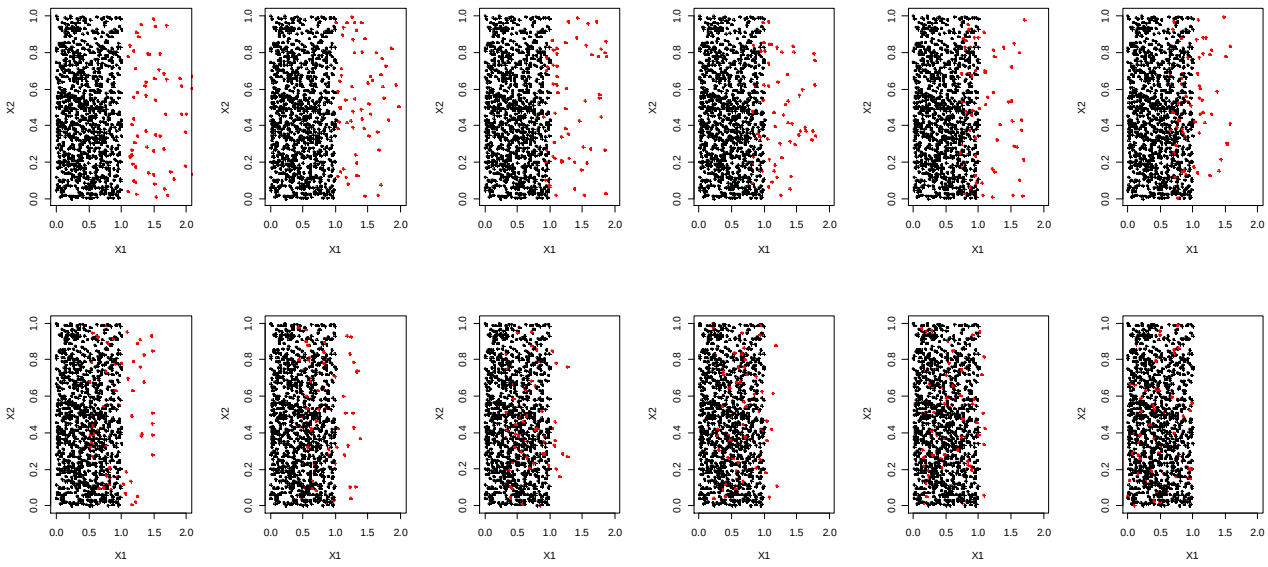
ERRORE COMPLESSIVO CASO (BILANCIAMENTO 50/50)																				
a	SVM Linear			SVM Sigmoid			SVM Polynomial			SVM Radial			Albero Decisionale			Random Forest			Rete Neurale		
	Min	Media	Max	Min	Media	Max	Min	Media	Max	Min	Media	Max	Min	Media	Max	Min	Media	Max	Min	Media	Max
-0,1	-	-	-	0,01	0,04	0,06	0,00	0,02	0,04	-	0,01	0,01	-	-	-	-	-	-	-	-	-
0	-	0,01	0,01	0,03	0,06	0,11	0,01	0,03	0,06	-	0,01	0,04	-	0,00	0,01	-	0,00	0,01	-	0,00	0,01
0,1	-	0,01	0,07	0,05	0,10	0,14	0,03	0,06	0,11	0,01	0,04	0,08	0,01	0,04	0,07	0,01	0,05	0,09	0,01	0,04	0,08
0,2	0,01	0,06	0,13	0,09	0,14	0,20	0,05	0,10	0,15	0,05	0,09	0,14	0,05	0,08	0,11	0,06	0,09	0,14	0,05	0,09	0,13
0,3	0,02	0,11	0,20	0,12	0,20	0,27	0,10	0,14	0,20	0,09	0,13	0,21	0,09	0,12	0,17	0,09	0,14	0,20	0,09	0,13	0,17
0,4	0,07	0,16	0,25	0,21	0,27	0,33	0,12	0,18	0,26	0,12	0,18	0,25	0,11	0,17	0,23	0,15	0,20	0,27	0,11	0,18	0,24
0,5	0,15	0,23	0,32	0,27	0,35	0,45	0,16	0,24	0,32	0,17	0,24	0,29	0,16	0,22	0,27	0,19	0,25	0,31	0,17	0,24	0,30
0,6	0,20	0,29	0,38	0,32	0,42	0,50	0,23	0,29	0,36	0,21	0,28	0,34	0,20	0,27	0,34	0,25	0,30	0,35	0,22	0,29	0,36
0,7	0,24	0,35	0,46	0,41	0,49	0,56	0,28	0,35	0,42	0,28	0,34	0,40	0,25	0,33	0,40	0,29	0,35	0,41	0,27	0,34	0,42
0,8	0,34	0,43	0,51	0,42	0,51	0,60	0,35	0,41	0,49	0,34	0,40	0,48	0,32	0,40	0,47	0,34	0,41	0,48	0,34	0,40	0,45
0,9	0,37	0,49	0,60	0,42	0,50	0,55	0,40	0,46	0,53	0,39	0,45	0,52	0,39	0,45	0,54	0,39	0,45	0,52	0,37	0,45	0,52
1	-	0,51	1,00	0,44	0,50	0,56	0,43	0,51	0,58	0,42	0,51	0,57	0,44	0,50	0,58	0,41	0,50	0,57	NA	NA	NA

Studiando la tabella saltano subito all'occhio i seguenti commenti:

1. tutti i classificatori tendono ad aumentare la percentuale di osservazioni classificate erroneamente all'aumentare del grado di confusione del modello
2. al più alto grado di confusione tutti i modelli arrivano a classificare correttamente solo il 50% dei record del test set azzerando l'utilità del modello di classificazione rispetto al classificatore casuale (sostanzialmente tirare una moneta presenta lo stesso grado di precisione)
3. la rete neuronale al più alto grado di confusione tra gruppi non è in grado di eseguire alcuna classificazione perché l'algoritmo di addestramento non è andato in convergenza
4. tutti i classificatori hanno un comportamento al variare del parametro di confusione simile e non emergono classificatori particolarmente performanti.

Lo stesso sperimento di cui sopra viene quindi riproposto ponendo che n sia pari a 0,06 cosi da poter rappresentare il caso (che affronteremo con maggior dettaglio nella quinta parte) in cui è presente una forte asimmetria nella frequenza dei due gruppi.

Rieseguendo i passaggi descritti sopra otteniamo la seguente situazione:



ERRORE COMPLESSIVO (NESSUN BILANCIAMENTO)																				
a	SVM Linear			SVM Sigmoid			SVM Polynomial			SVM Radial			Albero Decisionale			Random Forest			Rete Neuronale		
	Min	Media	Max	Min	Media	Max	Min	Media	Max	Min	Media	Max	Min	Media	Max	Min	Media	Max	Min	Media	Max
- 0,1	-	0,01	0,12	0,04	0,08	0,13	-	0,00	0,01	-	0,00	0,01	-	-	-	-	-	-	-	0,00	0,00
0	-	0,13	0,30	0,05	0,09	0,12	-	0,00	0,01	-	0,01	0,02	-	0,00	0,01	-	0,00	0,01	-	0,00	0,01
0,1	-	0,27	0,55	0,05	0,09	0,13	-	0,01	0,03	-	0,02	0,04	-	0,01	0,02	-	0,01	0,02	-	0,01	0,03
0,2	0,02	0,42	0,73	0,05	0,08	0,12	0,00	0,02	0,05	0,00	0,03	0,06	0,00	0,02	0,04	0,00	0,02	0,04	-	0,02	0,04
0,3	0,04	0,55	0,83	0,04	0,08	0,12	0,01	0,03	0,06	0,01	0,03	0,07	0,00	0,02	0,06	0,00	0,03	0,05	0,00	0,03	0,05
0,4	0,05	0,73	1,00	0,05	0,08	0,12	0,02	0,04	0,07	0,02	0,04	0,07	0,01	0,03	0,06	0,01	0,03	0,07	0,01	0,03	0,06
0,5	0,06	0,95	1,00	0,04	0,08	0,12	0,01	0,04	0,08	0,02	0,04	0,08	0,01	0,04	0,07	0,01	0,04	0,07	0,01	0,04	0,08
0,6	0,06	0,99	1,00	0,04	0,08	0,12	0,03	0,05	0,08	0,03	0,05	0,07	0,02	0,04	0,07	0,02	0,04	0,07	0,02	0,04	0,07
0,7	0,05	0,99	1,00	0,05	0,08	0,11	0,03	0,06	0,09	0,03	0,05	0,08	0,01	0,05	0,08	0,02	0,05	0,07	NA	NA	NA
0,8	0,05	0,99	1,00	0,06	0,08	0,11	0,03	0,06	0,09	0,03	0,06	0,09	0,03	0,05	0,09	0,03	0,05	0,08	NA	NA	NA
0,9	0,05	0,99	1,00	0,04	0,07	0,11	0,03	0,06	0,08	0,03	0,06	0,08	0,03	0,06	0,08	0,03	0,05	0,08	NA	NA	NA
1	0,05	0,99	1,00	0,05	0,07	0,12	0,03	0,06	0,09	0,03	0,06	0,09	0,03	0,06	0,09	0,03	0,06	0,09	NA	NA	NA

ERRORE NELLA CLASSE DI MINORANZA (NESSUN BILANCIAMENTO)																				
a	SVM Linear			SVM Sigmoid			SVM Polynomial			SVM Radial			Albero Decisionale			Random Forest			Rete Neuronale		
	Min	Media	Max	Min	Media	Max	Min	Media	Max	Min	Media	Max	Min	Media	Max	Min	Media	Max	Min	Media	Max
-0,1	-	0,01	0,12	0,67	0,93	1,00	-	0,01	0,13	-	0,04	0,24	-	-	-	-	-	-	-	0,00	0,05
0	-	0,13	0,30	0,60	0,91	1,00	-	0,08	0,22	-	0,15	0,36	-	0,01	0,12	-	0,01	0,17	-	0,02	0,17
0,1	-	0,27	0,55	0,56	0,91	1,00	-	0,23	0,54	-	0,30	0,55	-	0,16	0,36	-	0,16	0,36	-	0,18	0,36
0,2	0,10	0,42	0,73	0,50	0,90	1,00	0,10	0,40	0,73	0,10	0,45	0,73	-	0,29	0,60	-	0,30	0,60	-	0,30	0,60
0,3	0,22	0,56	0,83	0,57	0,92	1,00	0,17	0,52	0,82	0,22	0,56	0,83	0,09	0,41	0,82	0,09	0,41	0,82	0,06	0,42	0,82
0,4	0,40	0,74	1,00	0,59	0,93	1,00	0,33	0,66	1,00	0,39	0,68	1,00	0,17	0,52	0,85	0,17	0,53	0,85	0,17	0,55	0,85
0,5	0,64	0,95	1,00	0,69	0,91	1,00	0,40	0,78	1,00	0,50	0,78	1,00	0,30	0,62	0,89	0,30	0,62	0,89	0,30	0,64	0,89
0,6	1,00	1,00	1,00	0,72	0,93	1,00	0,65	0,86	1,00	0,60	0,85	1,00	0,38	0,72	0,95	0,38	0,71	0,95	0,38	0,73	0,95
0,7	1,00	1,00	1,00	0,64	0,94	1,00	0,80	0,99	1,00	0,78	0,95	1,00	0,31	0,78	1,00	0,31	0,77	1,00	NA	NA	NA
0,8	1,00	1,00	1,00	0,70	0,95	1,00	1,00	1,00	1,00	0,88	1,00	1,00	0,58	0,87	1,00	0,58	0,87	1,00	NA	NA	NA
0,9	1,00	1,00	1,00	0,83	0,97	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,73	0,98	1,00	0,79	0,95	1,00	NA	NA	NA
1	1,00	1,00	1,00	0,75	0,98	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,91	0,99	1,00	NA	NA	NA

ERRORE NELLA CLASSE DI MAGGIORANZA (NESSUN BILANCIAMENTO)																				
a	SVM Linear			SVM Sigmoid			SVM Polynomial			SVM Radial			Albero Decisionale			Random Forest			Rete Neuronale		
	Min	Media	Max	Min	Media	Max	Min	Media	Max	Min	Media	Max	Min	Media	Max	Min	Media	Max	Min	Media	Max
-0,1	-	-	-	-	0,03	0,07	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0	-	-	-	-	0,03	0,08	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0,1	-	-	-	-	0,03	0,07	-	-	-	-	-	-	-	-	0,02	-	-	0,01	-	-	0,01
0,2	-	-	-	-	0,03	0,08	-	-	-	-	-	-	-	-	0,02	-	-	0,02	-	-	0,02
0,3	-	-	-	-	0,03	0,08	-	-	-	-	-	-	-	-	0,04	-	-	0,02	-	-	0,02
0,4	-	-	-	-	0,03	0,06	-	-	-	-	-	-	-	-	0,01	-	-	0,02	-	-	0,02
0,5	-	-	-	-	0,03	0,07	-	-	-	-	-	-	-	-	0,02	-	-	0,02	-	-	0,02
0,6	-	-	-	-	0,03	0,08	-	-	-	-	-	-	-	-	0,02	-	-	0,02	-	-	0,01
0,7	-	-	-	-	0,03	0,06	-	-	-	-	-	-	-	-	0,02	-	-	0,01	NA	NA	NA
0,8	-	-	-	-	0,02	0,06	-	-	-	-	-	-	-	-	0,02	-	-	0,01	NA	NA	NA
0,9	-	-	-	-	0,02	0,07	-	-	-	-	-	-	-	-	0,01	-	-	0,01	NA	NA	NA
1	-	-	-	-	0,02	0,05	-	-	-	-	-	-	-	-	0,03	-	-	0,02	NA	NA	NA

La simulazione riconferma quanto precedentemente anticipato ossia il fatto che in una situazione di forte squilibrio tra le classi tutti i modelli di cui sopra tendono ad assegnare ogni nuova osservazione che si presenta in fase di test al gruppo di maggioranza garantendo in questo modo un elevato grado di precisione complessiva.

Tutti i classificatori infatti tendono a non fare alcun errore nel gruppo di maggioranza ma sbagliano tutte le osservazioni del gruppo alternativo. Complessivamente però tutti i classificatori vantano un grado di precisione complessiva superiore al 90% (tranne l'SVM lineare) quale che sia il grado di confusione del data base. Questo sostanzialmente perché avendo solamente un 6% di record appartenenti al gruppo di minoranza assegnando tutte le nuove osservazioni alla classe alternativa si ottiene una classificazione corretta nella stragrande maggioranza dei casi.

Questa elevata performance è però evidentemente una informazione distorta in quanto *nessuno dei 7 modelli di cui sopra esegue una effettiva classificazione essendo incapaci di distinguere tra le due classi.*

Quando si presentano situazioni del genere in letteratura vengono proposti due possibili approcci risolutivi, il primo (di tipo data driven) consiste nel ribilanciare la distribuzione tra i due gruppi mentre il secondo (di tipo algoritmico) va ad assegnare un diverso peso all'errore di classificazione dei due gruppi.

Per una più approfondita descrizione di queste due tecniche si rimanda alla quinta parte dell'elaborato dove saranno effettivamente impiegate nella risoluzione del caso pratico.

Questo esempio simulato è stato fatto soprattutto per mostrare come in assenza di una adeguata parametrizzazione del modello tutti i classificatori tendono ad avere un comportamento simile all'aumentare del grado di confusione e asimmetria del data base.

Come si vedrà nel capitolo successivo la scelta di un classificatore rispetto ad un altro è soprattutto guidata dal caso concreto che si sta analizzando e non è possibile stabilire a priori un modello dominante su tutti gli altri.

È quindi opportuno esplorare diverse soluzioni e contemporaneamente analizzare approfonditamente il problema oggetto di studio al fine di poter scegliere quel modello ottimale per il caso oggetto di studio ma non è possibile a priori dire quale dei sette modelli sovra utilizzati risulti essere il più performante.

Conclusione della terza parte

Con l'esperimento con dati simulati si conclude questa terza parte dedicata alla presentazione e confronto di metodologie alternative alle SVM ma comunque appartenenti alla famiglia delle tecniche di machine/statistical learning.

Dobbiamo ora concentrarci su una serie di problematiche che insorgono nelle applicazioni e che sono comuni a tutte e tre le tecniche fin'ora presentate.

Queste possono essere sinteticamente elencate come:

1. Gestione delle feature su cui sono definite le nostre osservazioni campionarie
2. Tecniche di ottimizzazione stocastica su problemi di tipo non lineare
3. Valutazione del risultato di classificazione della macchina addestrata

Come vedremo nelle applicazioni un problema estremamente delicato è quello della gestione delle feature che definiscono ogni osservazione campionaria. Infatti grazie all'avanzamento tecnologico degli ultimi anni la capacità di raccogliere e immagazzinare informazioni è cresciuta in maniera esponenziale fornendo così database estremamente ricchi di informazioni ove per ogni osservazioni di un certo fenomeno sono registrate anche migliaia di caratteristiche differenti (come avviene nell'ambito della bioinformatica).

Ciò causa serie problematiche nelle tempistiche di calcolo e quindi sorge la necessità di:

a) eliminare tutte quelle feature che non danno una effettiva informazione sul fenomeno oggetto di studio
b) costruire delle feature che contengano la maggior parte della informazione campionaria così da ridurre il problema su un minor numero di dimensioni

I due approcci definiscono quello che in letteratura è chiamato come "dimensionality reduction" ossia gestione delle feature campionarie in modo tale da migliorare la performance della learning machine.

Come vedremo nell'apposito capitolo questo problema si pone anche nelle SVM, in particolare la possibilità di gestire le feature in modo tale da ottenere un problema linearmente separabile fa venir meno il problema della "scelta di quale funzione kernel utilizzare" per il quale ricordiamo non esiste alcuna regola generale e quindi si basa sostanzialmente su una scelta soggettiva.

Se quindi il nostro obiettivo è definire quella combinazioni di feature da considerare che massimizzano la performance del classificatore intuitivamente potremmo pensare che basterebbe provare tutte le combinazioni possibili delle caratteristiche di partenza e scegliere quella che da il minor errore out of sample.

Purtroppo quando si calcolano tutte le combinazioni possibili di feature ottenibili si vede subito che questo approccio “a forza bruta” non è computazionalmente possibile. E’ quindi necessario definire una **strategia di ricerca** del miglior set di feature da considerare.

Notiamo il problema di selezione del set di feature migliore è sostanzialmente un problema di ottimizzazione. Dobbiamo immaginare uno spazio delle feature sul quale si definisce una superficie di perdita data dall’errore out of sample ottenuto dalla macchina addestrata su dati con quelle caratteristiche. Purtroppo non è possibile definire una funzione che data un certo set di feature in input mi restituisca l’errore out of sample che otterrei con la macchina addestrata e quindi in sostanza la superficie di perdita definita sullo spazio delle feature non è rappresentabile.

E’ quindi necessario introdurre una serie di metodologie di calcolo algoritmico che ci consenta di “muoverci” su questa ignota superficie di perdita alla ricerca del punto di minimo globale ossia la combinazione di feature che restituisca il minor errore out of sample. Per questo motivo nel secondo capitolo si parlerà nello specifico di metodologie di ricerca numerico stocastica della soluzione ottimale.

Questi metodi sono tipicamente utilizzati anche nell’ambito dell’ottimizzazione di funzioni non lineari ove la soluzione è ottenuta con logica algoritmico iterativa (si veda ad esempio gli algoritmi di Gauss Newton o di Newton Raphson).

Infinte ottenuto la nostra combinazione di feature ottimale ed addestrata la macchina sarà necessario effettivamente valutare come questa si comporta nella classificazione di punti non utilizzati in ambito di training. Introdurremo quindi lo strumento classico di valutazione della performance di un classificatore binario dato dalla ROC curve.

Questa in particolare fornisce una rappresentazione grafica e di immediata interpretazione della capacità della nostra macchina addestrata di classificare correttamente le nuove osservazioni campionarie.

Quarta Parte

1.Dimensionality Reduction: Feature Selection e Feature Extraction

1.1 Introduzione

Con feature selection ed extraction (sinteticamente FS ed FE) si fa riferimento ad un insieme di tecniche dell'ambito del machine e statistical learning volte al filtraggio ed alla manipolazione delle variabili che definiscono i record del training set con l'obiettivo di ottenere dei guadagni in termini di performance nella capacità di generalizzazione e nei tempi di addestramento dei modelli.

Grazie ai più recenti progressi tecnologici degli ultimi anni la massa di informazioni che è possibile raccogliere per un dato fenomeno è aumentata vertiginosamente così come la possibilità di accedere a grandi masse di dati tramite la diffusione che i big data stanno sperimentando attualmente. Ciò introduce un problema di misurazione della rilevanza delle diverse informazioni ai fini della ricerca che si sta eseguendo.

E' di tutta evidenza che in situazioni in cui ogni osservazione viene definita da migliaia di variabili (quindi ogni record è definito un punto in uno spazio n-dimensionale) diventa necessario introdurre un criterio volto a poter individuare ed eliminare tutte quelle variabili poco informative o addirittura ridondanti così da snellire il problema e rendere i risultati del modello più facilmente interpretabili.

D'altro canto il ricercatore potrebbe non essere interessato a definire un subset delle variabili di partenza (che resterebbero comunque un numero considerevole) ma preferisce invece definire delle nuove variabili in grado di contenere la maggior parte della massa informativa data dall'intero training set così da ridurre un problema definito in un iperspazio con moltissime dimensioni ad un problema definito su numero molto contenuto di variabili.

Queste due situazioni fanno riferimento rispettivamente alla FS ed FE, in questo capitolo si fornirà una breve introduzione a quelli che sono i principali metodi di entrambi gli ambiti. E' però nostro interesse focalizzare l'attenzione sui metodi di FS ed FE che sono risultati più performanti nel particolare ambito dei SVM.

Al fine di evitare confusione è bene evidenziare che nell'ambito della FS ed FE il termine feature indica la variabile con cui definiamo una osservazione quindi equivalgono alle variabili di input viste nell'ambito dei SVM. Quando tratteremo le tecniche di FS ed FE per le SVM dovremmo specificare cosa si intende ma per ora parlare di feature o input space è equivalente.

1.2 Feature Selection

Come indicato nell'introduzione l'obiettivo della FS è individuare un sottoinsieme di feature in grado di descrivere il fenomeno oggetto di studio in maniera equivalente (o addirittura migliore) dell'intero insieme di caratteristiche. Più formalmente l'obiettivo della FS è *selezionare il più piccolo subset di feature che realizzi la massima capacità di generalizzazione della learning machine*.

Per fare questo è dunque necessario eliminare tutte quelle variabili che risultano essere poco informative se non addirittura ridondanti (l'informazione che danno è già presente nelle altre feature). Nel perseguire tale obiettivo le tecniche di FS seguono diverse strategie e logiche che possiamo raggrupparle in due macro categorie:

1. Numero di feature considerate ad ogni iterazione:

    Univariato

    Multivariato

2. Rilevanza del classificatore che si sta utilizzando

    Wrapped ed Embedded

    Filter

1.2.1 Metodo Univariato

I metodi univariati di FS si basano sul concetto di variable ranking, in sostanza si analizza ogni singola feature che definisce un record e si costruisce una classifica che ordina queste variabili in base al loro livello informativo. Il punto fondamentale che distingue questi metodi da quelli multivariati sta nel fatto di considerare una sola variabile alla volta ad ogni iterazione.

Si consideri un set di esempi $\{x_k, y_k\}_{k=1,\dots,m}$ ove x_k è un vettore di n coordinate ed assumiamo che sia possibile eseguire una classifica del livello informativo delle n variabili basandosi su una funzione di score $S(i)$ con $i = 1, \dots, n$ calcolata sui x_{ki} e y_k ossia i valori assunti dal training set per la variabile i-esima.

Inoltre assumiamo che un elevato valore di $S(i)$ indichi una variabile molto rilevante per il modello. Grazie a queste ipotesi possiamo definire una classifica delle feature e creare un loro sottoinsieme che elimini tutte quelle ritenute poco rilevanti (ad esempio impostando un valore soglia minimo che deve assumere $S(i)$ perché la variabile sia considerata informativa).

Quello appena descritto è lo schema logico alla base di ogni metodo di FS univariato, ciò che distingue le diverse tecniche che andremo a presentare è la specificazione della funzione di score $S(i)$.

Il primo metodo che tratteremo identifica la funzione di score $S(i)$ con l'indice di correlazione del Pearson sulla base della seguente intuizione: un modo per determinare il potere predittivo di una singola feature può essere quello di verificare l'esistenza di una relazione tra la variabile $\widetilde{X}_i$ ed il vettore delle etichette y. Se ad esempio a fronte di certi valori assunti da $\widetilde{X}_i$ le etichette y tendono ad avere una determinata manifestazione si può immaginare che questa particolare feature abbia una buona capacità predittiva.

Per misurare tale relazione si utilizza l'indice di correlazione lineare del Pearson che si definisce nella seguente equazione

$$\rho(i) = \frac{Cov(x_i, y)}{\sqrt{Var(x_i)Var(y)}} \rightarrow \hat{\rho}(i) = \frac{\sum_{k=1}^{m}(x_{ki} - \bar{x_i})(y_k - \bar{y})}{\sqrt{\sum_{k=1}^{m}(x_{ki} - \bar{x_i})^2 \sum_{k=1}^{m}(y_k - \bar{y})^2}} \quad per\ i = 1, \dots, n$$

ove si osservi come l'indice i indichi il fatto che la statistica venga calcolata considerando le singole "righe" della matrice $X_{n \times m}$ del data set mentre il vettore delle etichette Y è sempre lo stesso.

Questo metodo permette di definire una classifica delle n feature dove le variabili più informative sono quelle che vantano il maggior $\rho(i)$.

Come il lettore avrà già intuito il problema insito in questo metodo sta nella sua incapacità di cogliere relazioni tra le feature ed il vettore delle etichette Y che non siano di tipo lineare. Ciò può portare ad una distorsione nel ranking delle variabili che potrebbe far eliminare dal data set variabili in realtà molto informative nella descrizione del fenomeno (soprattutto in presenza di relazioni complesse non rappresentabili da relazioni lineari).

Per evitare questo problema un altro metodo che viene utilizzato è quello del "Classificatore a singola variabile". Nella sostanza il ranking tra le variabili non è più eseguito cercando di valutare una qualche relazione tra le feature ed il vettore y ma si va a considerare ogni singola feature come un classificatore a se stante e si valuta l'errore che commetterebbe nel test out of sample.

E' di tutta evidenza che ogni feature possa essere vista come un classificatore che si basa su una unica variabile (ove le osservazioni sono rappresentate in uno spazio ad una dimensione) e che quindi classifica i vari records in base a come questi si dispongono rispetto ad un certo valore soglia.

Nonostante questo metodo sia molto intuitivo a livello teorico la sua applicazione comporta una serie di problematiche abbastanza rilevanti, tra queste le più evidenti sono:

1.Come definire il valore soglia?

Dato che stiamo trattando classificatori univariati l'addestramento non dovrebbe essere una task troppo complessa ma va notato che in presenza di un n molto elevato addestrare tutti i singoli classificatori potrebbe essere molto time consuming

2.Come misurare la capacità predittiva di ogni feature

Dato che si stanno trattando principalmente problemi di classificazione uno strumento di valutazione della capacità predittiva della feature potrebbe essere il calcolo della curva ROC sul test set, il doverlo però fare n volte comporta però gli stessi problemi in termini di tempi di calcolo.

