Data Mining e Bioimmagini

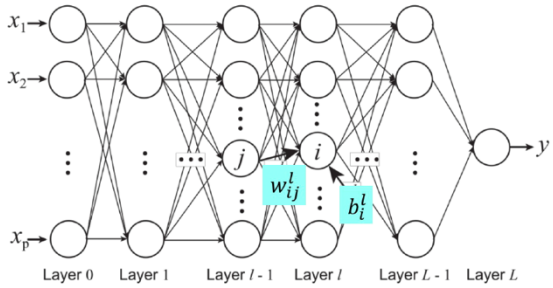
Reti multilivello, SVM, Ensemble, Weka

Prof. Pierangelo Veltri – 31/10/2023- Autori: Panarello, Saeedzadeh - Revisionatori: Panarello, Saeedzadeh

RETI MULTILIVELLO E FUNZIONE DI ATTIVAZIONE

La struttura di una rete multilivello (ad esempio una rete neurale) prevede una serie di nodi posti a livelli diversi interconnessi tra loro. In particolare, la complessità della rete è data dal numero di layer che la formano e da come sono strutturate le connessioni tra i nodi (input-output). Nell'architettura di una rete multilivello, infatti, sono fondamentali le **funzioni di attivazione** di ogni nodo intermedio, in quanto in base a come vengono concatenate si hanno input/output diversi (*NOTA: il prof non va troppo nello specifico perché lo ritiene un argomento troppo tecnico*). Ogni ingresso, inoltre, viene pesato da un valore weight ed un valore che rappresenta il rumore; quindi, se x è l'ingresso esso viene moltiplicato per il weight (w) e sommato al rumore b per ottenere l'output y:

$$y = x \cdot w + b$$



$$a_i^l = f(z_i^l) = f\left(\sum_j w_{ij}^l a_j^{l-1} + b_i^l\right)$$

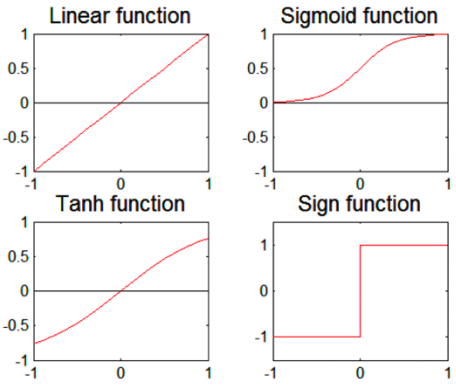
Activation value at node i at layer l

Activation Function

Linear Predictor

Le funzioni di attivazione possono essere di diversi tipi:

- Lineari
- A gradino
- Logaritmica
- Ad andamento tangenziale



$$a_i^l = \sigma(z_i^l) = \frac{1}{1+e^{-z_i^l}}$$

)

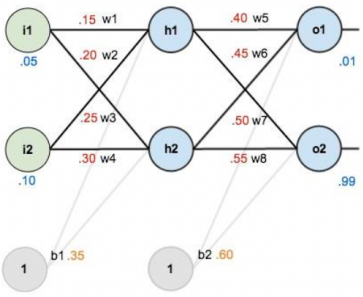
$$a_i^l = \tau(z_i^l) = \frac{e^{z_i^l} - e^{-z_i^l}}{e^{z_i^l} + e^{-z_i^l}}$$

$$\frac{\partial a_i^l}{\partial z_i^l} = \frac{\partial \tau(z_i^l)}{\partial z_i^l} = 1 + a_i^l$$

Per stabilire l'efficienza e l'efficacia di una rete è necessario il calcolo dell'errore; infatti, minimizzando l'errore si può misurare il peso e il rumore. Per tale calcolo viene usato un meccanismo di convergenza basato sulle derivate, dove si calcola la variazione dell'errore utilizzando delle funzioni che si attestano verso un target.

Nell'esempio riportato, sono definiti dei valori di peso (w1= 0.15 peso tra nodo di input 1 e hidden 1 (nodo intermedio)) e da questi si calcola l'output come prodotto del peso per lo specifico ingresso (i1 per w1) e si applica alla funzione di attivazione (che ricordiamo può avere un andamento lineare, logaritmico, ecc.). L'errore viene poi minimizzato tramite modifica dei pesi all'interno dei nodi e ripetizione dei calcoli.