3.Come selezionare le feature nel caso ve ne siano molte con lo stesso potere predittivo?

Nel caso in cui tutte le feature risultino essere dei classificatori univariati con una performance simile risulta difficile eseguire un filtraggio di quelle variabili a basso livello.

L'ultimo metodo univariato che verrà esposto è il così detto "Information Theoretic Ranking Criteria" che può essere visto come una sorta di generalizzazione del criterio basato sull'indice del Pearson. L'ITRC si basa su una visione probabilistica del training set definito dalla matrice delle osservazioni $X_{n \times m}$ e dal vettore di etichette $Y_{m \times 1}$. E' infatti possibile vedere ogni riga della matrice X come il vettore delle m manifestazioni della variabile aleatoria che rappresenta l'i-esima feature. Lo stesso Y può essere visto come il vettore delle m manifestazioni della variabile aleatoria che rappresenta la classe delle osservazioni.

E' dunque possibile misurare quanto una particolare feature è informativa basandosi sulla relazione che intercorre tra la sua distribuzione quella Y. Ciò viene misurato tramite una statistica chiamata "mutual information" definita come

$$I(M) = \int \int P(x_i, y)\, log \frac{P(x_i, y)}{P(x_i)P(y)} dxdy \quad per\ i = 1, \dots, n$$

Evidentemente il problema nell'applicare questo metodo sta nel fatto che tipicamente la distribuzione di $\tilde{X}_i$ e di $\tilde{Y}$ non sono note così come non è nota la distribuzione congiunta delle due. Dover stimare quale distribuzione meglio si adatta alle osservazioni x_{ik} per ogni i risulterebbe però molto dispendioso dal punto di vista del tempo macchina e non è detto che le diverse distribuzioni congiunte siano tutte definibili in forma chiusa.

1.3 Metodo Multivariato

Nel precedente paragrafo si sono presentati alcuni tra i metodi più diffusi di FS in ambito univariato. Nonostante a livello teorico tutti e tre i metodi risultino essere molto intuitivi abbiamo visto come la loro applicazione comporti non poche problematiche.

E' però opportuno notare che il principale problema delle metodologie univariate non risiede nella loro difficoltà di effettiva attuazione quanto più nel seguente quesito concettuale: *è possibile che una variabile, che presa singolarmente risulti inutile (poco informativa o ridondante), diventi informativa quando combinata con altre feature?*

Se ciò fosse vero tutti e tre i metodi di FS univariati sopra indicati potrebbero potenzialmente non essere in grado di individuare un subset di feature ottimale perché si basano sul concetto di classifica delle variabili che vengono prese singolarmente.

Per questo motivo i metodi che considerano sottoinsiemi di variabili risultano preferibili ma ciò introduce una nuova problematica. Abbiamo infatti visto nel secondo metodo univariato che ogni feature veniva valutata in base alla sua capacità predittiva, potevamo eseguire tale ragionamento perché andavamo a considerare ogni singola feature come un classificatore a se stante. Ora la nostra valutazione si basa su dei sottoinsiemi delle n variabili di partenza il che pone i seguenti quesiti:

1 come determinare i vari sottoinsiemi di feature da considerare?

Evidentemente considerare ogni possibile combinazione delle n feature farebbe esplodere in maniera esponenziale il numero di operazione da eseguire rendendo i tempi di calcolo insostenibili.

2 come valuto la capacità informativa di ogni sottoinsieme di feature?

Qui entra in gioco il fatto di considerare o meno nella valutazione il particolare strumento di classificazione che si intende usare (SVM, RN, alberi decisionali ecc…)

Come vedremo a breve i metodi di FS multivariati si articolano su tre categorie le quali affrontano con logiche differenti i due problemi di cui sopra, si distinguono metodi:

1 Wrapped

queste metodologie determinano tramite un qualche algoritmo l'insieme dei subset di feature da considerare, lo score ottenuto da ogni subset è determinato dall'errore out of sample ottenuto dalla learning machine addestrata su quel subset.

2 Filter

queste metodologie tipicamente utilizzano metodi di ricerca dei subset di feature simili (se non identici) a quelli adottati nei Wrapped ma a differenza di quest'ultimi la valutazione di ogni sottoinsieme non è funzione della particolare learning machine che si vuole utilizzare ma è calcolata in base ad un criterio di filtro "generico" (ad esempio quelli visti in ambito univariato).

3 Embedded

metodi eseguono la selezione del miglior subset di feature nella fase di addestramento del learning machine utilizzata.

Dato che i metodi Embedded (come ad esempio il Recursive Feature Elimination) sono molto diffusi nell'ambito dei SVM in questa sede focalizzeremo più l'attenzione sui primi due riservando al prossimo capitolo la trattazione del terzo metodo.

Come sopra accennato i metodi Wrapped e Filter si differenziano sostanzialmente in base alla metrica utilizzata nella valutazione della bontà di un certo subset di feature. Tipicamente i metodi Wrapped presentano una complessità computazionale maggiore visto che per ogni subset di dati bisogna:

1. addestrare il classificatore di interesse su quel subset di feature
2. valutare l'errore out of sample eseguito dal classificatore così addestrato

Il metodo utilizza la learning machine come una sorta di black box che restituisce lo score ottenuto da un certo subset di feature ma non pone vincoli sul tipo di macchina da utilizzare. Nella sostanza i Wrapped method possono essere applicati indistintamente ad ogni tipo di learning machine.

I metodi Filter invece non utilizzano alcuna learning machine nel processo di scoring di un dato subset ma si basano su metodi quali ad esempio il coefficiente di correlazione del Pearson od il calcolo della Mutual Information già presentati nell'ambito univariato.

Nonostante i metodi Filter siano computazionalmente meno dispendiosi rispetto ai Wrapped il non aver alcun legame con la specifica learning machine che si vuole adottare può portare a selezionare dei sottoinsiemi di feature non ottimali quando li si utilizza per addestrare il particolare modello scelto dal ricercatore.

Una volta che si è stabilita la metrica di score la ricerca del subset di variabili ottimale può essere invece osservato come un problema di ottimizzazione. L'obiettivo è quindi individuare il punto nello spazio di tutti le possibili combinazioni delle n variabili che massimizza la funzione obiettivo senza dover valutare ogni possibile combinazione.

Data la complessità della funzione obiettivo così come l'alta dimensione dello spazio in cui si definisce l'individuazione del punto di massimo (miglior subset di variabili) è ricavabile con i seguenti metodi:

1. Stochastic Hill climbing
2. Simulated Annealing
3. Genetic Algorithm
4. Greedy Forward Selection
5. Greedy Backward elimination

Ad esempio nello SHC si inizializza l'algoritmo considerando una soluzione candidata (ossia un certo subset di feature) che verrà confrontato con un nuovo subset di variabili selezionato casualmente. Se spostandosi nel nuovo punto si ottiene un guadagno di quota allora ci si muove in quel punto altrimenti si esegue lo spostamento solo in probabilità.

Va notato che questi metodi fanno comunque parte della famiglia delle metodologie di ottimizzazione stocastica le quali non garantiscono l'individuazione di un punto di ottimo globale.

1.4 Feature Selection utilizzando i SVM

Nei capitoli precedenti abbiamo visto come la logica fondamenta dei SVM consiste nell'individuare un iperpiano di separazione che garantisca la massima distanza (margine) tra le osservazioni appartenenti alle diverse classi.

Ovviamente la maggior parte dei problemi reali non presenta la comoda situazione di avere training set linearmente separabili e quindi i SVM utilizzano una mappatura in grado di riscrivere le stesse osservazioni in uno spazio in cui sono effettivamente classificabili tramite un iperpiano.

Formalmente questo spazio viene indicato con il simbolo H (perché si dimostra essere uno spazio di Hilbert) mentre l'input space viene rappresentato con una X. Si ricordi che tipicamente

$$\dim(X) \ll \dim(H)$$

Come accennato nell'introduzione in questa sede risulta necessario specificare quando si sta parlando di feature e quando di variabili definite nell'input space.

L'operazione di mappatura del training set nel feature space comporta il fatto che per ogni dimensione in cui le variabili sono definite originariamente esistono diverse dimensioni corrispondenti in H quindi l'eliminare una dimensione in X comporta l'eliminazione di diverse variabili in H.

Per meglio capire il funzionamento delle metodologie di FS basate sui SVM iniziamo però con un caso molto più semplice in cui le osservazioni sono linearmente separabili già nell'input space che assumiamo definito su due dimensioni così da poterlo comodamente rappresentare graficamente.

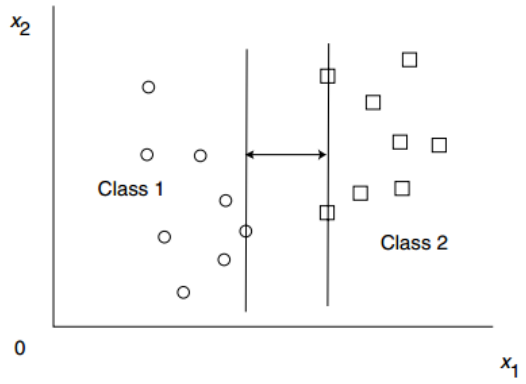
Come facciamo a capire quando una variabile risulta essere poco informativa? Si ricordi che l'obiettivo è definire una retta in grado di classificare correttamente tutti i punti e che contemporaneamente vanta la maggior distanza con i diversi cluster di punti.

Ne segue che una variabile possa essere considerata poco informativa quando:

1. non è in grado di separare correttamente il training set
2. nel caso in cui venga eliminata il *nuovo margine di separazione che si ottiene non è inferiore rispetto a quello ottenuto considerando tutte le variabili*

possiamo cogliere agevolmente quest'ultimo punto con un esempio.

Si consideri un training set rappresentato graficamente qui sotto

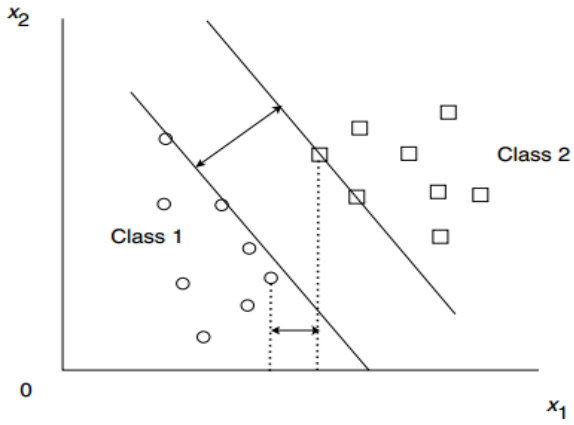


In questo esempio la retta di separazione in grado di ottenere il massimo margine risulta essere parallela all'asse cartesiano della variabile x_2.

Notiamo però che se considerassimo solamente la variabile x_1 riusciremmo comunque a classificare correttamente tutte le osservazioni con una retta cosa impossibile se considerassimo solamente la variabile x_2.

Si può inoltre osservare come il margine della retta di separazione *non cambierebbe* lavorando al problema unicamente in termini di x_1.

Diverso è invece il seguente caso



Qui è evidente che l'eliminazione della variabile x_2 comporterebbe una diminuzione del margine di separazione ottenibile dal classificatore il che ci fa pensare che l'eliminazione della variabile non abbia apportato benefici in termini capacità di generalizzazione del modello

Cos'è che distingue i due casi? Nel primo caso la retta di classificazione è perfettamente parallela ad un asse cartesiano e quindi l'informazione fornita da quella variabile non ha alcuna rilevanza nella determinazione del classificatore.

In generale si dimostra che se l'iperpiano di ottimo non è parallelo all'asse definito da una variabile di input allora eliminare quella variabile farà diminuire il margine e quindi la qualità della soluzione. Operativamente ciò significa che se alcuni elementi di w sono nulli allora eliminare le corrispondenti variabili di input non provocherebbe una variazione del margine di separazione mentre eliminare variabili associate a dei $w_i \neq 0$ lo farebbe diminuire.

Evidentemente nella pratica difficilmente si osserveranno variabili i cui assi sono perfettamente paralleli all'iperpiano di separazione, per questo motivo possiamo definire

un valore positivo soglia ε così da eliminare tutte quelle variabili il cui corrispondente w_i sia minore o uguale ad ε.

Eliminare variabili che non sono perfettamente parallele all'iperpiano comporterà una riduzione del margine di separazione quindi potrebbe anche essere necessario introdurre un livello massimo di riduzione del margine che si ritiene accettabile.

Fondamentalmente in un problema linearmente separabile possiamo eliminare variabili esplicative fintanto che il problema rimane linearmente separabile nel ridotto spazio di input e finché il nuovo margine non si riduca al di sotto di una certa soglia.

Estendendo questa logica al feature space va considerato il fatto che eliminare una variabile nell'input space provoca l'eliminazione di diverse feature all'interno di H.

Anche in questo caso possiamo misurare l'impatto dell'eliminazione di una variabile di input in termini di modifica dell'iperpiano di separazione ma per farlo dovremo considerare le coordinate mappate nel feature space. La variazione di $\|w\|$ a seguito dell'eliminazione di una generica variabile di input $x^{(k)}$ è quindi data da:

$$\Delta^{(k)}\|w\|^2 = \sum_{i,j \in S} \alpha_i \alpha_j \, y_i y_j \left(k\left(x_i, x_j\right) - k\left(x_i^{(k)}, x_j^{(k)}\right) \right)$$

con $x^{(k)}$ vettore ove l'elemento k-esimo è posto pari a zero.

Se più di una variabile ha un $\Delta^{(k)}\|w\|^2 = 0$ possiamo tranquillamente eliminare le variabili di input associate, in alternativa si può decidere di eliminare tutte le variabili con un $\Delta^{(k)}\|w\|^2 < soglia$ anche se la definizione del valore soglia può far sorgere delle problematiche.

1.4.1 Feature Ranking ed Embedded Method

Nel capitolo precedente abbiamo sostanzialmente intuito la modalità con cui nell'ambito dei SVM si identifica una variabile poco informativa distinguendo tra problemi linearmente separabili e non. In sostanza una variabile di input risulta essere poco informativa quando non concorre alla definizione dell'iperpiano di separazione ottimale.

E' possibile capire quanto una variabile $x^{(k)}$ concorra alla soluzione tramite il valore assunto dal rispettivo elemento di w. Se si osserva un $w^{(k)} = 0$ la k-esima variabile non da alcun contributo nel calcolo di $\rho(w, b)$ e quindi la possiamo eliminare senza problemi. Nella pratica però difficilmente accade di osservare iperpiani perfettamente paralleli agli assi di certe variabili quindi è molto più verosimile osservare dei $w^{(k)}$ molto contenuti ma non nulli. Queste variabili concorrono alla definizione di $\rho(w, b)$ ma in maniera molto contenuta.

Sinteticamente possiamo dire che quanto più w^k è piccolo e tanto meno la variabile k-esima è importante quindi è possibile definire una classifica di importanza di tutte le variabili di input prese singolarmente (variable ranking).

Una volta definita questa scala di importanza delle variabili l'obiettivo è definire quel subset che permetta di massimizzare la performance del modello.

E' però evidente che tutti i problemi indicati nel capitolo dei metodi di FS multivariati riemergono anche in questo ambito.

Infatti un approccio che determina tutte le possibili combinazioni di variabili di input, addestrare una SVM per ognuna e quindi selezionare la combinazione di variabili di input ottimale diventa computazionalmente impraticabile all'aumentare delle dimensioni con cui è definito il problema. E' quindi necessario definire una qualche strategia volta all'individuazione del miglior subset di variabili.

Uno dei metodi più semplici è il c.d. Greedy Backward Elimination nel quale si addestra inizialmente la SVM usando tutte le variabili di input disponibili, dopodiché si elimina la prima variabile (ossia quella con il minor valore di w) e si ri-addestra la SVM. Il processo viene iterato fino al raggiungimento di un certo criterio di blocco.

Schematicamente l'algoritmo prevede i seguenti passi computazionali

1. addestro la SVM con tutte le variabili ottenendo un margine di separazione pari a δ_0
2. sia x_k la variabile con il minor $\Delta^{(k)}\|w\|^2$ allora elimino questa variabile
3. addestro la SVM con il subset di variabili ottenendo δ_t, se
    a. $\frac{(\delta_0 - \delta_t)}{\delta_0} < \varepsilon$ con $\varepsilon \in (0,1)$ o

b. il problema non è più linearmente separabile allora stop, altrimenti tornare al punto 2

Il metodo GBE ha il pregio di essere molto intuitivo e facilmente implementabile in un linguaggio di programmazione ma non è propriamente efficiente dal punto di vista computazionale. Infatti se il problema di partenza è definito su n dimensioni il GBE richiede di addestrare n diverse SVM secondo la logica sovra esposta.

Come il lettore avrà intuito una modalità che eviti di dover ripetere n volte il processo di addestramento del modello al fine di ottenere il miglior subset di variabili sarebbe sicuramente preferibile al GBE.

Questo è l'obiettivo che si pongono i così detti metodi Embedded di FS i quali appunto incorporano la fase di FS all'interno della costruzione del modello il quale viene addestrato una sola volta.

L'intuizione alla base del metodo è la seguente: visto che l'obiettivo è definire il più piccolo subset di variabili in grado di ottenere la maggior performance di generalizzazione possiamo impostare il problema di addestramento della SVM in modo tale che venga minimizzato l'errore di classificazione sul training set (come abbiamo visto nei precedenti capitoli) nonché il numero di componenti di w non nulle.

Avremo quindi:

$$\min(1-\lambda)\frac{1}{M}\sum_{i=1}^{M}\xi_i + \lambda\sum_{i=1}^{n}w_i^*$$

$$sub\ y_i(w^T x_i + b) \geq 1-\xi_i \quad per\ i = 1,\dots M$$

ove

- $\lambda \in (0,1)$ è un parametro di regolazione
- $w_i^* = \begin{cases} 1 & se\ w_i \neq 0 \\ 0 & altrimenti \end{cases}$

2.Ottimizzazione stocastica

2.1 Introduzione

Il simulated annealing (SA da ora in poi) è una metodologia di individuazione dei punti di massimo/minimo di una funzione che rientra all'interno della famiglia delle tecniche di ottimizzazione stocastica (come lo stochastic hill climbing o i genetic algorith). Il metodo deve il suo nome al fatto di ispirarsi nel suo meccanismo al fenomeno di riscaldamento di un materiale metallico e successivo raffreddamento controllato.

Senza entrare nei particolari fisici il riscaldamento di un materiale provoca un aumento di energie al suo interno dato dal fatto che gli atomi si possono muovere liberamente aumentando così la frequenza con cui questi si scontrano tra di loro

Nel momento in cui la temperatura del materiale viene fatta calare i movimenti degli atomi diventano sempre più vincolati ad un certo range di variazione. Se il calo della temperatura avviene in maniera lenta e controllata all'interno del materiale si formeranno delle strutture cristallizzate che seguiranno un determinato pattern. Se il calo della temperatura avviene invece troppo velocemente non si creeranno queste strutture e la cristallizzazione avverrà in maniera disordinata.

L'SA cerca di mimare questo comportamento fisico nella risoluzione di un problema di ottimizzazione di una funzione complessa per la quale non è nota la struttura (quindi la presenza di punti di massimo o minimo globale e/o locale).

In questo capitolo verrà esposta una breve introduzione al metodo che non ha alcuna pretesa di essere esaustiva nella trattazione ma che cerca di dare al lettore la nozione fondamentale dell'intuizione che vi è alla base del metodo.

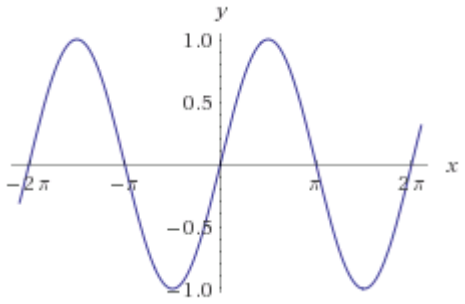
Per questo motivo verranno prima introdotti i metodi di ottimizzazione hill climbing e la sua estensione probabilistica chiamata appunto stochastic hill climbing. Come vedremo il SA risulta essere una sorta di generalizzazione di quest'ultimo metodo.

2.2 Hill Climbing

Si consideri il problema di massimizzazione di una funzione $f(x)$ che per semplicità assumiamo definita su due dimensioni (quindi x è uno scalare).

Supponendo di non poter tracciare il grafico di questa funzione un approccio estremamente semplice ed intuitivo di ricerca del massimo è dato dal hill climbing. Questo metodo ricorsivo cerca di individuare il massimo (o minimo) di una funzione partendo da una soluzione arbitraria impostata dal programmatore. Da lì l'algoritmo cercherà di individuare una soluzione migliore modificando in via ricorsiva la soluzione candidata e spostandosi nella direzione in cui più si verifica il maggior guadagno di quota.

Per meglio comprendere questo meccanismo assumiamo che $f(x) = sen\, x$ e che la soluzione di inizializzazione sia $x = 0$. Graficamente avremo:



Essendo vincolati alle due dimensioni la nostra soluzione può spostarsi solo a destra o sinistra di $x = 0$.

Assumendo che l'algoritmo esegua degli step di 0,1 è evidente che $f(x = 0{,}1) > f(x = -0{,}1)$ e quindi la nuova soluzione candidata diverrà $x' = 0{,}1$.

Da $x' = 0{,}1$ l'algoritmo comparerà nuovamente il livello raggiunto dalla funzione se si aumentasse o diminuisse x' di uno 0,1. Questo viene fatto in maniera ricorsiva fino a che non si arriverà ad un valore di x^* dal quale non sarà più possibile ottenere un guadagno in termini di $f(x)$.

Va notato che nel caso di una funzione definita su più dimensioni l'hill climbing si distingue dalla metodologia dello steepest descent perché nel primo per ogni iterazione viene modificato un solo parametro della funzione a differenza del secondo metodo che calcola il gradiente della funzione obiettivo e quindi si muove verso la direzione che permette il maggior guadagno di quota in base a tutte le coordinate.

Come il lettore avrà già intuito questa metodologia è facilmente implementabile in un linguaggio di programmazione e permette di ottenere una soluzione con dei tempi di calcolo molto ridotti.

D'altro canto è anche evidente l'estrema facilità con cui l'algoritmo possa rimanete intrappolato in punti di ottimo locale.

Infatti se lo step di variazione adottato dall'algoritmo risulta essere troppo contenuto una volta raggiunto un massimo globale può capitare che in ogni direzione si verifichi una perdita di quota facendo concludere al programma di aver raggiunto un punto di massimo o minimo globale.

D'altro canto utilizzare uno step troppo elevato porta ad un possibile loop dell'algoritmo il quale continua ad oscillare nell'intorno del punto di massimo o minimo senza mai raggiungerlo.

Come superare questo problema? Nella sostanza dobbiamo trovare un modo per evitare la trappola dei punti di ottimo locale. Per fare ciò una soluzione molto intuitiva consiste nel permettere all'algoritmo di spostarsi da una soluzione candidata ad un'altra anche quando questa comporti una perdita di quota.

In questo modo l'algoritmo potrebbe uscire dall'intorno del punto di massimo o minimo locale e continuare a muoversi nella direzione che gli permetta di raggiungere maggiori guadagni di quota.

Va però notato che comunque l'obiettivo è trovare un punto di massimo o minimo globale, ne segue che l'algoritmo non potrà sempre andare nella direzione in cui si verifica una perdita di quota perché evidentemente così non si troverà mai la soluzione.

Un metodo quindi consiste nel permettere all'algoritmo di muoversi nella direzione sbagliata solo *in probabilità* come avviene nello stochastic hill climbing o di modificare nel tempo questa possibilità del metodo andando via via a diminuirla (come nello SA).

2.3 Stochastic Hill Climbing e Simulated Annealing

Supponiamo nuovamente di voler calcolare il punto di massimo globale di una funzione ignota che ora assumeremo definita su tre dimensioni ($f(x, y)$).

Anche in questo caso selezioneremo una soluzione di inizializzazione nonché uno step di variazione della funzione da adottare nei vari passi dell'algoritmo.

Definiamo infine con c il punto in cui l'algoritmo è posizionato al momento corrente e con n il punto in cui il metodo potrebbe spostarsi.

A differenza di quanto accade nell'hill climbing deterministico non si andrà a confrontare la variazione di quota che si otterrebbe spostandosi secondo lo step selezionato in ogni direzione possibile ma si selezionerà *casualmente* un punto nell'intorno di c.

Selezionato n l'algoritmo confronta la variazione di quota che si avrebbe nello spostarsi in quel punto, questa formalmente è indicata come:

$$\Delta E = f(n) - f(c)$$

Ricordando che qui stiamo considerando un problema di massimizzazione l'algoritmo seguirà la seguente logica:

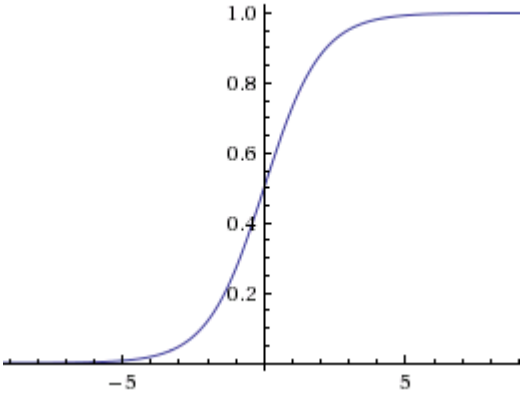
- se $\Delta E > 0$ allora "stiamo salendo" quindi spostarsi in n
- se $\Delta E < 0$ allora spostarsi in n con una certa probabilità

Nella sostanza al posto di rifiutare in toto ogni spostamento che comporti una perdita di quota nello SHC si permette in probabilità di muoversi lungo direzioni che comportano una diminuzione di $f(.)$. Va però notato che l'algoritmo comunque "tende" a muoversi verso direzioni che consentono di aumentare $f(.)$ visto che nel caso di un ΔE positivo ci si sposta sempre in n.

Come il lettore avrà già intuito il problema che ora sorge sta nel definire la probabilità con cui sono ammessi movimenti nelle direzioni che diminuiscono la funzione obiettivo. Intuitivamente un modo per definire questo legame è assumere una qualche relazione tra ΔE e la probabilità di spostamento.

Uno dei primi legami che furono proposti fa utilizzo della distribuzione di probabilità di Boltzmann che quindi definiva una funzione del tipo

$$P(c, n) = \begin{cases} \exp\left(-\frac{\Delta E}{kT}\right) & se\ \Delta E < 0 \\ 1 & altrimenti \end{cases}$$



Volendo inserire una maggior stocasticità nel modello assumiamo che anche i casi in cui $\Delta E > 0$ comportino uno spostamento in probabilità. Possiamo quindi rappresentare il legame tra ΔE e la probabilità di spostamento come:

$$P(c,n) = P(spostarsi\ da\ c\ a\ n) = \frac{1}{1 + e^{-\Delta E}}$$

La funzione sopra definita non è ancora pienamente soddisfacente perché il programmatore potrebbe voler controllare quanto ΔE effettivamente impatti sulla probabilità di spostamento. Per fare ciò abbiamo bisogno di un ulteriore parametro che storicamente viene indicato con T da temperatura visto il legame che il metodo qui presentato ha con l'annealing fisico.

La probabilità di spostamento diviene quindi:

$$P(c,n) = P(spostarsi\ da\ c\ a\ n) = \frac{1}{1 + e^{-\Delta E/T}}$$

Per capire come T impatti su $P(c,n)$ si consideri il seguente esempio: sia $T = 10$ ed $f(c) = 107$ in un problema di massimizzazione; avremo quindi

$f(n)$	ΔE	$P(c,n)$
80	-27	0,062973
100	-7	0,331812
107	0	0,5
120	13	0,785835
150	43	0,986613

si osservi come il passaggio da $f(c) = 107$ a punti che comportano una diminuzione della quota avvenga con una probabilità che è tanto minore quanto più elevata la perdita di quota. D'altro canto variazioni in aumento hanno un probabilità molto accentuata di comportare uno spostamento facendo si che complessivamente l'algoritmo comunque tenda a salire.

Supponendo ora di focalizzare sul caso in cui $f(n) = 120$ vediamo ora come il variare T impatti sulla probabilità di spostamento assumendo che un $T = 1$ sia un valore basso di temperatura.

T	$P(c,n)$
1	0,999998
5	0,930862
10	0,785835
20	0,65701
1E+10	0,5

si osservi come ad alte temperature l'algoritmo assuma comportamenti più stocastici visto che una variazione da 107 a 120 avverrebbe con una probabilità più contenuta man mano che aumenta T. D'altro canto a temperature molto basse l'algoritmo ritorna ad essere quello dell'HC deterministico visto che con $T = 1$ nella sostanza ci si muove sempre e solo verso la direzione che comporta un guadagno di quota.

Graficamente quello che accade aumentando T è che la funzione sigmoide di cui sopra assume una forma sempre più simile a quella di una funzione a gradino che assegna zero ad ogni input inferiore uguale a 0 e uno altrimenti.

E' interessante osservare come il parametro T permetta di controllare il grado di stocasticità che viene ammesso nell'algoritmo. Nella sostanza l'intuizione che è alla base del SA sta proprio nel controllo del parametro T. Al posto di definirlo a priori e mantenerlo costante per tutta la durata del processo (come avviene nello SHC) quello che viene fatto è partire con un livello alto di T e man mano diminuirlo ad ogni iterazione dell'algoritmo fino convergere al comportamento dell'HC deterministico.

Evidentemente l'introduzione di un dinamismo in T rispetto a quanto visto nell'HCS richiede nel SA di definire di quanto varia il parametro ad ogni iterazione del loop. In coerenza con il fatto di ispirarsi al processo fisico di riscaldamento e successivo raffreddamento dei metalli il parametro che governa il comportamento di T viene chiamato *cooling rate* che dovrà essere una qualche funzione con un comportamento monotono decrescente.

L'iniziale alta stocasticità dell'algoritmo dovrebbe permettere alla soluzione candidata di uscire da possibili punti di ottimo locale pur facendolo tendere verso valori più elevati della funzione obiettivo, il successivo raffreddamento dovrebbe invece permettere di convergere verso un punto di massimo.

Non a caso è stato stressato il termine dovrebbe perché il simulated annealing è un metodo di ottimizzazione stocastica che quindi non garantisce l'individuazione di un punto di ottimo globale e potrebbe rimanere comunque intrappolato in punti di ottimo locale.

Il metodo quindi non permette di avere la certezza di aver individuato la miglior soluzione possibile ma comunque da delle soluzioni empiricamente molto buone nelle applicazioni ed è meno suscettibile a rimaner bloccato in punti di ottimo locale come invece accade nell'HC.

3.Valutazione della performance di classificazione: ROC curve

Una volta addestrata la SVM sulla base del set di feature ottimale è necessario valutarne la capacità di classificazione out of sample. Per farlo uno dei metodo più semplici ed intuitivi consiste nel partizionare il data base (quando sufficientemente grande) in due parti così da avere un set di osservazioni da usare per l'addestramento ed un secondo per valutare la performance di classificazione.

In questo modo abbiamo un insieme di osservazioni per le quali è noto la classe di appartenenza ma che non sono mai state "viste" dalla macchina in fase di addestramento, in questo modo possiamo simulare la classificazione di nuove osservazioni potendo però quantificare effettivamente l'errore che verrebbe commesso dalla macchina addestrata.

Ma cosa intendiamo con errore di classificazione? Fin'ora questo concetto l'abbiamo sempre considerato in termini generici ma va evidenziato che nell'ambito della classificazione binaria il tipo di errore che si può commettere è di due tipi: potremmo classificare come 0 un'osservazione che in realtà è 1 o viceversa.

Quindi l'errore è un concetto più complesso del semplice fatto di non classificare correttamente l'osservazione. E' quindi possibile definire in fase di addestramento apposite funzioni di perdita che diano un maggior peso ad un tipo di errore piuttosto che ad un altro. Il peso da assegnare ad ogni tipo di errore non è però definibile a priori ma è funzione del problema specifico che si sta affrontando.

In generale bisognerebbe definire la funzione di perdita in base al *costo* che si sostiene con un particolare tipo di errore. Ad esempio nell'ambito medico un classificatore volto a riconoscere se un paziente è o meno malato potrebbe condurre ai seguenti errori di classificazione:

- classifico sano l'individuo malato allora il soggetto non riceve una cura che gli serve
- classifico malato l'individuo sano allora il soggetto riceve una cura che non gli serve

Il problema è che difficilmente è possibile quantificare questi errori in via ex ante e quindi il peso da dare ad ogni tipo di errore è in gran parte definito dalla prassi o dalla soggettività dello sperimentatore.

Notiamo che il problema del definire la funzione di errore da utilizzare non è affatto una questione minore perché come abbiamo visto nella prima parte *l'intero concetto di apprendimento nel nostro ambito su basa sulla minimizzazione di una funzione di errore generico chiamata* $R(\alpha)$.

3.1 Curve ROC

Dato un classificatore binario che a fronte di un esempio x restituisce una etichetta $y = \pm 1$ nella fase di verifica out of sample si possono avere quattro possibili situazioni.

1. la macchina classifica $+1$ correttamente $\Rightarrow$ vero positivo (TP)
2. la macchina classifica $+1$ non correttamente $\Rightarrow$ falso positivo (FP)
3. la macchina classifica -1 correttamente $\Rightarrow$ vero negativo (TN)
4. la macchina classifica -1 non correttamente $\Rightarrow$ falso negativo (FN)

Possiamo sintetizzare queste quattro casistiche in una tabella che ha sulle righe i valori restituiti dalla macchina addestrata e sulle colonne quelli reali

SINTESI	$y = +1$	$y = -1$	TOTALE
$f(w,b) = +1$	TP	FP	Totale valutati $+1$
$f(w,b) = -1$	FN	TN	Totale valutati -1
TOTALE	1	1	U

Con queste grandezze (tipicamente normalizzate in modo tale da avere una somma per colonna pari a 1) vengono quindi calcolati i seguenti indici

1. $Precision = \frac{TP}{TP+FP}$
2. $Accurancy = \frac{TP+TN}{U}$
3. $Sensivity = \frac{TP}{TP+FN}$
4. $Specificity = \frac{TN}{TN+FP}$

Una volta addestrata la macchina in una classificazione binaria tipicamente si determina un valore sogli s in base al quale si assegna un certo vettore alla classe $+1$ o -1.

Se si modificasse questa soglia evidentemente si avrebbero delle variazione nelle quattro grandezze di cui sopra.

La curva Receiver Operating Charactheristic (ROC) è uno strumento grafico per la valutazione di performance di classificatori binari che ci permette di valutare come i rapporti di cui sopra variano al variare del parametro soglia.

Il grafico considera nell'asse delle ascisse l'indice $1 - Specificity$ e per le ordinate la $Sensitivity$ e quindi per ogni valore di s calcola il punto (x, y) e lo plotta nel grafico generando la ROC curve.

Notare che l'indice $1 - Specificity$ rapporta il numero di classificati $+1$ erroneamente dalla macchina con il totale degli individui che appartengono alla classe -1. Quindi è la percentuale di errore negli individui classificati $+1$. Tale indice viene anche denominato falsi allarmi perché considerando l'indice $+1$ come un evento negativo indica la percentuale di allarmi in verità sbagliati.

L'indice $Sensitivity$ rapporta il numero di classificati $+1$ correttamente dalla macchina con il totale degli individui che appartengono alla classe $+1$. Quindi è la percentuale di correttezza negli individui classificati $+1$. Per la logica di cui sopra tale indice viene anche chiamato percentuali di allarmi veri.

Evidentemente la somma dei due indici da sempre 1 quindi per ogni valore soglia s la ROC curve sarà contenuta nell'area del quadrato definito dai punti (0,0), (0,1), (1,0) e (1,1) perché dato il valore di uno dei due indici (compreso tra 0 e 1) l'altro sarà pari al suo complemento a 1.

Interpretando in chiave probabilistica l'area del quadrato si ricava che l'area sottostante la ROC curve rappresenta la probabilità che la macchina classifichi correttamente un vettore out of sample.

Se quindi l'area sottostante la ROC curve tende al quadrato di lato 1 il modello classifica correttamente l'intero campione out of sample, se la ROC assomiglia alla retta a quarantacinque gradi la probabilità di una corretta classificazione è di circa $1/2$ e quindi è come se la macchina decidesse a quale classe assegnare un vettore lanciando una moneta.

Nello sventurato caso in cui la ROC curve abbia una forma tale da posizionarsi addirittura al disotto della bisettrice il modello è evidentemente un totale fallimento dato che la macchina addestrata ha una percentuale di errore superiore a quella del classificatore puramente casuale.

Sostanzialmente si otterrebbe una previsione più accurata basandosi sul risultato del lancio di una moneta che dando credito ad un modello con una tale forma nella ROC curve.

Conclusione della quarta parte

Con le curve ROC si conclude anche la quarta parte del presente elaborato. In questa sezione l'attenzione è stata focalizzata su tutta una serie di tecniche e strumenti volti soprattutto alla parametrizzazione e valutazione delle performance di un generico modello di classificazione.

Come già accennato nell'esperimento comparativo alla fine della terza parte di questo elaborato la fase di parametrizzazione del modello risulta essere assolutamente fondamentale per garantire una buona performance del classificatore. Infatti si era notato come in assenza di un adeguata ottimizzazione tutti e sette i modelli tendessero ad assomigliarsi nelle performance out of sample, quale che fosse il tipo di classificatore utilizzato.

Le tecniche di ottimizzazione in questa quarta parte saranno quindi fondamentali nella quinta ed ultima sezione di questo elaborato in cui si andrà a applicare quanto presentato nelle fino ad ora in un caso concreto relativo alla frode assicurativa.

In tale applicazione sarà evidente come un'attenta preparazione preliminare del data base ed ottimizzazione dei parametri permetta poi di ottenere performance più che considerevoli anche con modelli di classificazione elementari come può essere una SVM lineare.

Il poter utilizzare una modellizzazione più semplice (grazie alle tecniche di ottimizzazione di cui sopra) garantisce una maggior capacità di generalizzazione del modello (secondo la ben nota diseguaglianza fondamentale presentata nella prima parte) e nella riduzione dei tempi di calcolo richiesti dalla macchina nella fase di addestramento.

Quinta Parte

1. Obiettivi dell'elaborato

In questa seconda parte si propone un'applicazione dell'impianto teorico presentato precedentemente ad un problema di classificazione supervisionato riguardante il settore assicurativo. In particolare si vuole addestrare una support vector machine a classificare un sinistro in due possibili categorie: lecito o fraudolento. A tale scopo utilizzeremo un database contenente una serie di sinistri per i quali è riportato (oltre ad una serie di features che andranno presentate a breve) se vi è stato o meno un comportamento fraudolento da parte dell'assicurato.

Ovviamente un modello per quanto preciso non potrà mai riconoscere con assoluta certezza se un dato sinistro è o meno lecito. D'altro canto un buon classificatore è comunque uno strumento valido nella fase di liquidazione perché "accende un campanello d'allarme" su tutti quei sinistri potenzialmente fraudolenti permettendo alla compagnia di eseguire dei controlli più mirati su singoli casi razionalizzando la gestione del rischio frode. È però opportuno fare il seguente ragionamento: il data base fa una distinzione fra sinistri leciti e fraudolenti ma l'essere fraudolento è a sua volta funzione della capacità di chi ha fornito l'informazione di individuare tale attività criminale. Ne segue che ogni database disponibile sull'argomento è necessariamente distorto in una qualche misura perché verosimilmente all'interno del gruppo di sinistri "legal" vi sarà una percentuale fraudolenta che non è stata correttamente classificata.

La definizione di modelli sempre più evoluti ha anche un ritorno "di lungo periodo" perché permetterà in futuro di generare delle basi di dati sempre più precise e complete che a loro volta saranno fondamentali per evolvere ulteriormente la modellistica utilizzata nello studio del fenomeno della frode assicurativa. Oltre alle SVM verrà presentata la stessa applicazione avvalendosi però di una rete neuronale, un albero decisionale ed una random forest al fine di confrontare la performance dei diversi strumenti nel generalizzare il fenomeno oggetto di studio.

2. Presentazione del DataBase

La fonte informativa dell'elaborato è un data base pubblico fornito dalla società Angoss KnowledgeSeeker Software ed è scaricabile dal seguente link: http://clifton.phua.googlepages.com/minority-report-data.zip. Questo risulta essere l'unico data base pubblico disponibile sull'argomento dell'insurance fraud, ciò non dovrebbe sorprendere data la naturale riluttanza delle compagnie ad ammettere di subire perdite da attività di natura criminale (che tipicamente vengono coperte aumentando semplicemente i premi).

Il data frame è composto da 15.420 sinistri verificati tra il 1994 e 1996 sui quali è stata registrata la presenza o meno di un comportamento fraudolento da parte dell'assicurato. Il dato è di natura cross section visto che ogni osservazione fa riferimento ad una singola unità statistica (polizza) che viene rilevata una sola volta nel periodo di osservazione.

La variabile dipendente è quindi rappresentata dalla voce FraudFound che è che può assumere due livelli: Si, No.

Vista la elevata complessità del set di regressori disponibili per una migliore comprensione si è deciso di articolare le variabili esplicative nei seguenti macrogruppi:

1.Assicurato

Sex : variabile factor con una percentuale maschi-femmine di circa 85,15%

Marital Status: variabile factor suddivisa in 4 modalità: divorziato, sposato, celibe e vedovo. Gli assicurati sposati e single da soli sono più del 98% delle osservazioni.

Age: variabile quantitativa indicante l'età dell'assicurato e che varia tra 16 e 80 con un valor medio di 39 e mediano di 40 (distribuzione quasi simmetrica). Si osserva la presenza di record con età pari a zero ossia da interpretare come valori mancanti. Nel database sono presenti 310 di questi record ossia poco più del 2% di tutte le osservazioni. È però opportuno osservare che all'interno di questi record la percentuale di sinistri fraudolenti arriva risulta essere inferiore all'1% contro il 6% a livello dell'intero portafoglio.

AgeOfPolicyHolder: variabile categoriale suddivisa nei seguenti livelli: 16 to 17,18 to 20, 21 to 25, 26 to 30, 31 to 35, 36 to 40, 41 to 50, 51 to 65, over 65. Confrontando con la variabile Age in certi record non si ha l'equivalenza quindi interpretiamo tale variabile come l'età dell'assicurato da contrapporsi con l'età del soggetto alla guida nel momento del sinistro.

DriverRating: variabile factor suddivisa in 4 livelli: 1, 2, 3, 4 ove il livello 1 indica un guidatore più attento, si osserva una distribuzione quasi uniforme (ogni livello ha una frequenza relativa di circa il 25%)

PastNumberOfClaims: variabile factor suddivisa in 4 livelli: nessuno, uno, da due a quattro e più di quattro. La moda risulta essere il livello "da 2 a 4" con una frequenza relativa del 35% circa seguita da "nessuno" al 28% e "1" al 23%

2.Veicolo

VehicleCategory: variabile factor suddivisa in 3 livelli: Sedan, Utility e Sport. Rappresenta il tipo di veicolo dell'assicurato coinvolto nel sinistro. La categoria Sedan rappresenta poco più del 60% delle osservazioni mentre la Sport è circa al 35%.

VehiclePrice: variabile factor suddivisa nelle seguenti fasce di prezzo:

- less than 20,000 circa il 7%
- 20,000 to 29,000 circa il 52%
- 30,000 to 39,000 circa il 22%
- 40,000 to 59,000 circa il 2%
- 60,000 to 69,000 circa il 0,5%
- more than 69,000 circa il 14%

AgeOfVehicle: variabile factor suddivisa nelle seguenti fasce di età:

- new circa il 2%
- 2 years circa il 0,4%
- 3 years circa il 0,9%
- 4 years circa il 0,15%
- 5 years circa il 8,8%
- 6 years circa il 22%
- 7 years circa il 37%
- more than 7 circa il 26%

Make: variabile factor che presenta 19 marche automobilistiche, si osserva che i livelli Pontiac, Honda, Mazda e Chevrolet insieme formano circa il 90% del totale

3. Sinistro

AccidentArea: variabile dicotomica che assume livelli: Rural e Urban. Indica il tipo di zona in cui si è verificato il sinistro e presenta una forte asimmetria visto che il 90% dei sinistri si verifica in "Urban".

PoliceReportFile: variabile dicotomica che assume livelli: No e Yes. Può essere interpretata come la presenza o assenza della polizia al verificarsi del sinistro. Anche in questo caso si osserva una forte asimmetria visto che il No ha è presente nel 97% dei record.

WitnessPresent: variabile dicotomica che assume livelli: No e Yes. Indica la presenza o assenza di altri testimoni al momento del verificarsi del sinistro. In questo caso nella quasi totalità dei casi (99,5%) non vi sono stati testimoni.

NumberOfCars: variabile factor suddivisa in 4 livelli: 1, 2 vehicles, 3 to 4, 5 to 8 e more than 8. Indica il numero di veicoli coinvolti nel sinistro oltre a quello dell'assicurato, si osserva come il livello "1" da solo costituisca il 92% delle osservazioni.

Fault: variabile dicotomica che assume livelli: Policy Holder e Third Party. Indica l'individuo che ha causato il sinistro. Nel 72% dei records è stato lo stesso assicurato a causare l'incidente.

Data di accadimento e denuncia del sinistro: rappresentate dai campi: Year, Month, WeekofMonth, DayofWeek e MonthClaimed, WeekofMonthClaimed, DayofWeekClaimed. Questi verranno riconvertiti in date al fine di ricavare il numero di giorni che intercorrono tra l'accadimento e la denuncia del sinistro.

Day.Policy.Accident: variabile factor che assume livelli: none, 1 to 7, 15 to 30, 8 to 15, more than 30. Viene interpretato come il lag temporale che intercorre tra la sottoscrizione del contratto e l'accadimento del sinistro. Il livello "more than 30" da solo costituisce il 98% delle osservazioni.

Day.Policy.Claimed: variabile factor che assume livelli: none, 15 to 30, 8 to 15, more than 30. Viene interpretato come il lag temporale che intercorre tra la sottoscrizione del contratto e la denuncia del sinistro. Anche in questo caso il livello "more than 30" da solo costituisce il 98% delle osservazioni.

4.Polizza

PolicyNumber: codice di identificazione univoca di ogni polizza

BasePolicy: variabile factor che riferisce esclusivamente alla modalità di risarcimento stipulate (All perils, Collision, Liability), la distribuzione di queste tre modalità è quasi uniforme.
AgentType: variabile factor che assume livelli: External o Internal. Indica se la polizza è stata o meno stipulata tramite un agente interno alla compagnia. Anche questa variabile presenta una asimmetria molto accentuata con 98,5% di polizze acquisite da agenti esterni alla compagnia.

PolicyType: variabile factor definita secondo le modalità di risarcimento previste dalla polizza e con riferimento al tipo di veicolo assicurato. Si distinguono polizze: All Perils, Collision e Liability. Ogni copertura è poi riferita su tutte le tipologie di veicolo elencate nella sezione VehicleCategory; la categoria di riferimento è Sedan-All perils.

Deductible: variabile che indica la deducibilità del sinistro, nonostante sia una variabile numerica osserviamo come questa assuma solamente valori: 300, 400, 500 e 700. Il 96% dei record presenta un valore di Deductible pari a 400

5.Altro

RepNumber: variabile numerica che assume valori interi da 1 a 16, la distribuzione di questi livelli nei record del data base è quasi uniforme.

NumberOfSuppliments: variabile factor che assume livelli: none, 1 to 2, 3 to 5 e more than 5. L'ultimo livello da solo rappresenta il 45% dei records del data base.

AddressChange.Claim: variabile factor che assume i livelli: no change, under 6 months, 1 year, 2 to 3 years, 4 to 8 years. Si osserva come il livello "no change" da solo costituisca il 92% delle osservazioni.

Come si evince il data base è abbastanza complesso soprattutto per l'elevato numero di variabili factor presenti. In particolare si osservano i seguenti problemi:

PolicyType: questa variabile non è altro che un combinazione delle variabili Vehicle Category e BasePolicy. Considerarla comporta problemi di collinearietà tra i regressori.

MonthClaimed e DayOfWeekClaimed: nel data base è presente un record in cui queste due variabili assumono valore 0 (vi è quindi un missing value). Avendo una sola osservazione (per la quale non è stata individuata la frode) con tale problema si è deciso di eliminarla dal database.

WitnessPresent, AgentType e PolicyReportFiled sono tutte variabili estremamente asimmetriche ove il livello di minoranza ha una frequenza di circa il 3%. Ciò nonostante si è deciso di mantenere invariati tali attributi.

3. Descrizione del programma

Al fine di facilitare la presentazione dell'applicazione proposta andremo a scomporre questo capitolo in cinque sezioni che seguono l'evoluzione logica che si è seguita nella costruzione del modello:

1. Pulizia del data base e correzione dei formati
2. Partizionamento del data base
3. Bilanciamento delle partizioni di training
4. Addestramento dei modelli di classificazione
5. Modelli alternativi di classificazione
6. Rete Neuronale
7. Albero Decisionale
8. Random Forest
9. Confronto tra i modelli

Nelle prossime pagine sarà data una breve descrizione di ognuno degli step di cui sopra riservandosi un maggior livello di dettaglio nel commento dei risultati finali e nel confronto delle performance tra i modelli.

3.1 Pulizia del data base e correzione dei formati

Il primo step del programma ha come input il data base grezzo presentato nel capitolo precedente ed esegue una serie di operazioni di preparazione alla sua successiva trattazione.

Queste operazioni si suddividono a loro volta in tre macro tipologie: correzione degli errori, correzione dei formati e creazione di nuove feature.

Nella correzione degli errori vengono sostanzialmente eliminati dal data base tutti i records che presentano dei missingi value, questi sono stati individuati su tre feature:

1. Età non valida ossia pari a zero
2. DayofWeekClaimed non validi ossia posti pari a 0
3. MonthClaimed non validi ossia pari a 0

La seconda operazione di preparazione consiste invece nel correggere il formato della feature DriverRating in variabile factor. Infine per migliorare la performance del classificatore e contemporaneamente diminuire il numero di variabili esplicative si è deciso di condensare l'informazione di diverse feature creandone di nuove. In particolare sono state create tre nuove feature chiamate: Holder_Driver, Dilazioni Temporali e Coerenza Policy Days.

1. Holder_Driver

Nel data base grezzo sono presenti due campi che fanno riferimento all'età di un individuo: Age (definita puntualmente) ed AgeOfPolicyHolder (definita in un insieme di range di età).

Si è notato come le due variabili non sempre siano tra loro coerenti nel senso che su una data riga del data frame il valore di Age non rientra nell'intervallo definito da AgeOfPolicyHolder.

Questo fatto viene interpretato nel seguente modo: Age fa riferimento all'età del conducente dell'auto assicurata al momento del sinistro mentre AgeOfPolicyHolder è vista come l'età del soggetto assicurato risultante dal contratto. Quando le due osservazioni non sono coerenti allora si presuppone che il soggetto alla guida fosse diverso dall'assicurato.

La nuova feature Holder_Driver è una variabile binaria che assume valore 1 se le due osservazioni sono coerenti e 0 altrimenti.

2. Dilazioni Temporale

Come indicato nel capitolo precedente per motivi di privacy il data base disponibile non riporta esplicitamente le date di accadimento e denuncia del sinistro ma indica nel seguente formato:

$$Il\ giorno\ dell'i-esima\ settimana\ del\ j-esimo\ mese\ dell'h-esimo\ anno$$

Evidentemente un primo lavoro che si è fatto è stato ricostruire la data di accadimento e denuncia del sinistro in una forma più chiara.

In altri studi (vedi minority report) questa ricostruzione è stata fatta per capire se vi sia una maggior tendenza al tentare fenomeni fraudolenti nei fine settimana o durante le diverse festività. In questa sede si è preferito concentrarsi sul delta temporale che avviene tra l'accadimento del sinistro e la sua denuncia alla compagnia da parte dell'assicurato.

3. Coerenza Policy Days

Sempre nel capitolo precedente si è visto come vi siano due feature che riguardano il delta temporale tra due eventi chiamate: Days.Policy.Accident e Days.Policy.Claim.

Tali variabili vengono interpretate come la distanza in giorni tra la sottoscrizione della polizza e l'accadimento/denuncia del sinistro. Le due variabili non danno un valore numerico ma semplicemente dei range definiti in giorni. Se non vi è coerenza tra le due feature significa che il delta temporale tra accadimento e denuncia è stato considerevole quindi si costruisce una nuova feature chiamata coerenza_policy_days di tipo binario che segnala se i due range in giorni sono coerenti o no.

Fatte questi aggiustamenti il data base è pronto per le successive elaborazioni e viene passato allo step successivo. A differenza del data base grezzo il numero di osservazioni è sceso a 15.100 (a causa dell'eliminazione delle osservazioni senza informazione) ed il numero di feature è sceso a 23 grazie soprattutto alla nuova variabile "dilazioni_temporali" che condenza tutta l'informazione contenuta nelle feature di definizione della data di accadimento e denuncia del sinistro.

3.2 Partizionamento del data base

In questo secondo step di programmazione si riceve in input il data base pulito precedentemente e si attua il partizionamento nelle tre categorie: training, test e validazione. A tal fine si è deciso di ripartire l'informazione nel seguente modo: 70% nel training, 15% nel test e altri 15% nella validazione.

Avremo quindi un data base di 10.570 osservazioni per l'addestramento della macchina e altri due di 2.265 records per il test out of sample e la validazione.

Questo partizionamento viene fatto estraendo casualmente senza ripetizione dal data base di partenza e ponendo che tutti i record abbiano la stessa probabilità di estrazione (distribuzione uniforme su interi da 1 a 15.1000).

Il data base di addestramento verrà poi ulteriormente trattato nel terzo step di programmazione relativo al bilanciamento delle due classi (frode e non frode) mentre le partizioni di test e validazione non saranno più modificate. Per ogni partizione si verifica che la percentuali di frodi sul totale rimanga più o meno invariata (all'incirca il 5%).