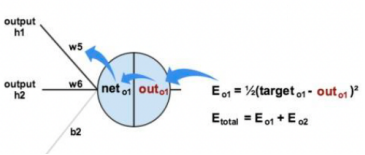
INPUT	TARGET
ι1 = 0.05	o1 = 0.01
ι2 = 0.10	o2 = 0.99



2. Backward step – Modifica pesi

Di quanto w5 influenza l'errore totale?

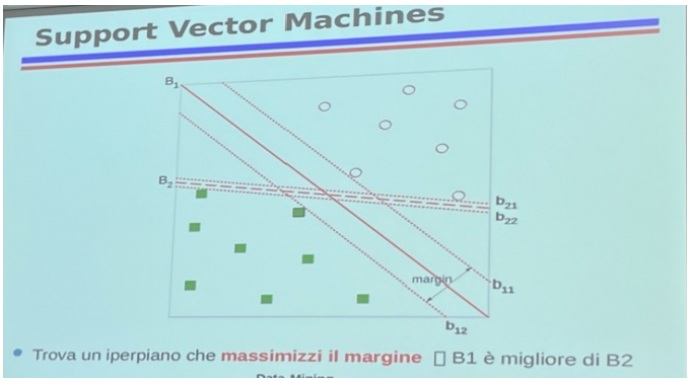
$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial w_5}$$



Ricapitolando, data una rete neurale dotata di una serie di funzioni di attivazione connesse ai vari nodi della rete, vengono definiti dei pesi sugli archi con lo scopo di ottenere i risultati di classificazione corretti con il minor peso. Per far ciò è necessario misurare l'errore ad ogni step (verificando che sia sotto una data soglia) tramite il calcolo della derivata parziale, ossia la differenza tra i valori di un certo livello rispetto alle differenze totali.

SUPPORT VECTOR MACHINE (SVM)

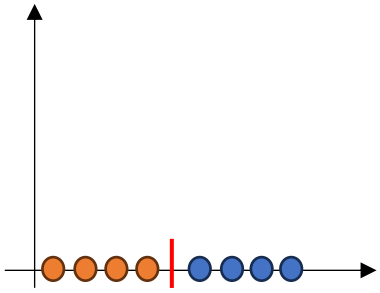
La Support Vector Machine è una tecnica che consente di identificare un oggetto geometrico (nel caso bidimensionale una retta) che consente di dividere lo spazio in due set separati (serve quindi a fare classificazione); il concetto di separazione si basa su un training factor (allenamento della macchina). Dato un test set, dunque, la SVM è in grado di classificare l'elemento come facente parte di una classe in base alla disposizione nello spazio geometrico. La classificazione, infatti, dipende soprattutto dalla distribuzione dei dati nello spazio e dalla tolleranza rispetto all'outlier. Di base i *support vector classifier* (la retta o l'iperpiano che divide lo spazio geometrico) possono essere infiniti. Esistono diverse tecniche di divisione dello spazio; generalmente si ricerca l'iperpiano che massimizzi il margine, cosicché le distanze tra gli oggetti più vicini al margine per ogni classe siano minime (nell'immagine b_{21} e b_{22} hanno distanza minima dal margine B_2 rispetto al margine B_1, quindi tale margine viene scelto come iperpiano).

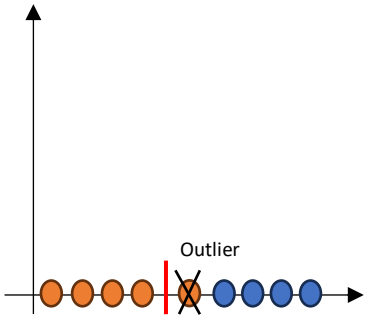


NOTA: La soglia si chiama support vector classifier; vector perché lo spazio di classificazione è vettoriale (ogni punto è considerato un vettore con modulo, direzione e verso)

Esempi

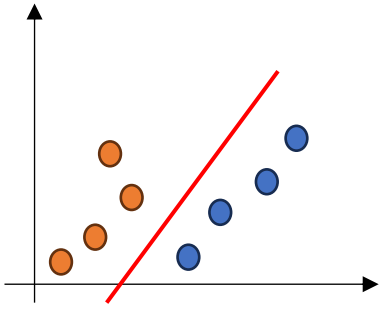
Sia dato il seguente dataset di un insieme di pazienti in cui i pallini arancioni rientrano nella classe NON OBESI mentre i blu in quella OBESI. In questo caso unidimensionale ed ordinato, il valore soglia che discrimina tra le due classi risulta "semplice" da individuare grazie alla distribuzione uniforme dei dati ed è indicato in rosso. Tale valore soglia risulta equidistante dal quarto pallino arancione e dal primo pallino blu.