3.3 Bilanciamento delle partizioni di training e parametrizzazione

Nella partizione di addestramento si può osservare come la percentuale di frodi sul totale delle 10.570 osservazioni sia solamente il 6%. Quando si verifica un tale sbilanciamento nelle due partizioni l'addestramento della macchina su dati "non trattati" può portare ad una tendenza del classificatore ad assegnare sempre l'etichetta del gruppo di maggioranza ad ogni caso out of sample che gli si presenta.

Infatti anche nel caso in cui l'SVM classificasse come non frode ogni osservazione nel test out of sample la performance complessiva della macchina sarebbe verosimilmente superiore al 90% anche se tutte le frodi vengono errate.

Per tale ragione è consigliabile eseguire un bilanciamento della partizione di training così che la macchina non ricada nel comportamento di cui sopra. Nel fare questo bilanciamento si è optato un approccio basato sul ricampionamento casuale del training set. A partire dalla partizione di addestramento originale si definiscono due subset che definiscono gli insiemi di osservazioni: frodi e non frodi.

A questo punto si definisce il nuovo data base di training bilanciato attraverso un

1. sovra campionamento del gruppo di minoranza tramite estrazione casuale con reimmissione
2. sotto campionamento del gruppo di maggioranza tramite estrazione casuale senza reimmissione

i due data frame ottenuti vengono quindi aggregati ottenendo la partizione bilanciata.

Sorge però un problema, in un data frame bilanciato quanto deve essere il peso delle due classi? Non essendo semplice individuare il peso ottimale che le due classi devono avere affinché si addestri un classificatore più performante si è seguito un approccio molto pragmatico "provando" diverse partizioni e verificando la loro performance out of sample,

Tre sono i bilanciamenti provati:

- NULL: ossia nessun bilanciamento è stato fatto sul data base
- 50/50: il numero di osservazioni delle due classi è perfettamente bilanciato
- 20/80: un quinto dei record di training appartiene alla classe fraudolenta

Per valutare l'effetto del modificare il bilanciamento tra le due classi si è deciso di confrontare le performance out of sample di tre identiche SVM addestrate sui tre diversi data base di training. L'addestramento è stato fatto con la funzione svm del pacchetto e1071 utilizzando una parametrizzazione standard in tutti e tre i casi, quindi si è optato per un kernel lineare con parametro C pari a 1.

Il confronto tra le tre SVM addestrate viene fatto sulla partizione di test (definita nello step precedente) e basandosi sulle tabelle di contingenza e le curve ROC per valutare la performance dei tre classificatori.

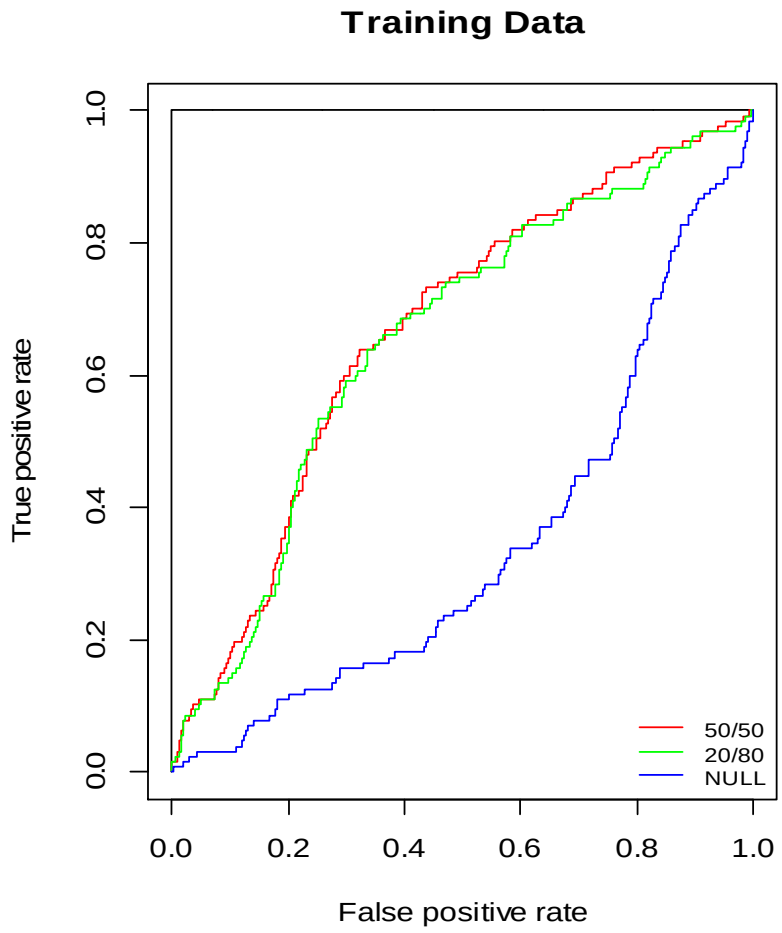
La performance out of sample dei tre classificatori è stata

Partizione 50/50	No	Si
No	1.238	900
Si	7	120

Partizione 20/80	No	Si
No	2.136	2
Si	127	0

Partizione NULL	No	Si
No	2.138	0
Si	127	0

Mentre dal confronto delle curve ROC si è ottenuto



L'attività di bilanciamento sul data base di training ha un impatto rilevante nella performance out of sample del classificatore il quale, in assenza di tale accorgimento, non è in grado di riconoscere i sinistri fraudolenti.

Questa situazione viene definita in letteratura come *classificazione asimmetrica* ossia il caso in cui vi è un forte squilibrio nella numerosità delle osservazioni nelle due classi. A fronte di tale problematica gli approcci tipicamente seguiti sono:

- ribilanciamento dei due gruppi tramite un'attività di sovra campionamento della classe di minoranza e/o sotto campionamento della classe di maggioranza
- assegnare un diverso "costo" dell'errore di classificazione nei due gruppi

La prima strategia equivale a quella presentata poc'anzi e può essere implementata con diverse modalità anche se empiricamente in diversi studi comparativi si è notato come il semplice sovra/sotto campionamento casuale portasse a performance migliori rispetto ad altre strategie di estrazione (ad es. sovra campionare gli esempi positivi situati vicino alla decision function).

Nella seconda modalità si cerca di assegnare un peso che vada ad aumentare la gravità del commettere un errore nel riconoscimento di una specifica classe. Questa strategia può quindi essere adottata in fase di addestramento per migliorare la performance nel classificatore nel riconoscere una certa categoria di record. Nel nostro caso vogliamo definire un set di pesi che permetta di addestrare una macchina in grado di riconoscere i sinistri fraudolenti.

Prima di presentare la modalità con cui si è individuato questo set di pesi ottimali si vuole introdurre un breve ragionamento per capire come interpretare questi valori nel contesto specifico della frode assicurativa che si sta trattando in questa applicazione. Riprendendo la notazione di uno dei più importanti modelli della letteratura attuariale (il Co*llective Risk Model)* possiamo definire il costo aggregato dei sinistri di un portafoglio assicurativo come la somma di un numero aleatorio N di singoli sinistri che provocano un indennizzo Z alla compagnia.

$$X = \sum_{i=1}^{N} Z_i$$

Supponiamo che all'interno di questi N sinistri ve ne siano N^F di tipo fraudolento ed N^{NF} leciti, possiamo quindi immaginare di scomporre la notazione di cui sopra in:

$$X = \sum_{i=1}^{N^F} {Z^F}_i + \sum_{i=1}^{N^{NF}} {Z^{NF}}_i = X^F + X^{NF}$$

dove X^F ed X^{NF} rappresentano il costo aggregato dei sinistri fraudolenti e leciti.

Se valgono le ipotesi del CRM possiamo calcolare il valore atteso del costo complessivo di portafoglio come:

$$E(X) = E(N)E(Z) = E(N^F)E(Z^F) + E(N^{NF})E(Z^{NF})$$

dividendo il tutto per $E(N)$ otteniamo

$$\frac{E(X)}{E(N)} = \frac{E(N^F)}{E(N)}E(Z^F) + \frac{E(N^{NF})}{E(N)}E(Z^{NF}) = f\ E(Z^F) + (1-f)\ E(Z^{NF})$$

A questo punto dobbiamo porci il seguente quesito: quanto costa la frode per l'assicuratore?

Intuitivamente la compagnia ha tutta una serie di costi (soprattutto di tipo indiretto come ad esempio il danno di immagine) che rendono la frode assicurativa una potenziale minaccia per la solvibilità della compagnia.

Non conoscendo a priori questo maggior costo rispetto a quello di un "normale" indennizzo conviene proseguire il ragionamento in termini di proporzione raccogliendo nell'equazione precedente $E(Z^{NF})$ ed ottenendo

$$\frac{E(X)}{E(N)} = E(Z^{NF}) \left[f \frac{E(Z^F)}{E(Z^{NF})} + (1 - f) \right] = E(Z^{NF}) [f \beta + (1 - f)]$$

riportando $E(N)$ a destra dell'equazione otteniamo

$$E(X) = E(N)E(Z^{NF}) [f\beta + (1 - f)] = E(N)E(Z^{NF}) \alpha$$

Quest'ultima scrittura può essere interpretata nella seguente maniera: il primo fattore che compare $E(N)E(Z^{NF})$ rappresenta il costo aggregato di tutti i sinistri nel caso in cui non vi sia alcun fenomeno fraudolento nel portafoglio oggetto di studio.

A questo si va a moltiplicare un fattore α che (nel caso in cui f e β siano strettamente maggiori di 1) va ad amplificare il costo complessivamente sostenuto dalla compagnia. In questa formulazione è evidente come il costo della frode sia rappresentato dal fattore β che può essere interpretato come il peso da assegnare alla classe sinistri fraudolenti nella fase di addestramento della macchina.

Riscalando il tutto rispetto a $E(N)E(Z^{NF})$ dal commento di cui sopra capiamo che se la percentuale di sinistri fraudolenti f si attesta nell'intorno di un 6% come nel nostro caso l'avere un β ad esempio pari a 5 significa che l'incapacità della SVM a riconoscere le frodi non causa alla compagnia un costo di 6% ma di $6\% * 5 = 30\%$.

Ai fini dell'addestramento della SVM il nostro obiettivo è quindi definire quel valore di β che massimizzi la capacità del classificatore di discriminare tra le due classi. Nel percorrere questa seconda strategia si è quindi deciso di utilizzare come data base di addestramento quello originale ossia senza alcuna azione di bilanciamento.

Innanzitutto si è dovuto scegliere il kernel da utilizzare, per farlo si sono addestrate 4 differenti SVM utilizzando come funzioni kernel:

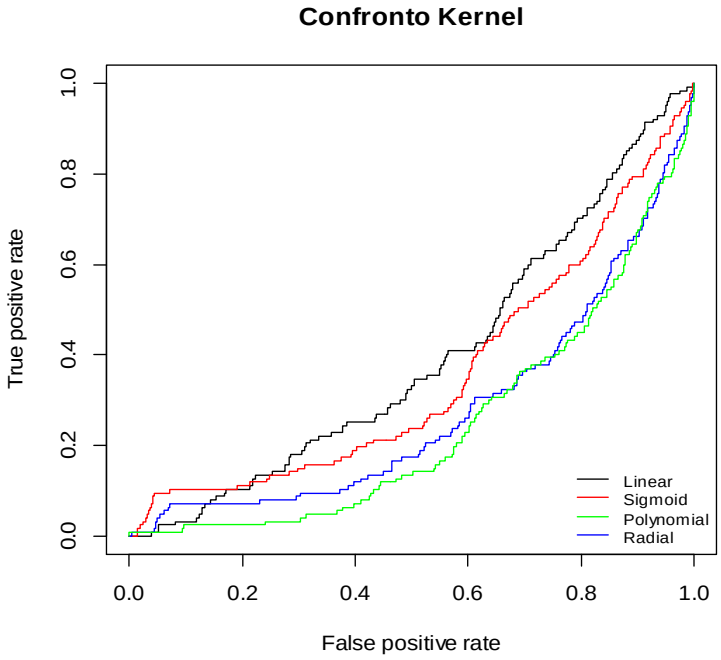
1. Radial $k(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right)$
2. Polynomial $k(x_i, x_j) = (\alpha {x_i}^T x_j + c)^d$
3. Linear $k(x_i, x_j) = {x_i}^T x_j + c$
4. Sigmoid $k(x_i, x_j) = tanh(\alpha {x_i}^T x_j + c)$

queste vengono confrontate nella loro performance out of sample tramite una serie di indicatori quali le curve ROC e le tabelle di contingenza.

I risultati ottenuti da questa prima analisi sono riassunti nella seguente tabella

Kernel	Percentuale Errori Si	Percentuale Errori No	Percentuale di Errore Totale	Area sotto ROC
Linear	1	0	0,05607064	0,395446477
Sigmoid	1	0	0,05607064	0,354640071
Polynomial	1	0	0,05607064	0,275579503
Radial	1	0	0,05607064	0,244370705

mentre il confronto delle curve ROC ha portato a



Si osserva come nessuno dei kernel proposti è in grado di riconoscere il caso dei sinistri fraudolenti, si è quindi optato per il kernel lineare in quanto computazionalmente meno gravoso e perché presenta la migliore curva ROC tra le quattro ottenute.

Scelto il kernel è necessario impostare il peso da assegnare alla classe di minoranza. Visto che l'obiettivo è capire di quanto pesare gli errori di classificazione nelle frodi rispetto ai sinistri legal poniamo un peso di riferimento unitario alla classe di maggioranza mentre il peso dei si viene fatto variare tra 1 e 16 (infatti 1/16 è all'incirca pari al 6% ossia la frequenza con cui è presente la classe di minoranza nel data base di addestramento).

Vengono quindi addestrate 16 SVM identiche tra loro salvo il peso da associare agli errori di classificazione nei Si. Il confronto tra le diverse SVM avviene osservando le relative performance out of sample tramite una serie di indicatori che andremo a descrivere.

I risultati di questa prima analisi vengono riassunti nella seguente tabella

Yes	No	Totale	FP	FN	ROC	Errore Pesato 1	Errore Pesato 2
1	1	0,05607	1,00000	-	0,39545	5,05757	2,67058
2	1	0,05695	1,00000	0,00094	0,32615	6,12650	3,23501
3	1	0,05651	0,99213	0,00094	0,28715	6,87646	3,61820
4	1	0,05651	0,99213	0,00094	0,21374	9,23809	4,86082
5	1	0,06269	0,99213	0,00748	0,19891	9,86202	5,18937
6	1	0,17925	0,54331	0,15762	0,20477	3,37965	1,60178
7	1	0,18940	0,46457	0,17306	0,20020	2,74500	1,37055
8	1	0,18940	0,45669	0,17353	0,19961	2,68969	1,35060
9	1	0,18985	0,45669	0,17399	0,18541	2,89417	1,45495
10	1	0,38366	0,07874	0,40178	0,18998	0,96376	2,56881
11	1	0,38499	0,07874	0,40318	0,21415	0,85924	2,28934
12	1	0,38764	0,07874	0,40599	0,22353	0,83136	2,21330
13	1	0,39514	0,07874	0,41394	0,20769	0,92013	2,44361
14	1	0,39514	0,07874	0,41394	0,24621	0,77616	2,06127
15	1	0,40353	0,05512	0,42423	0,20935	0,91407	2,57833
16	1	0,40353	0,05512	0,42423	0,20981	0,91208	2,57272

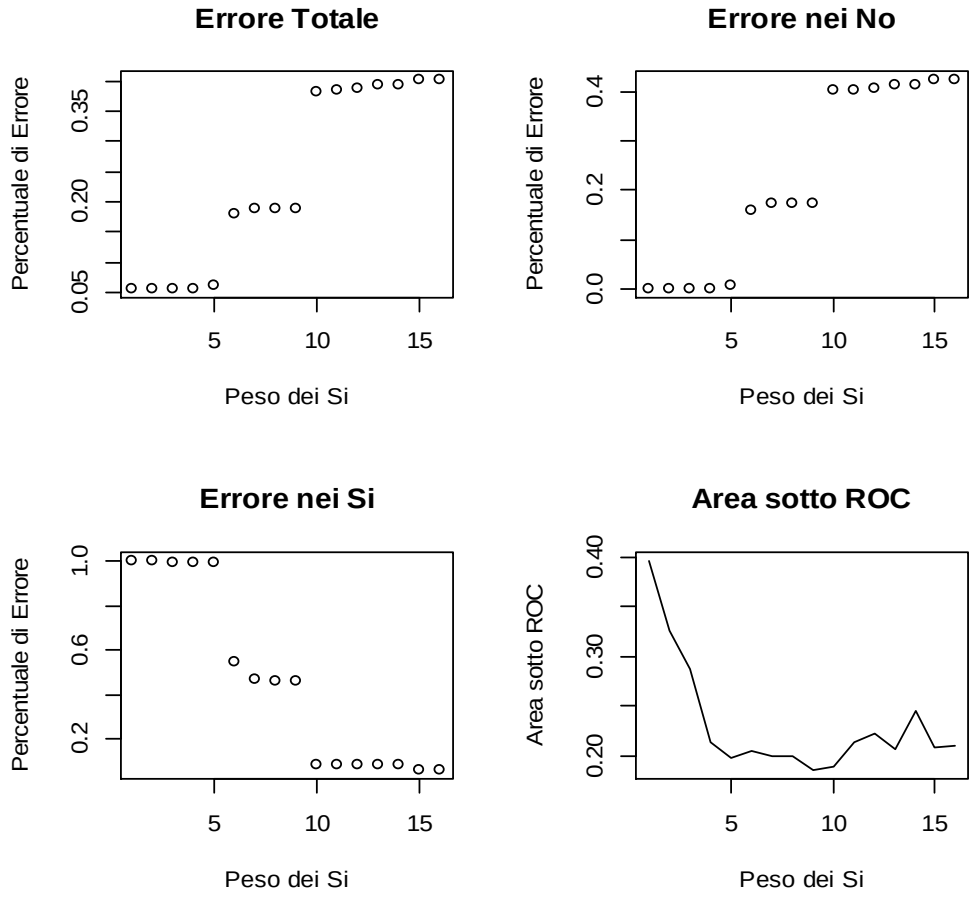
Da questa tabella si possono fare le seguenti considerazioni:

1. al variare del peso assegnato ai False Positive (FP) varia sensibilmente la performance del classificatore nel riconoscere i casi di frode assicurativa

2. vi è un evidente trade off tra i FP e FN in particolare al migliorare della performance nel riconoscere i Si crolla la capacità della macchina nel riconoscere in No, la variazione di performance nel riconoscimento delle due classi al variare del set di pesi è una funzione a gradini

3. l'area sotto la curva ROC (tipico parametro utilizzato per valutare la performance di un classificatore) non è sufficiente a stabilire il miglior peso da assegnare alla classe dei Si perché si osserva come nel caso di classi così fortemente sbilanciate la situazione di partenza (nessun peso specifico) risulti più performante dal punto di vista della ROC nonostante non venga fatta alcuna vera classificazione da parte del modello.

Per una più semplice analisi possiamo rappresentare graficamente la tabella riassuntiva di cui sopra.



Dato l'ultimo punto evidenziato si è deciso di costruire una serie di indicatori di performance che possano sostituirsi alla classica misura dell'area sottostante la curva ROC.

Nella costruzione di questi indicatori l'obiettivo era forzare una sorta di bilanciamento tra i FP e FN evitando situazioni di eccessiva propensione della macchina a riconoscere una sola classe di sinistri. Quindi l'idea è quella di minimizzare la differenza tra FN e FP, d'altro canto potremmo avere che entrambi i tipi di errore sono molto elevati ma simili tra loro (ad es. FN = 0.95 e FP = 0.85) quindi per penalizzare questa situazione si somma alla differenza tra FN e FP l'errore complessivo. Infine si divide il tutto per l'area sottostante la ROC al fine di penalizzare la situazione in cui questa fosse troppo contenuta.

Le misure di errore definite secondo questi principi sono quindi:

$$Errore\ Pesato_1 = \frac{(FP - FN)^2 + FP}{AREA\ ROC}$$

$$Errore\ Pesato_2 = \frac{(FP - FN)^2 + TOT}{AREA\ ROC}$$

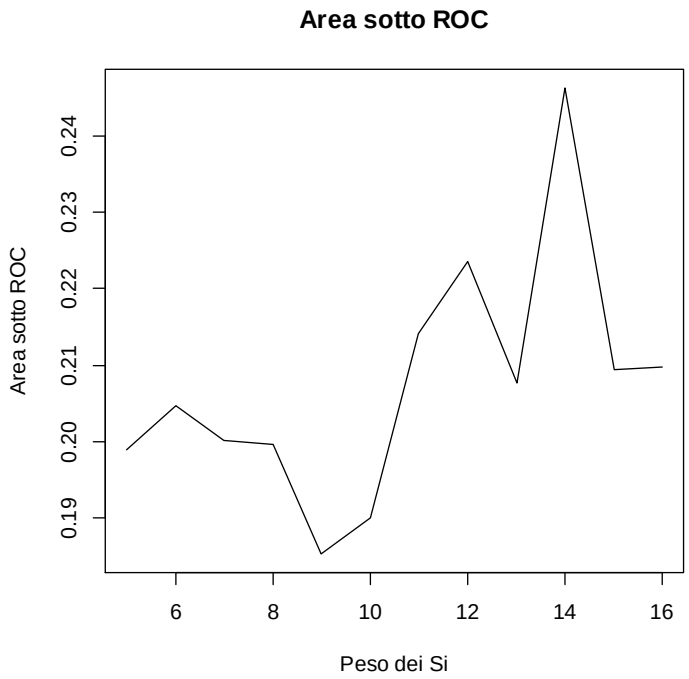
Selezioniamo quindi il set di pesi che minimizza una delle due misure di errore di cui sopra, dal primo indicatore otteniamo come set ottimale la coppia

$$Yes = 14\ e\ No = 1$$

mentre la seconda misura di errore (meno sensibile alla performance nel riconoscere le frodi) definisce un ottimo in

$$Yes = 9\ e\ No = 1$$

Visto che il nostro principale obiettivo è migliorare la performance della macchina nel riconoscimento dei sinistri fraudolenti scegliamo la coppia proposta dal primo indicatore di performance a cui è associata ad una macchina che commette solo un 8% nei FP contro il 46% che si adotterebbe utilizzando l'altra coppia di pesi. D'altro canto se focalizziamo l'attenzione sui soli casi in cui l'SVM commette un errore di classificazione in entrambe le categorie di sinistri (ossia tutti i casi con un peso nei Si tra 5 e 16) notiamo come la coppia 14/1 sia quella che massimizza l'area sotto la ROC.



Sulla base di tutte queste considerazioni si è deciso di adottare come set di pesi ottimale la coppia 14/1 a cui è associata una tabella di contingenza

Partizione NULL con pesi	No	Si
No	1.253	889
Si	10	117

mentre la performance di partenza senza alcuna pesatura era

Partizione NULL	No	Si
No	2.138	0
Si	127	0

Riprendendo la formulazione di cui sopra e sostituendo i valori osservati nel nostro caso concreto abbiamo

$$E(X) = E(N)E(Z^{NF}) \ [f\beta + (1-f)\,] = E(N)E(Z^{NF}) \ [0.06 * 14 + 0.94\,]$$
$$= 1.78\, E(N)E(Z^{NF})$$

ove 14 è da interpretare come il peso da assegnare all'errore nel riconoscimento delle frodi che porta al massimo grado di separazione tra le due classi.

Individuato il set ottimale di pesi da utilizzare e la funzione kernel le successive operazioni di parametrizzazione consistono nella ricerca di un valore ottimale del parametro C e di un subset di feature che massimizzi la performance out of sample.

In tale ricerca si è deciso di adottare in entrambi i casi la tecnica di simulated annealing applicata su una funzione obiettivo data dalla performance out of sample della macchina (misurata con il primo indicatore di errore pesato) la quale varia a seconda di dove ci poniamo nello spazio dei parametri e delle variabili. Per quanto riguarda il parametro di costo complessivo C l'algoritmo non ha individuato nessun valore che desse un errore pesato inferiore a quello di inizializzazione. Per tale motivo si è deciso di tenere il parametro C pari ad 1.

Per inizializzare l'algoritmo nella ricerca del miglior subset di feature si è innanzitutto calcolato la funzione obiettivo su 16 punti nello spazio delle variabili definiti casualmente.

L'algoritmo di SA è stato quindi inizializzato a partire dal punto nello spazio delle feature a cui corrispondeva un minor errore pesato del primo tipo. La tecnica di SA non è andata a

convergenza e quindi non è stata trovata una soluzione preferibile a quella di inizializzazione.

Ciononostante a questa corrispondeva la seguente tabella di contingenza

Partizione NULL con FS	No	Si
No	1.239	899
Si	7	120

in cui si osserva una miglior performance nel riconoscimento della frode senza andare a peggiorare la quantità di errore nei No. Con la feature selection si chiude l'attività di parametrizzazione della SVM.

Ricapitolando il modello ottimale che è stato scelto presenta

1. kernel: linear
2. peso nei Si pari a 14
3. $C = 1$
4. Feature utilizzate (1, 4, 5, 6, 7, 9, 10, 12, 15, 18, 19, 21, 24)

Per avere una migliore stima di come questo modello si comporti nella performance out of sample è stata eseguita una k - fold Cross Validation utilizzando 10 partizioni.Nelle due tabelle sottostanti si riporta la performance ottenuta nelle 10 valutazioni out of sample.

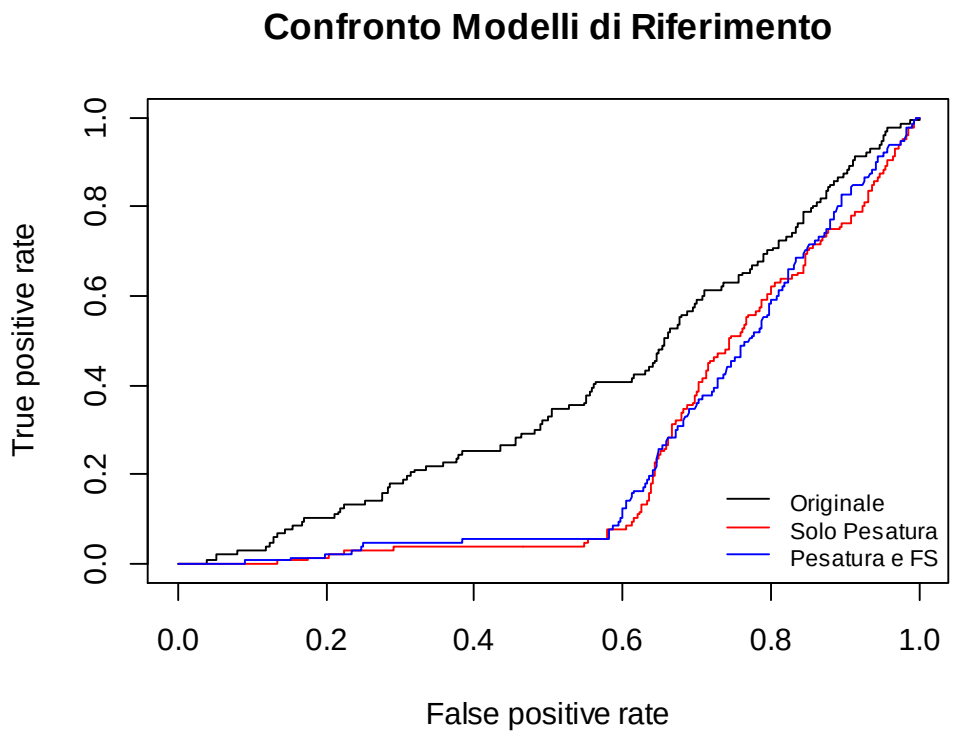
Totale	FP	FN
0,436	0,066	0,459
0,413	0,043	0,438
0,404	0,048	0,430
0,389	0,067	0,410
0,400	0,010	0,426
0,392	0,038	0,412
0,389	0,049	0,409
0,405	0,047	0,426
0,405	0,070	0,425
0,403	0,043	0,427

Statistica	Totale	FP	FN
Minimo	0,389	0,010	0,409
1st QU	0,394	0,043	0,415
Mediana	0,404	0,047	0,426
Media	0,404	0,048	0,426
3rd QU	0,405	0,062	0,429
Massimo	0,436	0,070	0,460

Guardando il valor medio si riconferma quanto osservato in fase di parametrizzazione del modello in termini di errore nelle singole classi e complessivo.