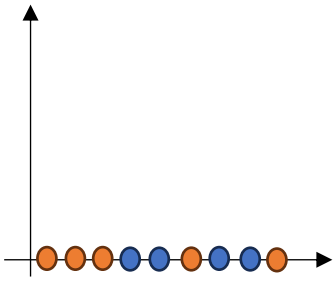
Se si aggiunge un valore che noi sappiamo appartenere alla classe NON OBESI ma che di fatto rientra nello spazio geometrico della classe OBESI (come mostrato in figura), si tende ad indicare tale valore come outlier e lo si ignora nella misurazione del support vector classifier; tuttavia, se dovesse arrivare un ulteriore valore con analoghe caratteristiche, quest'ultimo non potrà essere ignorato e verrà indicato come OBESO.

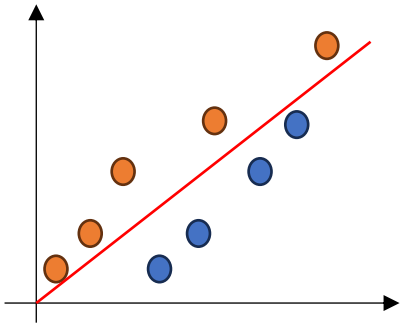
In quest'altro esempio abbiamo, invece, una variabile a due dimensioni, ossia peso + altezza; qui si ha una retta come classificatore. Nel caso di un dataset di variabili a tre dimensioni, si avrebbe un iperpiano come divisore.



Nel seguente caso, la classificazione non è immediata in quanto gli oggetti per la variabile indicata sono "mescolati"; non risulta possibile, dunque, individuare il classificatore. Per risolvere tale situazione:

- Si prendono gli oggetti e si spostano (mediante tecnica omomorfica) in uno spazio di un'altra dimensione k
- Si cerca la soglia in questa nuova dimensione.





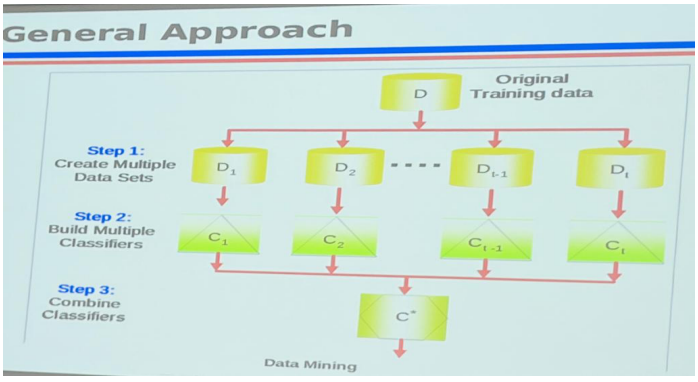
Ad esempio, per k = 2 si eleva ogni valore al quadrato; quello che ottengo è uno spazio in cui gli oggetti che erano posizionati sull'asse x (dimensione 1) possono essere più facilmente divisi in una modalità che permette di trovare un iperpiano che nella precedente dimensione non poteva essere individuato. La funzione che permette di passare da uno spazio all'altro deve essere invertibile in modo che sia sempre possibile, partendo da un oggetto elevato a k, ritornare alla dimensione iniziale.

NOTA: questo è un esempio, il passaggio da una dimensione all'altra è legata alla funzione che sta alla base della classificazione (in questo caso $y=x^k$); in sostanza viene stabilita da chi fa classificazione in base al dataset che ha a disposizione!

Questo approccio si usa quando ci sono troppi outlier, casi di missclassification e non è possibile individuare un iperpiano adatto alla classificazione.