Grazie all'attività di parametrizzazione del modello si è ottenuta una performance di modello nel riconoscimento dei Si più che soddisfacente a discapito però di un forte aumento nel rate dei FN che fa elevare la percentuale complessiva di misclassificazione del modello al 40%. Va notato come la performance di partenza del modello senza parametrizzazione fosse del 94% ma è anche vero che in quel caso non veniva fatta alcuna classificazione dalla macchina che assegnava ogni osservazione out of sample alla classe di maggioranza. Quindi non è corretto confrontare il 60% di precisione ottenuto con il 94% di partenza non essendo quest'ultima una vera performance.

Infine per quanto riguarda le curve ROC possiamo osservare come il confronto tra il modello di partenza e quelli parametrizzati evidenzi un forte peggioramento nell'area sottostante la curva ROC



Per tale motivo si è deciso di scartare il modello sovra descritto e di rieseguire la procedura di parametrizzazione sovra descritta basandosi sul data base di addestramento ribilanciato in modo tale che il numero di Si e No fossero equivalenti (bilanciamento 50/50).

Rieseguendo gli step sovra presentati si è individuato il seguente modello:

1. kernel: radial
2. peso nei Si pari a 1
3. $C = 1$ e $Gamma = 1$
4. Feature utilizzate c(1, 5, 12, 14, 18, 21)

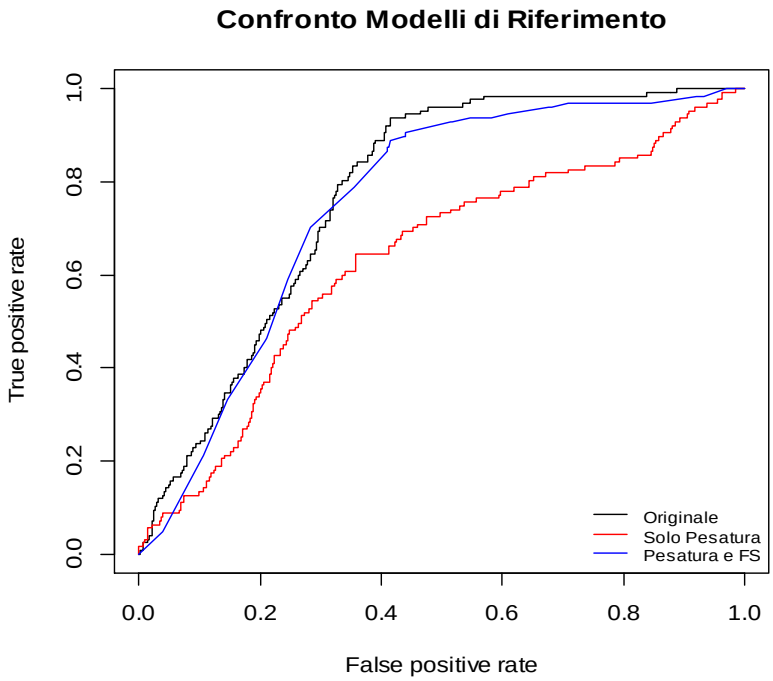
dalla k - fold Cross Validation si è ottenuto

Totale	FP	FN
0,293	0,371	0,218
0,301	0,384	0,216
0,306	0,389	0,224
0,295	0,293	0,296
0,299	0,354	0,243
0,309	0,378	0,239
0,293	0,375	0,211
0,344	0,267	0,348
0,349	0,256	0,355
0,336	0,141	0,348

Statistica	Totale	FP	FN
Minimo	0.2934	0,1413	0,2107
1st QU	0.2957	0,2738	0,2192
Mediana	0.3036	0,3627	0,2407
Media	0.3125	0,321	0,2697
3rd QU	0.3290	0,3775	0,3353
Massimo	0.3490	0,3893	0,3546

Il nuovo modello ottiene un miglior equilibrio tra i due livelli di errore aumentando la performance complessiva all'incirca del 10%.

Il confronto delle curve ROC mostra una situazione molto più incoraggiante rispetto a quella del modello senza bilanciamento



In particolare si osservi come l'attività di feature selection permetta un forte miglioramento della curva ROC rispetto al caso in cui solo l'azione di pesatura delle classi è eseguita. Si osservi come in questo caso il set di pesi ottimale fosse dato dalla coppia 1/1 il che evidenzia come l'effetto ottenuto sulla performance out of sample dall'azione di bilanciamento del data base di training si sostituisce alla selezione del peso ottimale eseguita nel primo modello.

Infine si è voluto provare una combinazione delle due tecniche sovra proposte partendo dal data base bilanciato 20/80. Rieseguendo la parametrizzazione sovra descritta si è ottenuto il seguente modello:

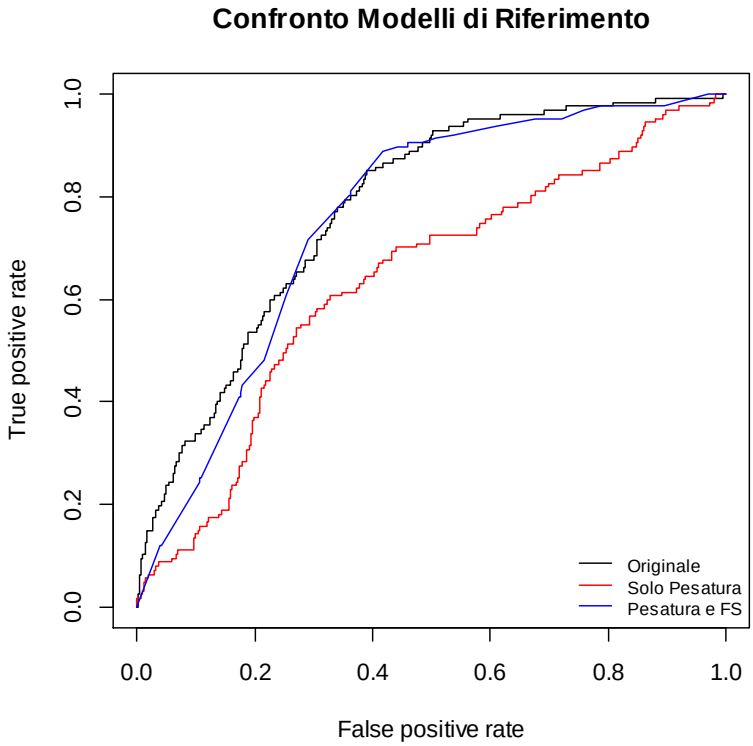
1. kernel: radial
2. peso nei Si pari a 8
3. $C = 1$ e $Gamma = 1$
4. Feature utilizzate (1, 5, 12, 14, 18, 21)

dalla k - fold Cross Validation si è ottenuto

Totale	FP	FN
0,362	0,069	0,431
0,335	0,081	0,404
0,348	0,092	0,415
0,351	0,052	0,421
0,329	0,052	0,400
0,371	0,073	0,441
0,330	0,063	0,398
0,405	0,070	0,426
0,401	0,163	0,415
0,395	0,076	0,415

Statistica	Totale	FP	FN
Minimo	0,3291	0,05178	0,3980
1st QU	0,3382	0,06418	0,4067
Mediana	0,3563	0,07134	0,4152
Media	0,3626	0,07902	0,4167
3rd QU	0,3887	0,07958	0,4245
Massimo	0,4053	0,16279	0,4411

Si osserva una performance molto più simile a quella del primo classificatore rispetto al quale si ha una lieve diminuzione nella capacità di riconoscere i Si ma contemporaneamente una diminuzione dell'errore complessivo di circa il 5%. Dal confronto delle curve ROC si osserva invece una situazione molto più simile a quella del modello bilanciato al 50/50 infatti si ha che:



Dalle osservazioni di cui sopra possiamo individuare l'ottimo nel modello "di mezzo" ossia quello che sfrutta sia il bilanciamento ex ante del data base di training sia un set di pesi asimmetrico per l'errore di classificazione.

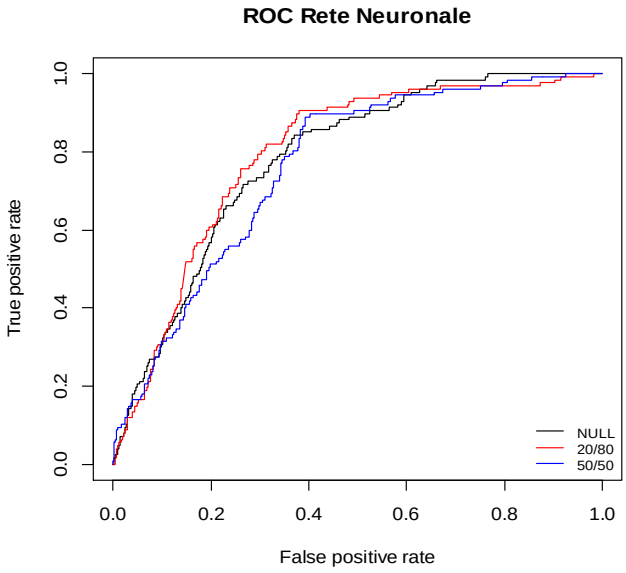
Nonostante tutto però si osserva come la performance complessiva del modello sia tutt'altro che soddisfacente con un grado di precisione che si attesta nell'intorno del 63,74%. Per tale motivo si è voluto rieseguire l'applicazione di cui sopra utilizzando altri modelli di classificazione così da poter confrontare le diverse performance di classificazione.

I modelli di confronto adottati sono: una rete neuronale con uno strato nascosto, un albero decisionale ed una random forest.

Per ognuno di questi classificatori si è confrontato inizialmente la performance out of sample addestrando la macchina sui tre data base con diversi bilanciamenti. Si è selezionato il miglior modello dei tre e quindi si è stimata la sua capacità generalizzazione del fenomeno tramite la tecnica della k - fold cross validation.

Rete Neuronale

Il primo modello di confronto adottato è una rete neurale caratterizzata da un unico strato nascosto di calcolo. Il confronto nella performance out of sample delle tre reti neuronali addestrate sui data base con differente bilanciamento ha portato a:



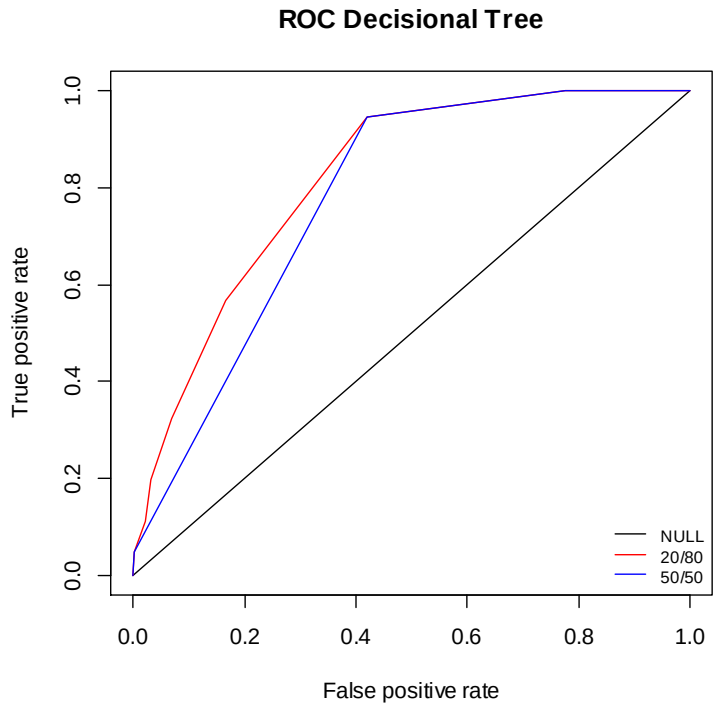
Si osserva come il bilanciamento del data base non sembra avere effetto sulla RN la quale ricava una ROC accettabile (sopra la bisettrice) anche nel caso di totale assenza di bilanciamento. Questo è un evidente punto di forza delle reti neurali che non seguendo la logica di massimizzazione del margine di separazione (come fanno le SVM) non sembrano avere la problematica di impostare un margine asimmetrico nella fase di parametrizzazione del modello.

Selezionato il data base 50/50 la k - fold cross validation ha portato ai seguenti risultati:

Statistica	Totale	FP	FN
Minimo	0,2417	0,1196	0,4911
1st QU	0,2772	0,3123	0,4999
Mediana	0,3215	0,52324	0,7713
Media	0,5703	0,5957	0,6770
3rd QU	1,0000	1,0000	0,7892
Massimo	1,0000	1,0000	0,8464

Albero Decisionale

Il secondo modello di confronto è un singolo albero decisionale, questo viene addestrato sui tre data base di training portando al seguente confronto tra le curve ROC



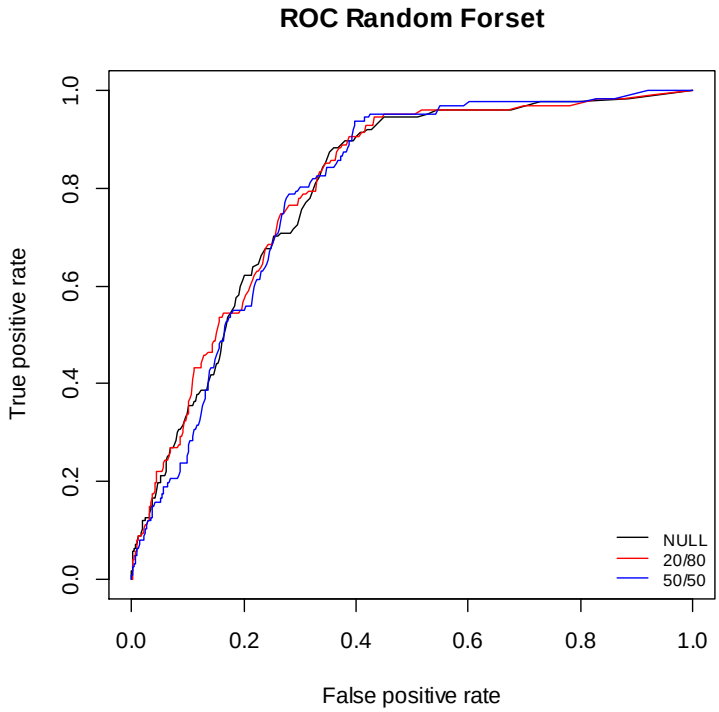
In questo caso si osserva come l'assenza dell'attività di bilanciamento porti il modello ad una ROC tutt'altro che soddisfacente anche se migliore rispetto al caso delle SVM suggerendo una minor sensibilità degli alberi decisionali a forti squilibri nella numerosità delle due classi.

Selezionato il data base 20/80 la k - fold cross validation ha portato ai seguenti risultati:

Statistica	Totale	FP	FN
Minimo	0,2298	0,09302	0,2000
1st QU	0,2503	0,19488	0,2146
Mediana	0,2778	0,23821	0,2487
Media	0,2750	0,25017	0,2589
3rd QU	0,2911	0,32844	0,2973
Massimo	0,3417	0,39060	0,3567

Random Forest

Infine l'ultimo modello di confronto è ottenuto tramite la aggregazione di diversi alberi decisionali tramite la tecnica del bagging. Come sempre il primo confronto è fatto con le curve ROC sui tre modelli addestrati con i data base bilanciati



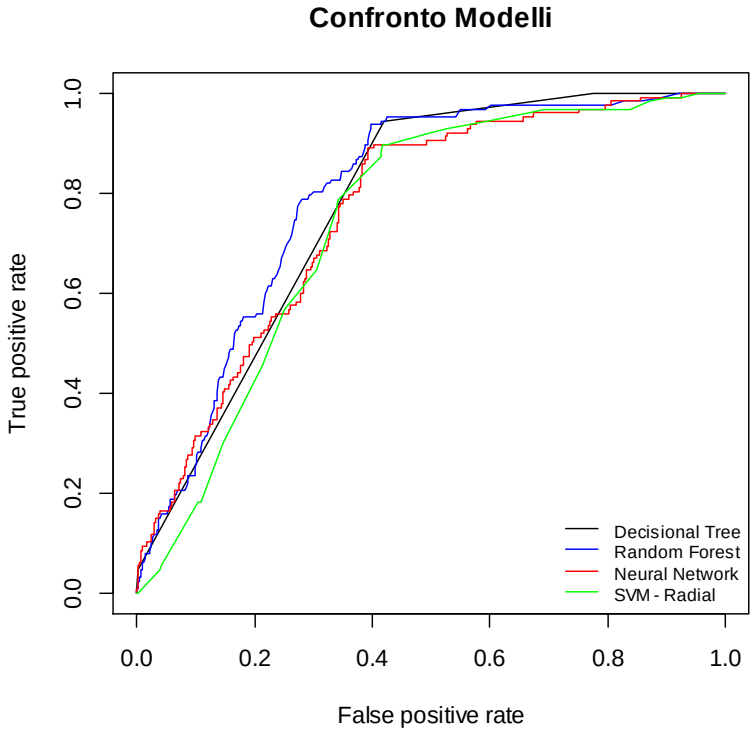
Così come nel caso della rete neurale il modello non risente dello sbilanciamento tra le classi nella fase di addestramento.

Selezionato il data base 50/50 la k - fold cross validation ha portato ai seguenti risultati:

Statistica	Totale	FP	FN
Minimo	0,0002649	0,00000	0,00533
1st QU	0,0006788	0,00000	0,01087
Mediana	0,007947	0,000134	0,01386
Media	0,018212	0,158077	0,01356
3rd QU	0,017219	0,107612	0,01521
Massimo	0,069536	0,930233	0,02266

3.4 Confronto tra i modelli

In questa sezione andremo a confrontare direttamente le curve ROC dei quattro modelli sovra presentati nonché la performance out of sample complessiva e nelle due classi specifiche. Dal confronto delle curve ROC si ottiene innanzitutto:



Che evidenzia come la random forest si presenti la miglior situazione rispetto a tutti e quattro i modelli mentre l'SVM non vanta un performance così soddisfacente. Questo confronto si può giustificare dal fatto che a differenza di tutti gli altri modelli l'SVM sembra particolarmente sensibile situazioni in cui è presente una forte asimmetrie nella numerosità degli esempi per le due classi.

Confrontando i gli errori ottenuti con la tecnica del k - fold validation otteniamo invece:

Totale	SVM	RN	AD	RF
Minimo	0,3291	0,2417	0,2298	0,0002649
1st QU	0,3382	0,2772	0,2503	0,0006788
Mediana	0,3563	0,3215	0,2778	0,007947
Media	0,3626	0,5703	0,2750	0,018212

3rd QU	0,3887	1,0000	0,2911	0,017219
Massimo	0,4053	1,0000	0,3417	0,069536

False Positive	SVM	RN	AD	RF
Minimo	0,05178	0,1196	0,09302	0,00000
1st QU	0,06418	0,3123	0,19488	0,00000
Mediana	0,07134	0,5232	0,23821	0,000134
Media	0,07902	0,5957	0,25017	0,158077
3rd QU	0,07958	1,0000	0,32844	0,107612
Massimo	0,16279	1,0000	0,39060	0,930233

False Negative	SVM	RN	AD	RF
Minimo	0,3980	0,4911	0,2000	0,00533
1st QU	0,4067	0,4999	0,2146	0,01087
Mediana	0,4152	0,7713	0,2487	0,01386
Media	0,4167	0,6770	0,2589	0,01356
3rd QU	0,4245	0,7892	0,2973	0,01521
Massimo	0,4411	0,8464	0,3567	0,02266

Possiamo innanzitutto osservare come la Rete Neurale risulti essere il peggiore modello tra i quattro proposti, infatti la performance complessiva e specifica nel riconoscimento delle singoli classi risulta essere sempre la più bassa.

Osserviamo poi come l'SVM risulti essere il miglior modello nel riconoscere i sinistri fraudolenti con un FP rate di circa l'8% contro un 16% della random forest e 25% dell'albero decisionale.

Nel riconoscimento dei no la random forest presenta una performance nettamente superiore a quella di ogni altro classificatore con un FN rate di circa il 1,3% contro un 26% dell'albero decisionale ed un 41% della SVM.

Questa elevatissima capacità di classificazione della classe di maggioranza aumenta la performance totale della random forest che presenta una percentuale di misclassificazione complessiva di circa il 2% contro un 27,5% dell'albero decisionale ed un 36,26% della SVM.

Evidentemente il confronto tra random forest e l'albero decisionale vede il primo modello come dominante data la sua miglior performance su tutti e tre i fronti.

Il confronto finale è quindi tra la SVM e la Random Forest. Innanzi tutto osserviamo come il tasso di errore nel riconoscere i sinistri fraudolenti nella SVM sia all'incirca la metà di quello della Random Forestm, ne segue che se l'effettivo costo economico della frode risultasse particolarmente elevato sarebbe ragionevole preferire l'SVM. D'altro canto un secondo aspetto che salta subito all'occhi è l'elevata asimmetria positiva che si osserva nella distribuzione dell'errore complessivo e specifico nei Si della Random Forest.

Infatti confrontando semplicemente il valor medio e mediano si nota come la SVM abbia una distribuzione abbastanza simmetrica con una lieve asimmetria a destra. La random forest invece presenta differenze relativamente molto più elevate tra media e mediana il che indica una maggior asimmetria positiva.

Quindi la distribuzione dell'errore della random forest sembra avere una coda a destra molto più pesante rispetto alla SVM. Per avere una idea del livello di variabilità tra le due distribuzioni si calcola la differenza tra il primo e terzo quartile e lo si rapporta alla media.

Variabilità	SVM	Random Forest
Totale	0,164368	0,93093
False Positive	0,35181	0,6808757
False Negative	0,042717	0,313131

È evidente come le stime della Random Forest vantino una maggior variabilità rispetto a quelle ottenute con la SVM, quindi oltre ad una maggior precisione nel riconoscimento della frode la tabella di cui sopra suggerisce una maggior stabilità delle stime fornite dalle macchine a vettori di supporto e quindi anche una maggior robustezza del modello.

Conclusione della quinta parte

Dal confronto dei quattro modelli di cui sopra non sembra emerge un classificatore che domina su tutti gli altri.

La rete neuronale e l'albero decisionale sono tranquillamente scartabili, infatti l'albero decisionale è interamente dominato dalla Random Forest mentre la rete neuronale risulta essere il peggior classificatore in tutti i tipi di errore.

La scelta tra l'SVM e la Random Forest non è però così immediata visto che da un lato si ha un modello migliore nel riconoscere le frodi ma che commette un elevato numero di errori nell'altra classe affossando la performance complessiva.

La Random Forest presenta una performance elevatissima nel riconoscere la classe dei sinistri legal (il che amplifica la precisione complessiva del modello quasi al 98%) a cui si associa una inferiore capacità del modello nell'individuare i casi fraudolenti. Va comunque detto che la Random Forest riconosce le frodi nel 84% dei casi dunque non si può dire che la sua performance nei Si sia insoddisfacente.

È quindi evidente come il criterio che deve guidare tra la scelta tra la SVM o Random Forest si basi sulla stima di quel parametro β che andava a indicare il costo della frode assicurativa per la compagnia.

Per meglio capire quest'ultimo punto si consideri il seguente ragionamento: sia N il numero complessivo di sinistri generati dal portafoglio assicurativo nel periodo di copertura e si definisca con f la percentuale fisiologica di sinistri fraudolenti che la compagnia subisce.

Da questo f ricaviamo quindi che $(1 - f)$ rappresenta il limite superiore nella capacità di riconoscimento della frode che non può essere superato da nessun classificatore. A fronte di questa formulazione possiamo definire il maggior costo che la compagnia subisce dalla frode (rispetto al livello fisiologico inevitabile) a causa dei limiti del classificatore adottato come:

$$Maggior\ Costo\ Frode = (FP - f)\beta$$

Va però considerato che anche l'errore nel classificare un sinistro lecito come fraudolento può causare una serie di costi ulteriori alla compagnia. Assumiamo che questi siano una descritti da una qualche funzione $g: FN \in [0,1] \rightarrow R^{+}$ che aumenti all'aumentare del rate nei false negative.

$$Maggior\ Costo\ Errori = g(FN)\ t.p.c. \frac{dg}{dFN} \geq 0$$

Date le assunzioni di cui sopra possiamo definire il costo complessivo derivante dagli errori di classificazione del modello come

$$(FP - f)\beta + g(FN)$$

È quindi evidente che dati due generici modelli di classificazione A e B la compagnia sceglie quello che presenta il minor costo. Introducendo anche una generica variabile $I > 0$ che va a rappresentare il costo di implementazione di un determinato modello avremo che la compagnia sceglie di adottare il modello A se e solo se:

$$(FP_A - f)\beta + g(FN_A) + I_A \leq (FP_B - f)\beta + g(FN_B) + I_B$$

isolando β ricaviamo quindi

$$\frac{E(Z^F)}{E(Z^{NF})} = \beta \leq \frac{[g(FN_B) - g(FN_A)] + (I_B - I_A)}{FP_A - FP_B}$$

e quindi in definitiva il modello A viene preferito solo quando

$$E(Z^F) \leq E(Z^{NF}) \frac{[g(FN_B) - g(FN_A)] + (I_B - I_A)}{FP_A - FP_B}$$

Supponendo che $FP_A < FP_B$ avremo che

$$E(Z^F) \geq E(Z^{NF}) \frac{[g(FN_B) - g(FN_A)] + (I_B - I_A)}{FP_B - FP_A}$$

Intuitivamente affinché sia preferibile il modello più performante nel riconoscere le frodi è necessario che il costo dei sinistri illeciti superi una certa soglia minima.

Questo ragionamento evidentemente semplifica molto il discorso (infatti non considera ad esempio la robustezza del modello e variabilità nel suo comportamento classificatorio) ma permette di cogliere la questione.

In definitiva non è possibile stabilire a priori se è preferibile la Random Forest o l'SVM perché nessuno dei due modelli domina sull'altro in relazione ad ogni tipo di errore che commette.

Nel caso in cui il costo della frode risulti essere particolarmente gravoso per la compagnia questa tenderà a scegliere la SVM che gli permette di ridurre quel maggior costo derivante dai sinistri illeciti rispetto al livello fisiologico. In alternativa potrebbe essere preferibile la Random Forest visto la sua comunque buona performance nel riconoscimento dei Si e contemporanea elevatissima performance nel riconoscimento dei No che porta ad una percentuale complessiva di osservazioni classificate erroneamente di molto inferiore a quella della SVM.

È evidente come l'intensità del maggior costo derivante dalla frode assicurativa dipenda da tutta una serie di fattori quali ad esempio il ramo assicurativo, l'area geografica ed il tipo di portafoglio detenuto.

Tutto ciò non fa che ribadire l'impossibilità di poter individuare un modello ottimale a priori senza un attento studio del caso concreto che si va a trattare.

Prima di concludere il discorso si vuole però fare un cenno al seguente ragionamento: nello studio presentato in questo elaborato tutti i modelli si sono focalizzati sulla distinzione tra la classe dei sinistri fraudolenti e leciti. L'attenzione si è quindi sempre concentrata sul label che viene associato ad ogni osservazione di training e test. È però opportuno osservare come a queste etichette sia associato un set di covariate che assumiamo in grado di descrivere il fenomeno della frode assicurativa.

Un approccio alternativo si potrebbe basare quindi sul seguente quesito: è possibile individuare un profilo nelle covariate che caratterizzi specificatamente il sinistro fraudolento e quello lecito?

L'intuizione è che se focalizziamo l'attenzione sui soli sinistri fraudolenti è possibile immaginare che esista un certo subset di feature particolarmente importanti nel descrivere questo tipo di sinistro. D'altro canto vi sarà un equivalente subset di covariate in grado di caratterizzare al meglio il profilo di un sinistro lecito.

Per rappresentare questi livelli di importanza delle covariate nel descrivere i due tipi di sinistri possiamo immaginare di associare un vettore di pesi (la cui somma deve necessariamente dare 1) a quello delle variabili esplicative.

Formalmente: dato un insieme di m osservazioni definito come

$$D_{m\times(n+1)} = \begin{bmatrix} y_1 & x_{1,1} & x_{1,n} \\ \cdots & \cdots & \cdots \\ y_m & x_{m,1} & x_{m,n} \end{bmatrix}$$

ove y è il label assegnato alla osservazione i - esima e X_{nx1} il vettore n dimensionale delle covariate associate a quella osservazione.

Possiamo quindi scomporre D in due data frame

$$D^F{}_{fm\times(n+1)} = \begin{bmatrix} 1 & x_{1,1} & x_{1,n} \\ \cdots & \cdots & \cdots \\ 1 & x_{fm,1} & x_{fm,n} \end{bmatrix}$$

$$D^{NF}{}_{(1-f)m\times(n+1)} = \begin{bmatrix} 0 & x_{1,1} & x_{1,n} \\ \cdots & \cdots & \cdots \\ 0 & x_{(1-f)m,1} & x_{(1-f)m,n} \end{bmatrix}$$

che vanno a rappresentare rispettivamente le sole osservazioni fraudolente e lecite.

Studiando separatamente i due data frame si cerca di individuare una sorta di gerarchia nel set delle feature che vada a rappresentare il grado di importanza con cui ogni variabile esplicativa va a caratterizzare il profilo di quel particolare tipo di sinistro.

Andremo quindi a rappresentare questa scala gerarchica con un vettore $n \times 1$ di pesi che formalmente indichiamo come:

$$w^F = \begin{pmatrix} w_1^F \\ \dots \\ w_n^F \end{pmatrix} ed\ w^{NF} = \begin{pmatrix} w_1^{NF} \\ \dots \\ w_n^{NF} \end{pmatrix} tpc \sum_{i=1}^{n} w_i^F = \sum_{i=1}^{n} w_i^{NF} = 1$$

Definiti questi due vettori l'idea è quella di prendere tutte le osservazioni di addestramento delle due classi e "reinterpretarle" sulla base della rispettiva gerarchia di importanza delle feature.

Avremo quindi due nuovi data base chiamati D_{super}^F e D_{super}^{NF} in cui vengono sostanzialmente accentuati i "tratti somatici" del sinistro fraudolento e lecito.

Schematicamente:

$$D^F \xrightarrow{stimo} w^F \xrightarrow{reinterpreto\ D^F} D_{super}^F$$

$$D^{NF} \xrightarrow{stimo} w^{NF} \xrightarrow{reinterpreto\ D^{NF}} D_{super}^{NF}$$

A questo punto si vanno a studiare separatamente i due nuovi data e tramite tecniche quali l'analisi dei cluster si definisce una sorta di profilo tipico del sinistro fraudolento e lecito che rappresentiamo come due punti X^F ed X^{NF} nello spazio R^n di input.

La classificazione di una nuova osservazione avverrà quindi secondo la seguente logica: preso il nuovo record di test lo si "reinterpreta" utilizzando entrambi i set di pesi w^F e w^{NF}. A questo punto si calcola la distanza intercorrente tra il nuovo record reinterpretato ed il profilo tipico di quella categoria di sinistro.

Formalmente indicando con x^{test} il generico record di test e rispettivamente con $x_{w^F}^{test}$ e $x_{w^{NF}}^{test}$ la stessa osservazione reinterpretata con i due vettori di pesi si vanno a calcolare le seguenti distanza

$$d_F = \left\| X^F - x_{w^F}^{test} \right\|$$

$$d_{NF} = \left\| X^{NF} - x_{w^{NF}}^{test} \right\|$$

x^{test} sarà quindi classificato F se e solo se $d_F < d_{NF}$ altrimenti sarà riconosciuto come lecito.

Data questa logica di classificazione la definizione dei due set di pesi potrebbe essere praticamente impostata come un problema di ottimizzazione vincolata (i singoli vettori di pesi devono comunque dare somma 1) in cui si vuole massimizzare il grado di separazione tra i due gruppi ossia la distanza tra i punti X^F e X^{NF} nello spazio di input.

Riassumendo potremmo dire che se i modelli studiati in questo elaborato (SVM, Rete Neurale, Random Forest) si focalizzano principalmente sull'etichetta associata alle diverse osservazioni questo approccio concentra maggiormente l'attenzione sullo studio delle cause che hanno portato alla generazione di quelle osservazioni nel data base di addestramento andando a definire quel profilo nelle variabili esplicative che più rappresenta la particolare classe oggetto di studio. In pratica si va a studiare maggiormente il processo che genera la frode piuttosto che focalizzarsi sulle etichette del training set che sono interpretabili come il risultato di tale processo.

Inoltre questo approccio potrebbe superare il problema dell'asimmetria nella distribuzione tra gruppi dato che si basa su due studi separati. Al più lo scarso numero di osservazioni nella classe di minoranza può portare a delle stime poco robuste del relativo vettore dei pesi anche se il modello potrebbe basarsi unicamente sul profilo della classe di maggioranza e stimare una distanza massima oltre la quale una nuova osservazione reinterpretata secondo un certo set di pesi viene assegnata alla classe alternativa.

Ad oggi, per quanto noto all'autore di questo elaborato, non sembrano esistere modelli di classificazione basati su questo approccio e potrebbe essere interessante cercare di sviluppare una tecnica di addestramento basata su tale logica per vedere come questa possa superare il problema dell'asimmetria nella distribuzione delle osservazioni e confusione tra i gruppi.

Conclusione dell'elaborato

In questo elaborato si è cercato di dare una descrizione semplice ma esaustiva di uno dei tre problemi fondamentali trattati dalla teoria dell'apprendimento statistico ossia il c.d. *pattern recognition* nella specifica fattispecie del supervised learning.

Partendo dalla metafora dello studente è stata inizialmente presentata la differenza che intercorre tra un modello che si limita a memorizzare una sequenza di input ed output ed uno che effettivamente riesce ad apprendere il processo alla base del fenomeno che genera le osservazioni di addestramento.

Si è quindi definito il concetto di apprendimento come la ricerca, a partire da un determinato set di ipotesi, di una funzione discriminante $f(x, \alpha)$ che sia in grado di ottenere un errore di classificazione atteso generico (ossia non limitato ai soli valori utilizzati nell'addestramento ma esteso a tutto lo spazio su cui si definisce il fenomeno) sufficientemente basso in base ai nostri obiettivi conoscitivi.

Abbiamo poi visto come sia possibile definire una relazione in probabilità tra questo rischio generico (non calcolabile per varie ragioni) ed una grandezza definita dalla somma tra il rischio empirico e una misura di complessità del set di funzioni che utilizziamo per la classificazione.

$$R(\alpha_l) \leq R_{emp}(\alpha_l) + \Omega(h, l, \eta)$$

La diseguaglianza è senza dubbio il concetto teorico fondamentale che sta alla base dello sviluppo di tutto l'elaborato. Infatti la relazione di cui sopra ha permesso di intuire come nelle concrete applicazioni di un qualsiasi modello di classificazione (in cui la dimensione campionaria è necessariamente limitata) la miglior performance non si ottiene minimizzando il solo rischio empirico ma la somma delle due componenti a destra della diseguaglianza.

Quindi il grado di complessità di un modello di classificazione diventa un fattore fondamentale nella valutazione della sua attesa performance out of sample.

Sulla base di questo risultato è stato quindi presentato il classificatore statistico che più si basa su tale apparato teorico ossia le macchine a vettori di supporto sviluppate dai matematici sovietici Vapnik – Červonenkis che furono non a caso padri anche della diseguaglianza di cui sopra.

A seguire sono stati presentati altri due celebri modelli di classificazioni i quali (pur non basandosi sulla teoria di Vapnik – Červonenkis) sono stati confrontati nella loro struttura teorica sulla base della diseguaglianza fondamentale.

Per poter avere una miglior comprensione delle differenze tra i classificatori si è poi proposta una sperimentazione su dati simulati in cui si andavano a confrontare le performance out of sample dei diversi modelli all'aumentare di due fattori particolarmente problematici nella teoria della pattern recognition: il grado di confusione tra le classi e l'asimmetria nella distribuzione tra i gruppi.

Questa sperimentazione ha evidenziato come in assenza di un'attenta attività preliminare di preparazione dei data base di addestramento e di parametrizzazione dei modelli le performance ottenute fossero insoddisfacenti quale che fosse il classificatore utilizzato.

Nella quarta parte è stata quindi presentata una serie di tecniche volte all'ottimizzazione del modello tramite la selezione del miglior set di parametri e feature da considerare nella fase di addestramento.

Infine nella quinta parte stata proposta una applicazione di tutta la teoria presentata ad un caso concreto di pattern recognition riguardante il settore assicurativo. Da tale applicazione è nuovamente emersa la centralità della attività di parametrizzazione e preparazione del data base per poter ottenere delle performance out of sample apprezzabili soprattutto in situazioni di forte squilibrio e confusione tra le classi.

Infine è stata evidenziato come sia sostanzialmente impossibile, dato un problema, individuare a priori il miglior classificatore da utilizzare a causa di tutta una serie di fattori esterni (quali ad esempio l'asimmetria nella gravità del commettere un particolare tipo di errore di classificazione) che caratterizzano e differenziano ogni caso oggetto di studio.

Si conclude quindi l'elaborato evidenziando come la scelta del classificatore da utilizzare non debba mai essere determinata da convinzioni a priori sulla capacità di un particolare modello nel dare migliori performance ma bisogna invece basarsi su uno studio approfondito del caso concreto che si sta analizzando per poi eseguire una serie di prove esplorative volte a valutare quale macchina di classificazione possa essere la più adatta al problema che si cerca di risolvere.

Appendice

Una proposta di modellizzazione economica del rischio frode per una compagnia assicurativa

1.Premessa

In questa appendice verrà sviluppato un ragionamento che trova le sue radici nel modello di microeconomia dell'informazione che studia la c.d. "assicurazione reciproca" ossia quella situazione in cui due soggetti esposti a rischi speculari decidono di indennizzarsi reciprocamente così da aumentare la loro utilità attesa rispetto al caso di autarchia.

Per semplicità rappresenteremo il contratto di assicurazione come una coppia di parametri (α, h) che rappresentano rispettivamente la quota di danno risarcito ed il premio pagato. Assumeremo inoltre che l'assicurato non abbia alcun potere contrattuale e che quindi potrà solamente decidere di accettare o meno il contratto proposto dalla compagnia.

Partendo da questo si andrà a studiare la relazione tra assicurato e compagnia ipotizzando uno scenario di frode in cui l'assicurato "simula" il sinistro al fine di ottenere il risarcimento senza aver subito alcuna perdita. Andremo poi a studiare il caso in cui l'assicurato è costretto a sottoscrivere il contratto anche se questo porti ad una diminuzione dell'utilità attesa rispetto alla situazione di autarchia cercando di valutare come questo possa modificare la propensione alla frode.

Infine verranno proposti alcuni ragionamenti su come sia possibile stimare la perdita attesa per frode subita nonché quantificare una misura di esposizione al fraud risk ai fini della normativa Solvency II

2.Introduzione

Supponiamo che l'individuo abbia una ricchezza di partenza aleatoria y che tra un periodo può assumere valore

1. $y = y_1$ con probabilità pari a π
2. $y = y_1 - L$ con probabilità pari a $1 - \pi$

ove $L > 0$ è la perdita in caso di sinistro .

A questo individuo viene offerto un contratto definito dalla coppia

1. $h = \pi L$ premio equo perché pari alla perdita attesa
2. $\alpha = 1$ copertura piena quindi il risarcimento in caso di sinistro sarà $R = L$

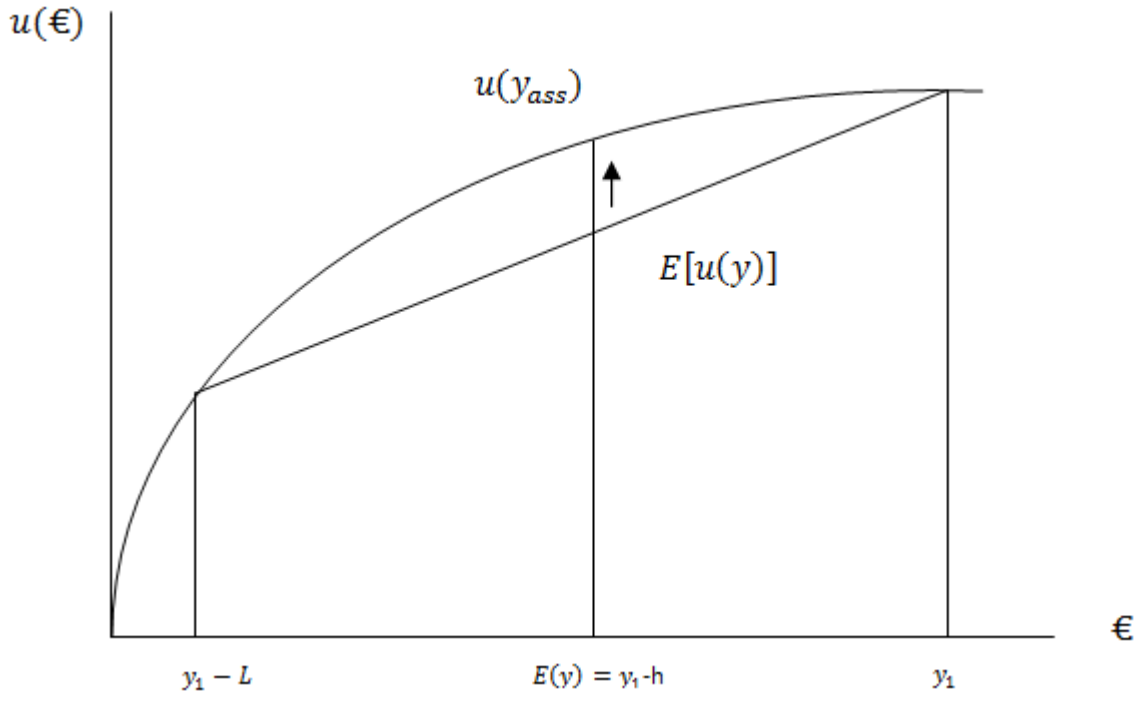
Dalla microeconomia sappiamo che l'assicurato accetterà questo contratto perché gli permette di ottenere con certezza la sua ricchezza attesa di partenza

$$y_{as} = y_1 - h = y_1 - \pi L = E(y)$$

ed assumendo che l'assicurato sia avverso al rischio avremo che la sua generica funzione di utilità u sarà di tipo concavo per la quale vale la diseguaglianza di Jensen

$$E[u(y)] < u[E(y)] = u(y_{as})$$

quindi l'individuo riesce ad aumentare la sua utilità attesa assicurandosi.



Si supponga che il danno da sinistro abbia un comportamento dicotomico del tipo $(0, L)$ cioè che non si possano verificare danni di entità compresi tra 0 ed L.

Posto quindi che l'individuo stipuli il contratto (dato che gli permette di aumentare la sua utilità attesa rispetto al caso in cui non si assicura) quand'è che gli conviene tentare una frode ai danni dell'assicuratore?

Per evitare confusione è bene evidenziare due punti:

1. La frode assicurativa è un fenomeno estremamente complesso che può avvenire in diversi momenti del rapporto assicurativo e con diversa entità. Per semplicità in questa sede con frode si intende il fatto che l'assicurato dichiari di aver subito il sinistro per ottenere un risarcimento $R = L$ (essendo la copertura piena) senza aver subito alcuna perdita così da poter aumentare la sua ricchezza di R
2. La frode è tentata solo nel caso di assenza del sinistro visto che la copertura piena e la dicotomia del danno comportano l'inesistenza di ogni possibile guadagno illecito in qualsiasi altro caso.

Sia dunque F la probabilità di riuscita della frode e sia S la sanzione che l'assicurato subirebbe se dovesse essere scoperto nel suo tentativo di frode (tipicamente $S > R$ anche se per ora non poniamo alcuna condizione). Lo schema dell'assicurato disposto a frodare diventa quindi

- $y_{as}^F = y_1 - h$ con probabilità π se non accade il sinistro
- $y_{as}^F = y_1 - h + R$ con probabilità $(1 - \pi)F$ se riesce nella frode
- $y_{as}^F = y_1 - h - S$ con probabilità $(1 - \pi)(1 - F)$ se non riesce nella frode

Per semplicità assumiamo che arrivati all'ultimo giorno di copertura assicurativa la probabilità di accadimento del sinistro sia praticamente zero e che quindi l'assicurato decida di inscenare il sinistro (ciò giustifica lo schema ex ante sopra indicato).

Quand'è che l'assicurato è disposto a tentare la frode? Quando l'utilità attesa di y_{as}^F è maggiore dell'utilità certa di y_{ass} ossia

$$u(y_{ass}) \leq E[u(y_{as}^F)] = (1 - \pi)[u(y_1 - h + R)F + u(y_1 - h - S)(1 - F)] + \pi u(y_1 - h)$$

scomponendo $u(y_{ass}) = (1 - \pi)u(y_{ass}) + \pi u(y_{ass})$ e ricordando che $y_{ass} = y_1 - h$ otteniamo

$$u(y_{ass}) \leq u(y_1 - h + R)F + u(y_1 - h - S)(1 - F)$$

$$u(y_{ass}) \leq F[u(y_1 - h + R) - u(y_1 - h - S)] + u(y_1 - h - S)$$

e quindi

$$\frac{u(y_1 - h) - u(y_1 - h - S)}{u(y_1 - h + R) - u(y_1 - h - S)} \leq F$$

Chiamando sinteticamente con

$$A = u(y_1 - h) - u(y_1 - h - S)$$

che rappresenta una sorta di effetto deterrente nel provare la frode (che dipende da S) e che dovrebbe quindi invogliare l'assicurato a non tentare la frode tenendosi la ricchezza certa.

E con

$$B = u(y_1 - h + R) - u(y_1 - h - S)$$

il range di variazione del livello di utilità nel caso in cui è tentata la frode, questo può essere facilmente scomposto come

$$B = [u(y_1 - h) - u(y_1 - h - S)] + [u(y_1 - h + R) - u(y_1 - h)] = A + C$$

ove

$$C = u(y_1 - h + R) - u(y_1 - h)$$

rappresenta l'aumento di ricchezza che invoglia l'assicurato nel tentare la frode.

Il risultato precedente è quindi riscrivibile come

$$\frac{A}{A + C} \leq F$$

alternativamente

$$A(1 - F) \leq CF$$

$$E[maggior\ utilità\ per\ frode] \leq E[minor\ utilità\ per\ frode]$$

Come ci aspettavamo l'assicurato ha convenienza a tentare la frode quando il maggior ritorno atteso nel caso di successo è superiore al minor ritorno atteso in caso di fallimento.

Notiamo che se:

- $F = 0$ allora $A \leq 0$ che è impossibile essendo $S > 0$ quindi l'assicurato non ha convenienza a tentare la frode quale che sia il suo livello di avversione al rischio
- $F = 1$ allora $0 \leq C$ il che è sempre vero essendo $R > 0$ quindi l'assicurato ha convenienza a tentare la frode quale che sia il suo livello di avversione al rischio

Essendo L ed S strettamente maggiori di 0 (quale dei due sia più alto per ora non ci interessa) si ha per definizione che $A \leq B$ essendo A contenuto in B e quindi vale il seguente risultato

se F tende a 1 *è razionale il comportamento di un assicurato che tenta la frode quale che sia il suo grado di avversione al rischio e livello della sanzione*

Il risultato è ovvio ma mostra un particolare interessante ossia che se $F \to 1$ l'assicurato preferisce passare da una situazione di ricchezza certa ad una di ricchezza quasi (perché F è quasi 1) aleatoria quale che sia il suo grado di avversione.

Per una maggior precisione possiamo riformulare la frase come:

Dati n assicurati avversi al rischio il numero m assicurati che provano la frode è tale per cui $lim_{F \to 1}\, m = n$

3.Quanto deve essere grande F

Nell'introduzione abbiamo visto che dati n assicurati vale che

$$\lim_{F\to 1} m = n$$

è opportuno chiedersi quale sia però un livello di F da considerare "pericoloso" per l'assicuratore.

Con pericoloso intendiamo quel F^* tale per cui

$$\frac{m}{n} \geq \delta$$

ove δ è un valore stabilito dall'assicuratore.

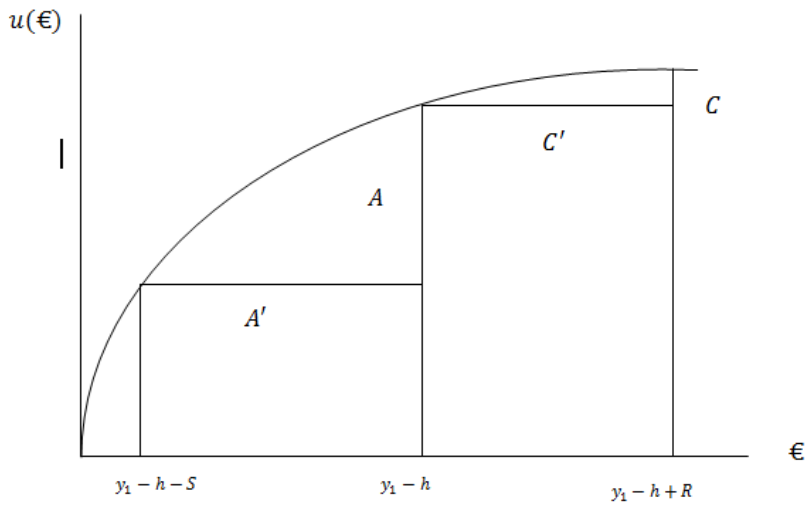
In sostanza vogliamo capire quel livello di F per cui una quota di assicurati superiore a δ trovi convenienza a provare al frode (si ricordi che ogni assicurato ha una diversa funzione di utilità ma per semplicità assumiamo che siano tutti avversi al rischio e quindi ordinabili in base al loro grado di avversione).

Ne segue che il ragionamento sarà fatto per una singola funzione di utilità (ossia uno specifico assicurato) posto che l'F^* che lo renderà indifferente tra il provare o meno la frode comporterà un interesse a frodare per tutti gli individui che sono meno avversi al rischio dell'assicurato di riferimento ed una rinuncia da parte di tutti gli assicurati più avversi al rischio.

Fatta questa premessa iniziamo il ragionamento partendo dal caso più semplice ossia quello di simmetria nell'intorno di $y_1 - h$ che è definito quando $S = R$.

1. $S = R$

Indicando con A e C le variazioni di utilità e con A' e C' le variazioni nei livelli monetari notiamo che a causa della concavità di u si ha che:



$$A' = C' perché\ S = R\ ma\ A >> C\ a\ causa\ della\ concavità$$

quindi se $F = \frac{1}{2}$ otteniamo

$$A(1-F) = A\frac{1}{2} \leq C\frac{1}{2} = CF$$

$$A \leq C$$

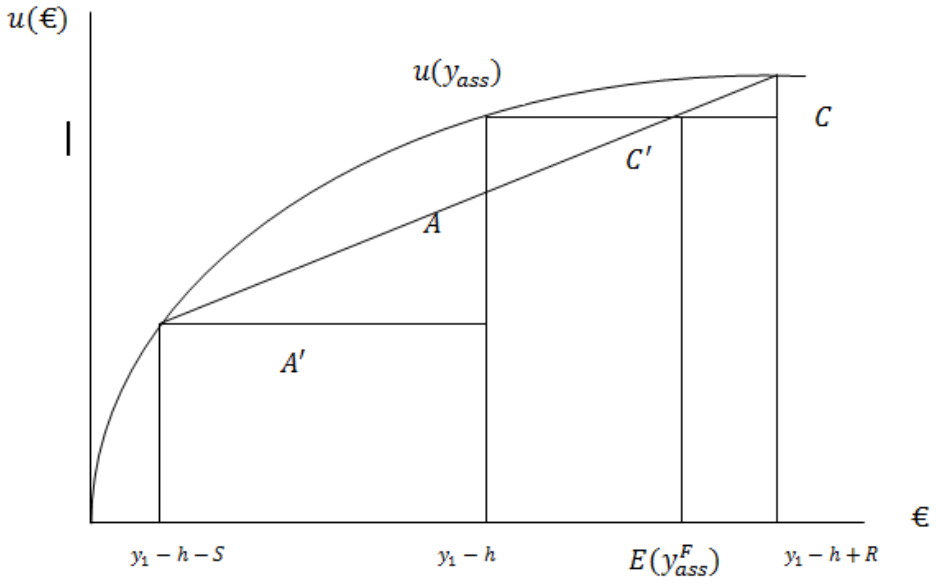
che contraddice quanto appena detto quindi nel caso di assicurazione con copertura piena ed $S = R$ basta un $F = \frac{1}{2}$ perché l'individuo avverso non abbia interesse nel provare la frode.

Perché la frode diventi conveniente all'assicurato è necessario che $F > 1/2$ allora all'assicuratore non serve fare grandi investimenti nella prevenzione frode.