ENSEMBLE

Nel quinto capitolo del libro, viene affrontato il tema degli ensemble. Si cerca di comprendere la possibilità di combinare quanto è stato precedentemente esaminato, sia in termini di tecniche di classificazione, sia in termini di dati di classificazione. L'obiettivo è ottimizzare i meccanismi di classificazione. Ad esempio, supponiamo di avere un training set che costituisce il 70% della nostra tabella. Se desideriamo utilizzare questo training set con diverse tecniche, come l'approccio basato su regole, alberi decisionali o SQL, ci poniamo la domanda: come possiamo evitare che questo dataset sia sfruttato esclusivamente da una singola tecnica?
Per affrontare questa sfida, possiamo adottare un approccio che coinvolge la creazione di ensemble di dati. In altre parole, possiamo generare nuovi dataset derivati dal nostro dataset originale, ognuno con le caratteristiche desiderate. Successivamente, possiamo utilizzare questi nuovi dataset per allenare i classificatori utilizzando la stessa tecnica.
Ad esempio, consideriamo l'uso di alberi decisionali. Creeremo vari sottoinsiemi dei dati originali e addestreremo diversi alberi decisionali su ciascuno di essi. Infine, possiamo combinare questi diversi alberi decisionali per ottenere un classificatore più affidabile e potenziato.



Questo approccio di ensemble offre un modo promettente per migliorare l'accuratezza e l'affidabilità delle nostre tecniche di classificazione, consentendoci di sfruttare appieno il nostro dataset di allenamento in modi diversi e complementari.
Si tratta di una tecnica di ensemble e ciò che è stato appena descritto è conosciuto come "Random Forests". In altre parole, si tratta di creare una serie di alberi decisionali utilizzando diversi input e combinare questi alberi decisionali in un insieme per ottenere una previsione più accurata.

L'ensemble method consiste in:

- Manipolare la distribuzione dei dati ingresso (bagging, boosting)
- Manipolare caratteristiche di ingresso (random forests)
- Manipolare le classi, cioè cercare di cambiare la classificazione dei dati di ingresso al fine di ridurre il rumore (error-correcting output coding)

Qual è il concetto chiave che deve essere mantenuto nelle tecniche di classificazione basate sugli ensembles? Uno di essi è la possibilità di manipolare l'input e le caratteristiche, cioè apportare modifiche alle caratteristiche che si basano principalmente su soglie. Una feature può essere considerata sopra o sotto una soglia per determinare l'appartenenza a una classe. Ad esempio, modificare le soglie basate sui dati può influenzare la classificazione. Variando le soglie in base ai dati in ingresso, si apportano cambiamenti nella definizione dell'appartenenza alle classi, causando lievi modifiche nella classificazione dei dati. In particolare, le tecniche che si utilizzano nella definizione dei nostri sistemi di classificazione sono la **precision** e la **recall**, date dalle seguenti formule:

$$\text{Precision, } p = \frac{TP}{TP + FP}$$

$$\text{Recall, } r = \frac{TP}{TP + FN}$$

Vero positivo (TP) o f++, che corrisponde al numero di esempi positivi correttamente previsti dal modello di classificazione.

Falso negativo (FN) o f+-, che corrisponde al numero di esempi positivi erroneamente previsti come negativi dal modello di classificazione.

Falso positivo (FP) o f-+, che corrisponde al numero di esempi negativi erroneamente previsti come positivi dal modello di classificazione.

Vero negativo (TN) o f--, che corrisponde al numero di esempi negativi correttamente previsti dal modello di classificazione.

Nel caso della precisione, si calcola sostanzialmente il numero di veri positivi rispetto al totale dei valori che sono stati classificati come positivi. Questo significa quanti sono stati correttamente classificati come positivi rispetto a tutti quelli che sono stati etichettati come positivi, sia quelli che erano correttamente classificati sia quelli che erano erroneamente classificati. In definitiva, la precisione misura il valore complessivo dell'accuratezza del classificatore.
Nell'ambito del recall, invece, si misura sostanzialmente quanti valori positivi sono stati correttamente predetti dal classificatore rispetto ai veri positivi e ai falsi negativi (quindi la totalità reale dei positivi).

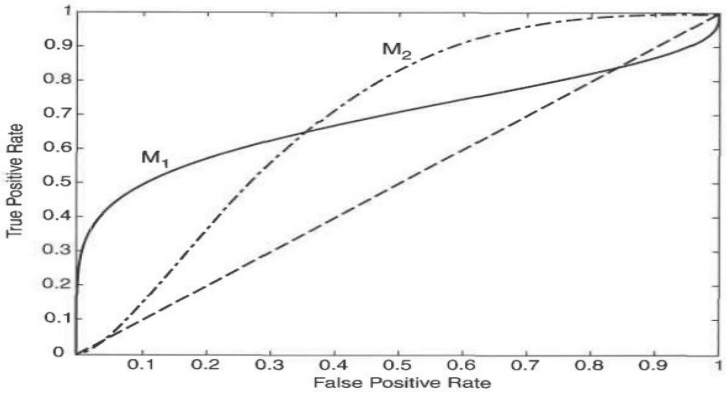


Figure 5.41. ROC curves for two different classifiers.