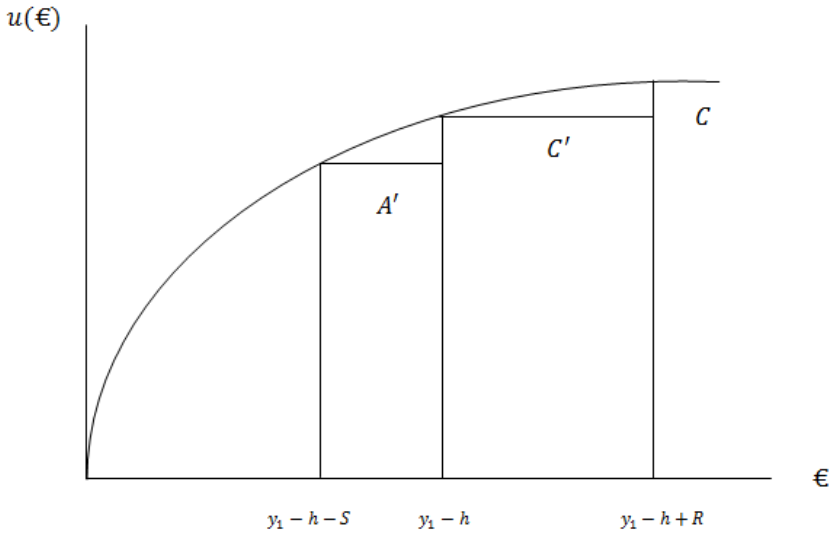
Notare che nella realtà L non è dicotomico ed $\alpha < 1$ quindi i risultati potrebbero variare soprattutto quando $h \gg \pi L$ che è il premio equo.

Ricordando poi che $h = \pi L$ si nota come l'aumento di π (a parità di L) faccia traslare verso destra i tre punti di interesse di egual misura (perché la dispersione non dipende da R ed S) ossia in un'area ove l'effetto di concavità è maggiore quindi basta un minor F perché sia conveniente la frode allora se $h > \pi L$ questo effetto è ancora più elevato.

In sostanza F deve essere sufficientemente alto affinché l'utilità attesa nel caso in cui si tenti la frode sia almeno pari all'utilità certa che si otterrebbe dal contratto assicurativo. Definito questo expected possiamo definire l'area alla sua destra come quella che definisce la zona di convenienza alla frode.

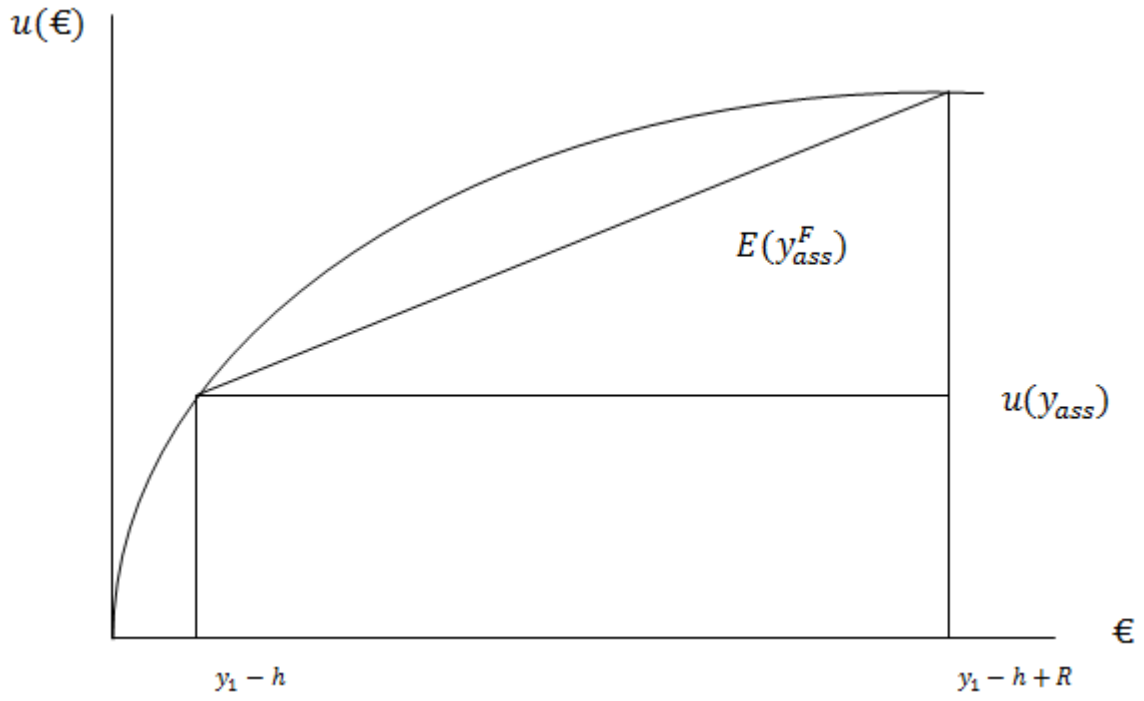


2. $S < R$



In tal caso C non varia ma A diminuisce ne segue che al diminuire di S diminuisce anche la F di equilibrio che rende l'individuo indifferente tra il provare o meno la frode.

Basti osservare il caso (volutamente estremizzato) in cui $S = 0$, in questa situazione l'utilità attesa nel provare la frode è sempre maggiore di quella certa ottenibile dall'assicurazione quale che sia il valore di F. Ne segue che anche con una F tendente a 0 l'assicurato avrebbe convenienza nel provare la frode.



3. $S > R$

Il commento è speculare a quello appena fatto nel caso in cui S < R.

4.Trade off: spesa per prevenzione e perdita da frode

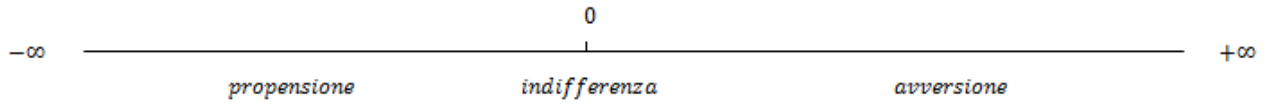
Come indicato nell'introduzione del capitolo precedente il risultato

$$\lim_{F \to 1} m = n$$

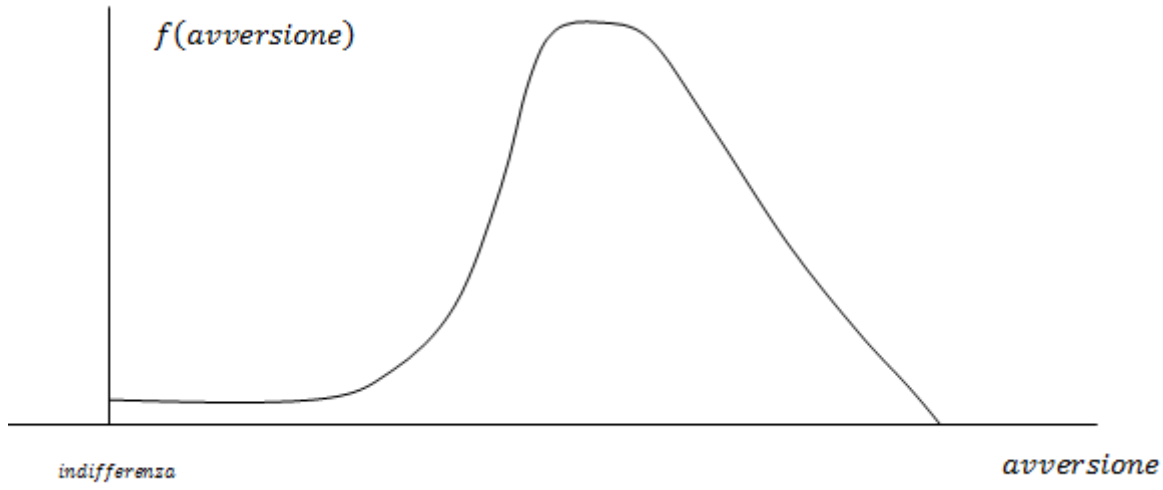
ci da la chiave di lettura nello studiare quale F^* renda indifferente l'assicurato tra il provare o meno la frode.

Notiamo infatti che nel caso in cui tutti gli assicurati abbiano la stessa funzione di utilità potremmo individuare un livello di F^* che sia valido per l'intero portafoglio e quindi basterebbe ottenere un effettivo $F < F^*$ (investendo in misure di prevenzione e detenzione frode) perché l'assicuratore possa interamente eliminare il rischio frode da quel portafoglio.

Assumere che tutte le $u_i(\,)$ siano uguali tra loro è però un'ipotesi troppo forte ed è più ragionevole pensare che vi siano n diverse funzioni di utilità, una per ogni assicurato. Assumendo che tutti gli individui siano avversi al rischio si hanno n funzioni di utilità diverse tra loro ma tutte di tipo concavo. Possiamo quindi immaginare di ordinare gli assicurati in base al loro grado di avversione al rischio (ossia la forza della concavità della rispettiva funzione di utilità) su una retta.



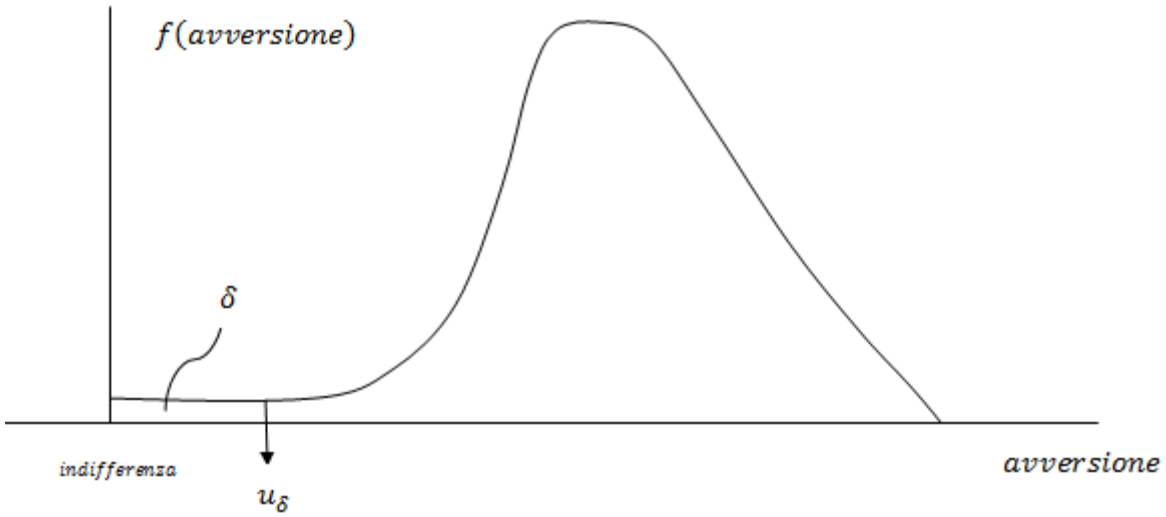
Dato che le funzioni di utilità sono tutte diverse tra loro possiamo immaginare una distribuzione dell'avversione al rischio definita tra gli assicurati del portafoglio.



Diventa quindi evidente il ragionamento prima indicato ossia

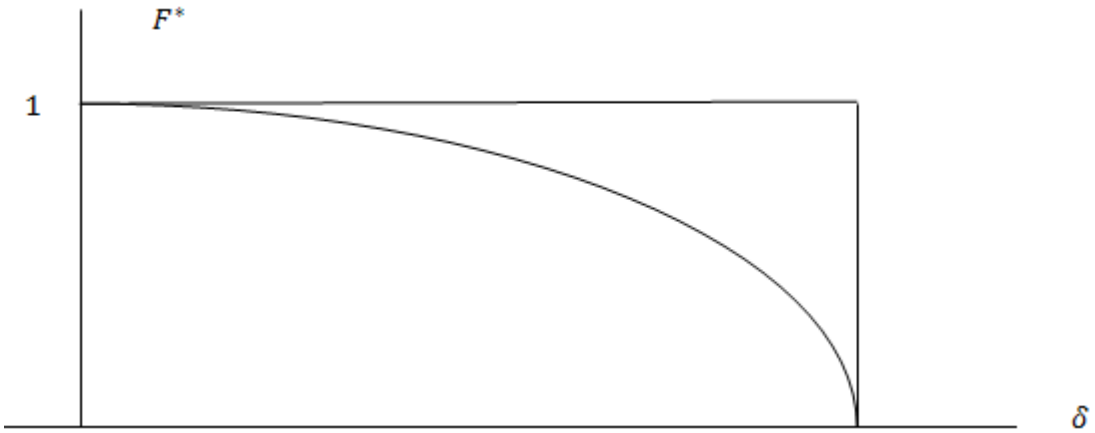
$$F^*: \frac{m}{n} \leq \delta$$

ove δ equivale ad un'area della distribuzione che individua nel percentile la funzione di utilità di riferimento (indicata con u_δ) da studiare per calcolare la F^* obiettivo.

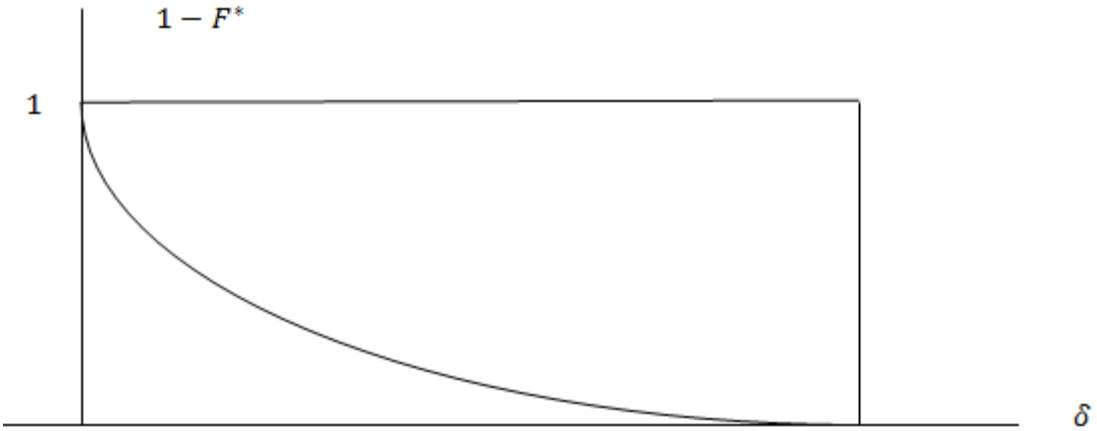


Evidentemente lo studio dello shape della distribuzione dell'avversione al rischio è di fondamentale importanza per capire come δ indichi un percentile vicino o meno all'origine.

E' bene osservare che valori più a destra della distribuzione fanno riferimento ad individui estremamente avversi al rischio per i quali serve una $F \to 1$ (cioè deve essere quasi certa la riuscita della frode) perché vi sia convenienza nel provare l'illecito. D'altro canto vicino allo 0 ho individui molto poco avversi ove anche una F molto contenuta li renderebbe indifferenti tra il provare o meno la frode.

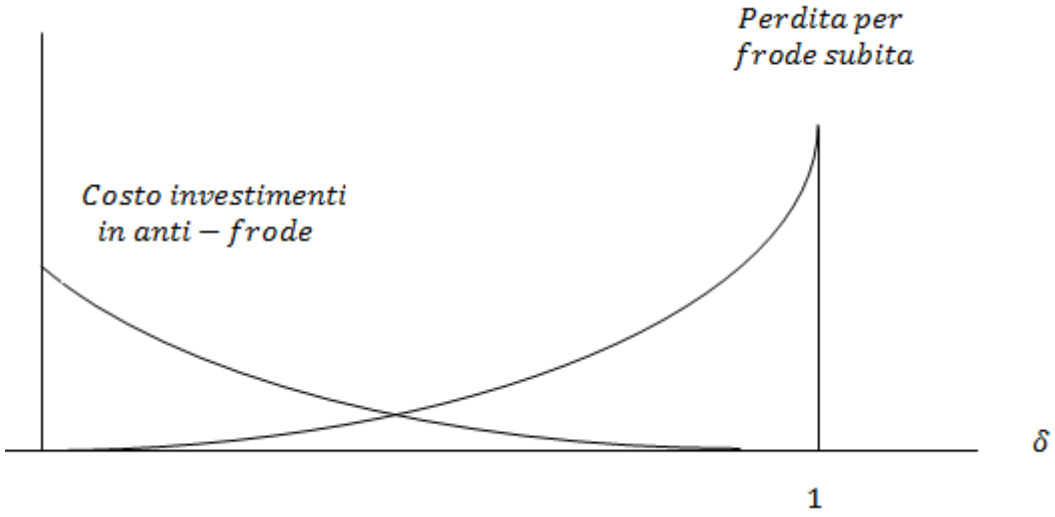


La F^* è il punto di equilibrio per la funzione di utilità definito da un certo grado di avversione. Con $1 - F^*$ si indica invece la probabilità che l'assicuratore scopra la frode e quindi possiamo vederlo come una sorta target della compagnia nella sua capacità di individuazione dei sinistri fraudolenti.



Lo shape della distribuzione a parità di δ è fondamentale per definire la funzione di utilità di riferimento e quindi l'F^* (cioè gli investimenti in misure di contrasto alla frode). A parità di shape il livello di δ ritenuto accettabile ha una conseguenza sul $1 - F^*$ tale per cui all'aumentare di δ diminuisce $1 - F^*$ perché l'assicuratore accetta una maggior percentuale di individui che provano la frode (individui a sinistra del percentile).

Quando L'assicuratore decide di investire in misure anti-frode dovrà comunque tener conto di un trade off di questo tipo



Quindi il quanto il valore ottimale di δ^* da cui ricavare lo stesso F^* dipende anche da quanto l'azienda debba investire nei processi antifrode. Si osservi che la curva *Perdita per frode subita* sta ad indicare i costi che l'azienda si attende di sostenere nonostante una performance nell'individuare la frode di $1 - F^*$ (si veda il capitolo 8 a riguardo).

E' di tutta evidenza che quanto fatto fin'ora serve ad individuare quel F^* tale per cui solo una percentuale δ di assicurati proverà la frode. Questa F^* definisce un $1 - F^*$ che a sua volta identifica una sorta di livello di evoluzione delle tecniche antifrode che l'assicuratore deve detenere.

D'altro canto l'assicuratore ha già implementato nell'organizzazione aziendale dei sistemi antifrode quindi vi è già un livello $1 - \dot{F}$ ossia una capacità dell'azienda ad individuare la frode allora l'investimento della compagnia non è definito da $1 - F^*$ quanto più dal risultato di una gap analisi del tipo

$$1 - \dot{F} \xrightarrow{investimenti} 1 - F^* \; tale \; per \; cui \; \frac{m}{n} = \delta$$

La curva del costo per investimento in antifrode coincide con la precedente quando $\dot{F} = 1$ l'aver minori costi dovrebbe garantire un punto di equilibrio più basso (minor costo) e più a sinistra quindi è possibile avere un δ più contenuto (a parità di F^* significa che le aziende con un sistema più evoluto ha minori costi.

Come il lettore avrà intuito tutto il ragionamento fin'ora fatto ha un evidente punto critico ossia

$$da \; \delta \; ottengo \; F^* \; e \; quindi \; \dot{F} - F^*$$

quindi basta superare la soglia di F^* perché si possa risolvere il problema della frode?

Ovviamente no ed infatti tutta l'analisi fin'ora fatta può essere pensata come un'analisi ad un periodo e dato che l'aggiornamento delle tecniche antifrode è un investimento non indifferente ha senso fare dei ragionamenti prospettici di medio lungo.

In questa analisi però sorgono due fattori di dinamismo che rendono particolarmente complesso il problema:

1.Evoluzione del portafoglio

Se nel tempo la dimensione e composizione del portafoglio varia allora potrebbe anche variare la distribuzione dell'avversione al rischio e quindi i livelli di ottimo definiti precedentemente.

Una gestione prospettica del rischio frode deve quindi tener conto di come si evolve nel tempo la distribuzione del grado di avversione al rischio del portafoglio. A tale proposito due sono gli scenari che si possono verificare:

a. esiste una distribuzione asintotica:

se fosse possibile individuarla allora con riferimento a questa distribuzione si definisce il percentile u_{δ}^{∞} da cui otteniamo F_{∞}^{*} quindi l'investimento che consente di passare da $1-\hat{F}$ a $1-F_{\infty}^{*}$ permette alla compagnia di gestire definitivamente il rischio frode.

b. non esiste una distribuzione asintotica

è necessario studiare al distribuzione dell'avversione su un Δt medio lungo e quindi definire un livello $1-F_{\infty}^{\Delta t}$ da cui ricaviamo l'investimento in antifrode che la compagnia deve porre in essere, è però necessario monitorare nel tempo il portafoglio per capire se vi sono delle variazioni rispetto a quando si è fatta originariamente l'analisi

2.Evoluzione delle tecniche di frode

Anche se la distribuzione dell'avversione al rischio non cambia nel tempo va considerato il fatto che la frode è notoriamente un fenomeno estremamente dinamico ove le tecniche poste in essere diventano sempre più raffinate proprio a causa del miglioramento dei sistemi di antifrode degli assicuratori.

Quindi la F effettiva nella sostanza aumenta rispetto alla F^{*} con cui la compagnia effettua la gap analisi facendo di conseguenza aumentare la distanza rispetto al nuovo F_{new}^{*} che garantirebbe $\delta = m/n$.

Si crea un circolo del tipo:

$$F \rightarrow analisi\ per\ ottenere\ \delta \rightarrow nuovo\ F \rightarrow variazione\ del\ \delta \rightarrow analisi\ per\ ottenere\ \delta$$

Dato che la situazione più verosimile è quella in cui non esiste una distribuzione asintotica dell'avversione (sempre che sia possibile averne una stima significativa) e data la dinamica evolutiva della F appena presentata una gestione organica e strutturale del rischio frode risulta essere un compito estremamente complesso che comporta per l'azienda costi di due tipi:

1. strutturali di lungo: con riferimento alla distribuzione di lungo dell'avversione (che δ voglio ottenere?)
2. di monitoraggio continui: come si evolve la distribuzione dell'avversione e le tecniche di frode

Tipicamente il fenomeno della frode (proprio per le difficoltà di cui sopra) non è stato gestito seriamente dalle compagnie visto sostanzialmente la possibilità di rigirare questo costo sugli assicurati con un aumento dei premi.

E' evidente che un aumento dei premi non fa altro che aumentare al propensione dell'assicurato a tentare la frode scatenando quindi un ulteriore circolo vizioso. Questo è particolarmente evidente in casi ove il premio raggiunge livelli particolarmente alti e l'assicurato è obbligato a contrarre l'assicurazione per motivi legali (si veda il prossimo capitolo).

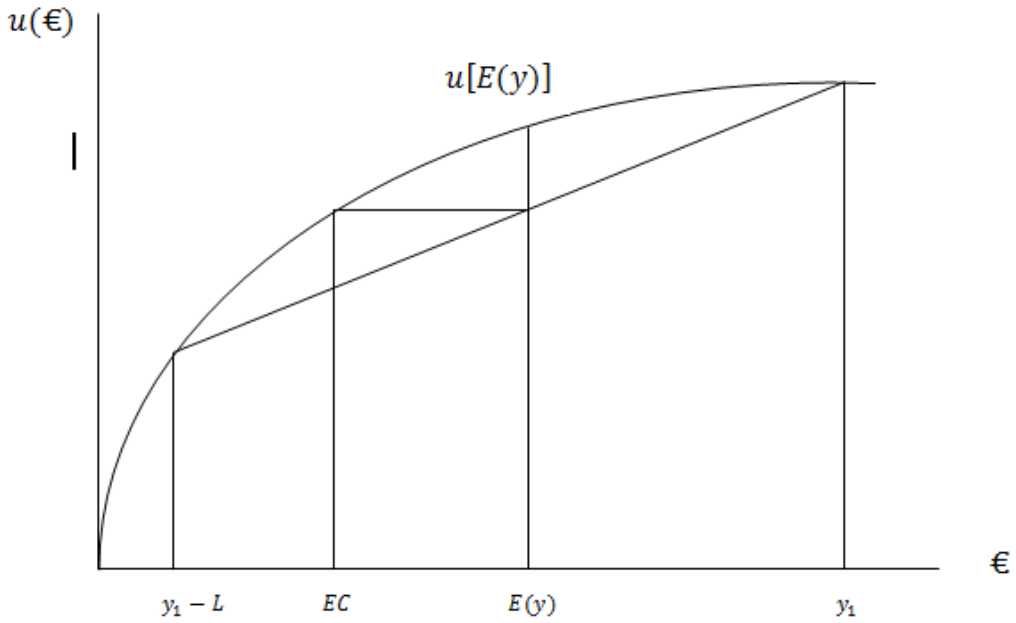
Il contesto di mercato sempre più competitivo, i livelli insostenibili dei premi su certi rami che fanno paventare un eventuale fallimento di mercato nonché l'imminente normativa di Solvency II che renderà sempre più pressante il bisogno di sapere a quali rischio l'azienda è effettivamente esposta rendono il "fraud risk" un tema estremamente delicato e attuale sia nella letteratura statistico attuariale che in quella più legata al risk management.

5 Forzatura di una scelta razionale

Fin'ora ci siamo concentrati sul caso in cui l'assicuratore offre un contratto all'assicurato (che può solo accettare o meno senza contrattare sul prezzo) ove $\alpha = 1$ e $\pi L = h$. Siamo certi che l'assicurato accetterebbe questo contratto perché sarebbe quello di equilibrio nel caso di mercato perfettamente concorrenziale (dove l'assicurato ha tutto il potere contrattuale).

Il contratto così definito permette di ottenere con certezza la ricchezza attesa nel caso di assenza della copertura assicurativa ed essendo l'individuo assunto avverso al rischio questi preferirà questa situazione rispetto all'avere una ricchezza aleatoria.

E' però evidente che esiste un intero range di contratti con $h > \pi L$ che sono comunque preferiti dall'assicurato rispetto alla situazione senza copertura perché gli garantiscono una ricchezza certa superiore alla ricchezza attesa di partenza.



Definiamo quindi con EC l'equivalente certo ossia quella somma che data con certezza rende l'assicurato indifferente tra ricevere EC o tenere la sua ricchezza aleatoria di partenza y. Quindi tutti i contratti che presentano un $\alpha = 1$ ed un premio tale da comportare una ricchezza certa y_{ass} compresa tra:

1. EC e $E(y)$ convengono all'assicurato rispetto al non assicurarsi
2. $y_1 - L$ ed EC non convengono all'assicurato che preferisce non assicurarsi

Notare che sotto $y_1 - L$ non ci interessa perché comunque il premio h non può essere maggiore o uguale al danno L provocato dal sinistro. In sostanza l'h è talmente alto da consumare tutta l'utilità che un individuo avverso al rischio ottiene riducendo la propria alea.

Sia h^* il premio tale per cui

$$u(y_1 - h^*) = E[u(y)]$$

e definiamo con h^c un generico premio compreso tra h^* ed L.

Supponiamo che l'assicuratore offra un contratto con $\{\alpha = 1; h^c\}$ e che l'assicurato sia costretto a stipulare l'accordo anche se questo gli provoca una disutilità, come cambia la sua propensione alla frode?

Va notato che all'individuo conviene essere assicurato nel caso di effettivo accadimento del sinistro perché per quanto h^c sia elevato comunque ragionevolmente varrà che

$$h^c \ll L$$

ma d'altro canto nel caso in cui il sinistro non accade l'assicurato **percepisce** una disutilità data dal fatto di essere assicurato e quindi ha una utilità inferiore a quella nel caso senza assicurazione.

Possiamo rappresentare questa disutilità percepita come un fattore D e quindi definire la ricchezza dell'individuo post assicurazione come una variabili bernoulliana y_{ass}^C che assume valori

1. $y_1 - h^c$ con probabilità π
2. $y_1 - h^c - D$ con probabilità $1 - \pi$

Definiamo inoltre la variabile y_{ass}^{FC} che va a rappresentare la ricchezza aleatoria dell'assicurato nel caso in cui questi tenti la frode, tal variabile assume valori

1. $y_1 - h^c$ con probabilità π
2. $y_1 - h^c + R$ con probabilità $(1 - \pi)F$
3. $y_1 - h^c - S$ con probabilità $(1 - \pi)(1 - F)$

Analogamente a quanto visto precedentemente l'assicurato avrà convenienza nel provare la frode quando l'utilità attesa di y_{ass}^{FC} supera quella di y_{ass}^C.

E' però opportuno notare che rispetto al caso in cui $h = \pi L$ si ha una importante modifica ossia che la ricchezza attesa y_{ass}^{FC} è confrontata con un'altra grandezza aleatoria y_{ass}^C mentre nel caso di contratto a premio equo la y_{ass} era in realtà una grandezza certa.

Essendo $\alpha = 1$ la ricchezza monetaria dell'assicurato è certa e pari a $y_1 - h^c$, quello che varia è la ricchezza **percepita** dal soggetto che sarà "contento" di aver stipulato la polizza solo nel caso in cui avviene il sinistro. Quest'ultimo punto evidenzia come il combattere la

frode passi anche attraverso una educazione finanziaria sul ruolo dell'assicurazione che potrebbe far diminuire l'entità del D sovra indicato.

Tornando ai nostri ragionamenti analogamente a quanto visto precedentemente la situazione in cui l'assicurato è indifferente tra il provare o meno la frode deve garantire la seguente uguaglianza

$$u(y_1 - h^c)\pi + u(y_1 - h^c - D)(1 - \pi) = u(y_1 - h^c)\pi + (1 - \pi)[u(y_1 - h^c + R)F + u(y_1 - h^c - S)(1 - F)]$$

e con passaggi analoghi a quelli già fatti otteniamo

$$\frac{u(y_1 - h^c - D) - u(y_1 - h^c - S)}{u(y_1 - h^c + R) - u(y_1 - h^c - S)} = F$$

che per semplicità possiamo riscrivere come

$$\frac{A}{A + C} = F$$

$$A(1 - F) = CF$$

con

$$A = u(y_1 - h^c - D) - u(y_1 - h^c - S)$$

$$B = u(y_1 - h^c + R) - u(y_1 - h^c - D)$$

$$D \in [0, S]$$

Notiamo che all'aumentare di D (ossia della disutilità percepita dall'assicurato per il fatto di essere costretto ad assicurarsi a condizioni da lui ritenute svantaggiose) diminuisce il numeratore del rapporto di cui sopra e quindi diminuisce anche la F^* che rende l'assicurato indifferente tra il provare o meno la frode.

Si osservi che la funzione di utilità $u(\ )$ di riferimento non è stata modificata e che questo effetto è interamente dovuto a D ossia al premio h^c. Come ci aspettavamo la presenza di un contratto di assicurazione con prezzi troppo elevati ed "obbligo a contrarre" da parte dell'assicurato comporta un aumento della propensione alla frode.

6 Il paradosso del bravo conducente

Nel capitolo precedente abbiamo visto come la propensione a provare la frode da parte dell'assicurato (a parità di ogni altra condizione) aumenti all'aumentare della disutilità percepita D che deriva dall'essere obbligati a stipulare un contratto che comporta una minor utilità rispetto alla situazione originale.

E' però opportuno dare una definizione più precisa di questa grandezza D per capire quali fattori lo influenzano. Si ricordi che il premio h^* garantiva l'eguaglianza

$$h^*: u(y_1 - h^*) = E[u(y_1)]$$

quindi $y_1 - h^*$ è l'equivalente certo di y ossia la ricchezza che data con certezza rende l'individuo indifferente tra l'assicurarsi o meno.

Anche nel contratto in cui premio è pari a h^C la ricchezza dell'assicurato è certa e pari a $y_1 - h^C$ essendo $\alpha = 1$ ma per costruzione dovrà valere che

$$u(y_1 - h^*) < u(y_1 - h^*)$$

$$y_1 - h^* < y_1 - h^*$$

$$h^C > h^*$$

possiamo quindi definire D come

$$D = h^C - h^*$$

Osserviamo come D diminuisca all'aumentare di h^* e che quest'ultimo possa essere definito agevolmente isolandolo dalla prima equazione

$$h^* = y_1 - u^{-1}[EU]$$

con $EU = E[u(y)]$

Assumendo infine che y_1 ed L rimangano fissi (come abbiamo sempre fatto) osserviamo in ultima analisi che h^* aumenta quando EU diminuisce ossia quando π aumenta. Non solo, essendo $u(\ )$ assunta come funzione concava la sua inversa sarà una funzione convessa e quindi l'aumento di h^* sarà più che proporzionale alla diminuzione di EU.

Si è quindi ottenuto che

$$D\ aumenta\ quando\ \pi\ \boldsymbol{diminuisce}$$

e ricordando che

$$\frac{u(y_1 - h^c - D) - u(y_1 - h^c - S)}{u(y_1 - h^c + R) - u(y_1 - h^c - S)} = F$$

abbiamo sostanzialmente ottenuto che

$$F^* \; diminuisce \; quando \; \pi \; \boldsymbol{diminuisce}$$

Tradotto in termini più discorsivi significa che un assicurato "buono" ossia con una minor probabilità di accadimento del sinistro ha una maggior propensione a frodare rispetto ad un individuo "cattivo" ossia con una maggior probabilità di subire il sinistro.

Il buon conducente è quindi più rischioso dal punto di vista del fraud risk rispetto ad un individuo meno attento alla guida.

Come si spiega tale situazione? Abbiamo visto che all'assicurato conviene assicurarsi solo nel caso in cui si verifica il sinistro essendo $h^C \ll L$ quindi un individuo con un π molto alto tenderà comunque a favorire l'essere assicurato rispetto al non esserlo perché sa di avere un'elevata probabilità di subire effettivamente L.

Un individuo "buono" invece sa di avere una π molto contenuta e quindi percepisce una maggior disutilità nell'essere costretto ad assicurarsi a quelle condizioni rispetto all'individuo "cattivo". L'individuo "buono" si sente quindi più vessato da questa coercizione e tenterà più facilmente di recuperare la sua perdita a scapito dell'assicuratore.

Questo risultato non è nulla di straordinario nella teoria economica dell'informazione (non è nient'altro che una rivisitazione in ambito frode di un concetto ben noto) ma permette di legare maggiormente il rischio frode con un ambito più tipicamente attuariale che è quello della personalizzazione del premio assicurativo.

Infatti se l'azienda fosse in grado di distinguere un assicurato in base al suo livello di rischio allora verosimilmente cercherà di modulare il premio h^C in modo tale che

1. Il soggetto buono paghi un premio $h < h^C$ così da diminuirne D
2. Il soggetto cattivo paghi un premio $h > h^C$ sfruttando il fatto che questi comunque preferisca essere assicurato (fino ad una certa soglia) avendo un elevato π

Va inoltre notato che l'assicurato "buono" sarebbe anche disposto ad accettare un $\alpha < 1$ ossia tenersi parte del rischio (e quindi diminuire il premio) a differenza dell'individuo "cattivo" per il quale il **costo** dell'accollarsi parte del rischio è maggiore avendo una più elevata probabilità di subire il sinistro (notare che le funzioni di utilità dei due soggetti sono comunque assunte uguali).

Questa personalizzazione del premio in ottica di fraud risk potrebbe permettere alla compagnia una riduzione delle perdite da comportamenti fraudolenti che potrebbe portare ad una ulteriore riduzione del premio e quindi innescare un circolo virtuoso del tipo

$$Personalizzazione \rightarrow Diminuzione\ Costo\ Frode \rightarrow Diminuzione\ Premio \rightarrow Diminuzione\ Costo\ frode$$

fino a convergere a valori di premio più sostenibili.

E' comunque opportuno evidenziare che il combattere la frode non può passare unicamente per la personalizzazione del premio perché la definizione di classi troppo specifiche può portare ad un eccessiva diminuzione della numerosità campionaria disponibile in ogni classe erodendo quindi la significatività dei modelli stimati.

Questo paradosso ben rappresenta un fenomeno già evidenziato nei diversi case study pubblicati sull'argomento ossia il fatto che la maggior parte delle frodi assicurative non sono perpetuate da organizzazioni criminali "professionali" ma da persone comuni che trovandosi nella condizione di poter ottenere un ricavo rispetto al danno del sinistro tentano la froda.

L'occasionalità di tale fenomeno e l'assenza di un *modus operandi* rende ancora più complessa la trattazione del problema.

7.Oltre la funzione di utilità

Dei ragionamenti fin'ora fatti il problema fondamentale è la loro scarsa applicabilità in un caso concreto data principalmente dalla difficoltà nello stimare la funzione di utilità di un dato individuo. Infatti tutto questo impianto teorico si basa sul fatto di conoscere la funzione di utilità di ogni assicurato in portafoglio, di riordinarle sulla base del grado di convessità e quindi di individuare quella funzione di riferimento su cui calcolare F.

A fronte della difficoltà nello stimare una qualsiasi funzione di utilità ci si chiede se è possibile mantenere il ragionamento fin'ora fatto bypassando la stima di $u(\ )$.

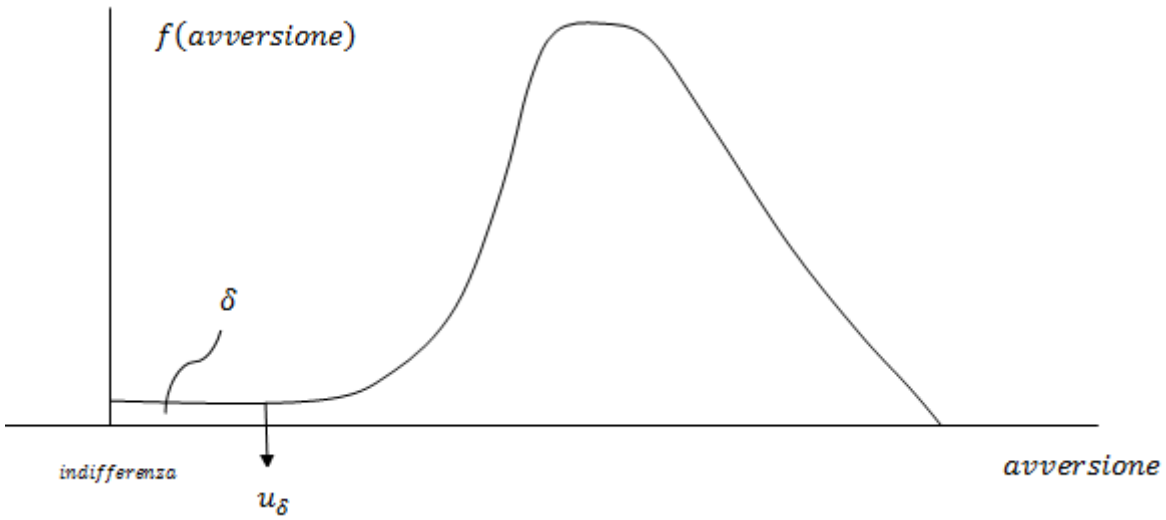
Riprendendo l'ultimo risultato

$$\frac{u(y_1 - h^c - D) - u(y_1 - h^c - S)}{u(y_1 - h^c + R) - u(y_1 - h^c - S)} = F$$

ove $u(\ )$ è la funzione di utilità percentilica di riferimento.

Possiamo osservare come la F di equilibrio sia interamente definita da quelle che sono le variazioni di quota della funzione di utilità $u(\ )$ e queste, fissati i valori monetari di interesse, dipendono dallo shape della funzione $u(\ )$ ossia dal grado di convessità/avversione.

Ne segue che la definizione della F di equilibrio dell'equazione di cui sopra non fa riferimento ad una $u(\ )$ particolare quanto più ad un livello di avversione al rischio percentilico data la distribuzione dell'avversione degli individui in portafoglio.



La distribuzione di cui sopra fa riferimento al grado di avversione degli assicurati in portafoglio e non direttamente alla loro funzione di utilità.

Se quindi fosse possibile definire una funzione $g(\ )$ tale per cui

$$u(x) - u(y) = g(avversione, x - y) \ \forall x, y \ e \ u(\ )$$

allora saremmo in grado di applicare il modello qui proposto perché non avremmo bisogno di stimare la funzione di utilità di ogni individuo.

L'effettiva possibilità di calcolare F^* è comunque condizionata alla possibilità di stimare la distribuzione dell'avversione al rischio del portafoglio che potrà essere in qualche modo definita come uno score funzione di un set di variabili osservate sull'assicurato (età, genere, reddito ecc) ad esempio con l'utilizzo di un modello GLM binomiale.

L'avversione al rischio di un individuo i-esimo sarà quindi definita come una generica equazione

$$avversione_i = g\left(\sum_{j=1}^{m} \beta_j X_{ji}\right) per \ i = 1, \dots, n$$

ove i coefficienti β_j mi dicono come e con che misura la generica variabile X_j impatti nel determinare l'avversione al rischio di un individuo.

Un'ultima nota sulla stima dell'avversione al rischio è il seguente ragionamento: il livello di rischiosità di un individuo è in qualche modo legato almeno in parte con la sua avversione al rischio.

Non è assurdo pensare che più un individuo è avverso al rischio e più il suo comportamento tenderà ad essere preventivo nei confronti dell'evento avverso e quindi parte della sua rischiosità sarà più contenuta. Questo legame è solo parziale perché non si può ignorare il fatto che il processo che genera il sinistro ha anche una componente esogena all'assicurato il quale può mettere in atto tutte le misure preventive e comunque subire il sinistro.

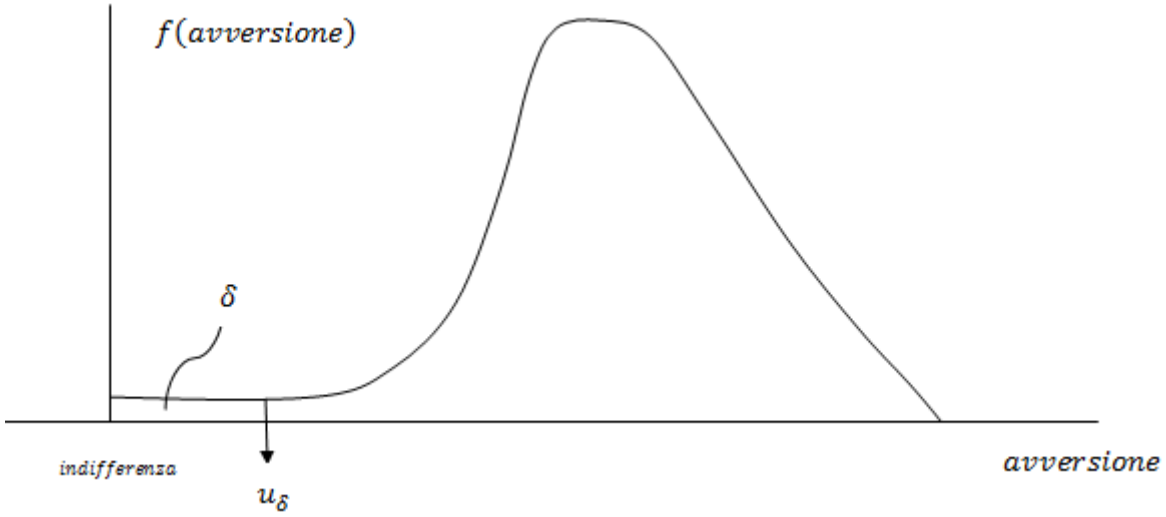
La tecnica attuariale ha storicamente una famiglia di metodologie atte ad individuare la rischiosità del singolo individuo (o per meglio dire della classe di rischio omogenea) seguendo il principio che il premio debba essere proporzionato al livello di rischio.

Ne segue che se fossimo in grado di assegnare ad ogni classe omogenea di rischio utilizzata nel pricing un livello di avversione allora ordinando le diverse classi di pricing in base all'avversione otterremmo la distribuzione dell'avversione di portafoglio e quindi saremmo in grado di applicare il metodo.

8 Stima della perdita per frode subita e definizione del requisito di capitale

Fin'ora ci siamo focalizzati sulla percentuale δ che indica la quota di assicurati che nonostante una fraud detection performance della compagnia di $1 - F^*$ hanno comunque convenienza a tentare la frode perché gli garantisce una maggior utilità attesa.

Questa percentuale rappresenta la quota di contratti per i quali l'azienda accetta la possibilità di subire una perdita per frode. E' quindi evidente la necessità di associare a questa quota una misura monetaria della perdita che l'azienda potrebbe sostenere così da calcolare un requisito di capitale in grado di garantire la solvibilità della compagnia per quanto riguarda il modulo fraud risk.



E' bene evidenziare che qui si sta cercando di stimare il costo della frode che la compagnia si attende di subire nonostante una performance nelle tecniche di antifrode pari a F^*. Questo non è l'intero costo della frode assicurativa perché vanno considerate anche tutte le spese ed investimenti fatti dalla compagnia nell'adeguamento e miglioramento del suo apparato anti frode.

Nei capitoli precedenti ci siamo chiesti, dato quindi un portafoglio di n assicurati, quale fosse il valore di equilibrio F^* (che rappresenta probabilità di riuscire nella frode) tale per cui solo una quota di assicurati pari a δ^* avesse convenienza a tentare la frode. Questa quota rappresenta quel subset di assicurati la cui avversione al rischio è più bassa rispetto a quella percentilica scelta dalla compagnia e che quindi hanno un'utilità attesa nel provare la frode maggiore rispetto allo scenario in cui rinunciano al guadagno illecito.

Quindi il numero di assicurati per i quali esiste ancora un rischio frode è pari a

$$m = n\delta^*$$

a fronte di tale quota di assicurati vogliamo stimare un valore monetario che rappresenti il costo complessivo atteso della frode subita dalla compagnia nell'anno di riferimento.

Si ricordi che nelle assunzioni di modello si era imposto che per tutti i contratti la percentuale di copertura α fosse pari al 100% e che quindi il risarcimento in caso di sinistro fosse esattamente pari alla perdita subita

$$L = R$$

per semplicità assumiamo che l'evento assicurato comporti necessariamente la distruzione totale del bene coperto e che quindi la perdita in caso di sinistro sia una variabile dicotomica che assuma valore 0 o L.

E' quindi di tutta evidenza che la perdita per sinistro equivale a sua volta al valore assicurato nel contratto

$$L = R = VA$$

questo ci da una prima misura monetaria della perdita che una singola frode (nei limiti della definizione data nel modello) può causare all'azienda, dobbiamo ora cercare di estendere il discorso a tutti gli m assicurati esposti al rischio frode.

Posto che nella base campionaria disponibile sia noto per ogni individuo la stima del suo grado di avversione (definito come uno score tra 0 e 1) evidentemente la compagnia sa quali sono questi m assicurati esposti al rischio frode e disporrà di tutta l'informazione relativa ai loro contratti sottoscritti.

Abbiamo quindi un data base con m valori assicurati sulla base dei quali cerchiamo di fittare una adeguata distribuzione di probabilità che andrà a modellizzare il singolo costo per frode.

Formalmente chiameremo con Z^F il costo del sinistro simulato dall'assicurato che tenta la frode, per le assunzioni di cui sopra sappiamo che questo sarà pari al valore assicurato relativo ai soli individui effettivamente esposti al rischio frode.

Possiamo quindi definire il costo aggregato della frode subita nell'anno come

$$X^F = \sum_{i=1}^{m} Z_i^F F_i^*$$

ove $F_i^* \sim Be(F^*)$.

Nella sostanza ad ogni assicurato esposto al rischio frode si associa una dummy che assume valore 0 o 1 in probabilità al fine di simulare la riuscita o meno della frode da parte

dell'assicurato. La probabilità di successo della Bernoulli è data dal valore di equilibrio definito nei capitoli precedenti.

Possiamo osservare come (per ora) il costo aggregato della frode subita rispecchia in parte quello che è il costo aggregato dei sinistri per una compagnia vita ove

1. Le somme assicurate sono note a priori (nel nostro caso il costo della frode del contratto $i-$ esimo è pari al valore assicurato in base alle ipotesi fatte)
2. L'evento assicurato è di tipo dicotomico (nel nostro caso il costo del sinistro simulato avviene solo nel caso in cui l'azienda non scopra la frode)

La definizione di cui sopra è però ancora incompleta perché non tiene conto del fatto che se l'assicurato esposto al rischio frode effettivamente sperimenta il sinistro questi non ha più motivo di tentare alcun comportamento illecito avendo già una copertura α pari ad 1.

Possiamo quindi schematizzare il processo con cui un assicurato può generare il costo di un sinistro alla compagnia come:

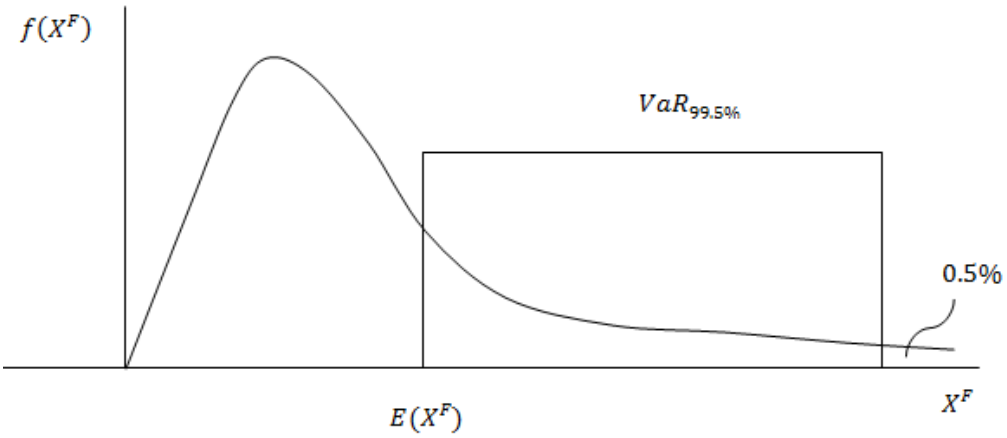
Assicurato

1. non esposto al rischio frode → accadimento del sinistro (CRM classico)
2. esposto al rischio frode
    a. accade il sinistro (CRM classico)
    b. non accade il sinistro → riesce nella frode

Formalmente indicheremo con m^* un intero compreso tra 0 ed m che va a rappresentare il numero di assicurati esposti al rischio frode per i quali non si verifica il sinistro, quindi il costo complessivo per la frode subita sarà pari a

$$X^F = \sum_{i=1}^{m^*} Z_i^F F_i^*$$

gli altri $m - m^*$ assicurati esposti al rischio frode ma per i quali si verifica il sinistro saranno genereranno un costo modellizzabile con il CRM classico (posto che ne valgano le ipotesi sottostanti).



Stimata la distribuzione di X^F (verosimilmente con un approccio simulato vista la complessità del processo di cui sopra) è possibile stimare la perdita attesa per frode subita nell'anno ed il requisito di capitale per il modulo frode (ad esempio un modello $VaR_{99.5}$ coerentemente con la logica SII).

Possiamo quindi definire il costo complessivo dei sinistri di una data LOB e/o portafoglio in logica CRM come

$$X = \sum_{i=1}^{n-m} Z_i + \sum_{i=1}^{m^*} Z_i^F F_i^* + \sum_{i=1}^{m-m^*} Z_i^F$$

ove le tre sommatorie rappresentano rispettivamente

1. il costo aggregato dei sinistri per tutti gli assicurati che non sono esposti al rischio frode
2. il costo aggregato della frode subita relativamente agli assicurati esposti al fraud risk per i quali non accade il sinistro
3. il costo aggregato dei sinistri per gli assicurati esposti al fraud risk per i quali accade il sinistro e quindi perdono interesse nella frode

Ovviamente per una tale definizione devono valere tutte le ipotesi del CRM e si osservi che se la distribuzione del valore assicurato tra tutti gli assicurati è la stessa e questi sono tra loro indipendenti sarebbe possibile semplificare il modello accorpando la prima ed ultima sommatoria.

Da questa scrittura di X^F capiamo inoltre come l'avere una performance nell'individuazione della frode inferiore a quella di equilibrio definita da $1 - F^*$ comporta un aumento della X^F rispetto alle aspettative perché aumenterà m e la probabilità di riuscire nella frode.

Bibliografia

“Statistical Learning Theory” di Vladimir N. Vapnik AT&T Research Laboratories

“The Nature of Statistical Learning Theory” di Vladimir N. Vapnik Springer Second Edition

“An Overview of Statistical Learning Theory” di Vladimir N Vapnik IEEE Transactions on Neural Networks, vol. 10 NO 5 Settembre 1999

“Support Vector Machines for Pattern Classification” di Shigeo Abe APR Second Edition

“The Elements of Statistical Learning: Data Mining, Inference and Prediction” di Trevor Hastie, Robert Tibshirani e Jerome Friedman Springer Second Edition

“An Introduction to Statistical Learning with Applications in R” di Gareth James, Daniela Witten, Trevor Hastie e Robert Tibshirani Springer

“Support Vector Machine in R” di Alexandros Karatzoglou e David Meyer, Journal of Statistical Software

“Support Vector Machines” di Marco Sciadrone, appunti delle lezioni dell’a.a. 2005-06

“Kernel Methods in Machine Learning” di Thomas Ofmann, Bernhard Scholkopf e Alexander J. Smola, The Annals of Statistics 2008 Vol 36 No 3

“Kernel Methods for Pattern Analysis” di Nello Cristianini e John Shawe Taylor, Cambridge 2004

“Tecniche di DM: Alberi di decisione ed algoritmi di classificazione” di Vincenzo Antonio Manganaro, Palermo

“An Introduction to Variable and Feature Selection” di Isabelle Guyon e André Elisseeff, Journal of Machine Learning Research 2003

“Feature Selectiong per la Classificazione” di F. Rinaldi, Università La Sapienza Roma

“Kernel Principal Components Analysis” di Max Welling University of Toronto

“Kernel Principal Components Analysis” di Bernhard Sholkopf, Alexander Smola e Klaus Robert Muller Bernil University

“Machine Learning and Statistical Techniques. An Application to Prediction of Insolvency in Spanish Non-life Insurance Companies” di Zuleyka Diaz, Maria Jesus Segovia e Jose Fernandez Universidad Complutense de Madrid

“Frodi nel settore assicurativo” di KPMG Advisory Settembre 2011

"Il quadro normativo nella prevenzione delle frodi e la prassi operativa delle imprese assicurative" di Pietro Negri Seminario AIMAV, Ottobre 2014

"Relazione Antifrode 2013" ANIA

"Fraud Risk Management" KPMG Advisory 2006

"Fraud Detection Using Reputation Features, SVMs and Random Forests" di Dave DeBarr e Harry Wechsler George Mason University

"The State of Insurance Fraud Technology" SAS Coalition Against Insurance Fraud September 2014

"Support Vector Machine Ensemble with Bagging" di Hyun-Chul Kim, Shaoning Pang e Hong-Mo Je, Pohang University Korea

"Asymmetric-margin support vector machines for lung tissue classification" di Adrien Depeursinge, Gilles Cohenm, Antoine Greissbuhler e Henning Muller

"Complementi di Analisi Statistica Multivariata" di Benito V. Frosini EDUCatt Milano 2013