Questi due valori (precision e recall) sono utilizzati per calcolare i grafici essenziali nella rappresentazione complessiva di un classificatore, tra cui la curva caratteristica operativa del ricevitore, comunemente nota come **curva ROC**. La curva ROC rappresenta essenzialmente il tasso di falsi positivi rispetto al tasso di veri positivi.
Idealmente, quando il tasso di falsi positivi è prossimo a zero e il tasso di veri positivi è molto alto, il nostro classificatore è altamente affidabile, poiché commette pochi errori globali e classifica correttamente una grande parte dei casi positivi. In altre parole, una curva ROC che si avvicina al margine sinistro del grafico indica che il classificatore funziona bene in generale.

WEKA

Weka è uno strumento open source sviluppato dai ricercatori dell'Università di Waikato.
Per installare Weka puoi seguire questi passi generali:

1. Vai sul sito ufficiale di Weka: https://www.cs.waikato.ac.nz/ml/weka/
2. Nella sezione "Download," troverai il link per scaricare l'ultima versione di Weka. Fai clic su quel link per accedere alla pagina di download.
3. Scegli la versione di Weka adatta al tuo sistema operativo. Weka è disponibile per Windows, macOS e Linux.
4. Una volta scaricato il file di installazione, segui le istruzioni specifiche per il tuo sistema operativo per completare l'installazione.

Una volta installato, per accedere agli esempi in Weka, puoi seguire questi passi:

1. Dopo aver installato Weka, avvia il programma.
2. Nella schermata principale di Weka, seleziona la scheda "Explorer".
3. Nella parte superiore sinistra, dovresti vedere un'opzione "Open file". Clicca su di essa.
4. Questo ti porterà a una finestra di dialogo per selezionare il file dati che desideri esaminare.
5. Naviga nella directory in cui hai installato Weka e cerca la cartella "wekadoc/examples".
6. All'interno della cartella "examples", troverai i file di esempio come "breast-cancer.arff", "diabetes.arff" e altri. Seleziona il file di dati che desideri esaminare e fai clic su "Open" o "Apri".

A questo punto, il file di dati selezionato verrà caricato in Weka e potrai esplorare e analizzare gli esempi clinici presenti nel dataset.
Weka offre la possibilità di selezionare e utilizzare diversi classificatori per analizzare i dati.

1. Dopo aver caricato il tuo dataset o eseguito la pre-elaborazione dei dati in Weka, vai alla scheda "Classify" (Classificare) nella parte superiore della finestra.
2. Nella scheda "Classify", vedrai diverse opzioni per selezionare il classificatore. Queste opzioni includono "Choose" (Scegli), "Lazy" (Pigri), "Rules" (Regole), "Functions" (Funzioni), "Meta" (Meta), "Trees" (Alberi), ecc.
3. Clicca su "Choose" per vedere una lista di classificatori disponibili all'interno di ciascuna categoria. Seleziona il classificatore che desideri utilizzare.
4. Ogni classificatore ha le sue impostazioni e parametri specifici che puoi configurare secondo le tue esigenze.
5. Dopo aver selezionato il classificatore, puoi fare clic su "Start" o "Build" per avviare il processo di classificazione. Weka eseguirà il classificatore scelto sui dati e mostrerà i risultati, inclusi i valori di precisione, recall e altre metriche.

Puoi esplorare diversi classificatori in Weka per vedere quale funziona meglio per i tuoi dati specifici. Puoi anche sperimentare con diverse impostazioni e parametri per ottimizzare le prestazioni del classificatore.
R è uno strumento ampiamente utilizzato nell'ambito del data mining e nella statistica, grazie alla sua semplicità d'uso. Tuttavia, per affrontare tecniche più avanzate, come la classificazione o la creazione di reti neurali, spesso si passa a Python o ad altri strumenti, ad esempio Weka, per le reti neurali e il deep learning.