Università di Pisa

Facoltà di Ingegneria

Corso di Laurea Specialistica in Ingegneria Informatica

Anno Accademico 2005/2006

TESI DI LAUREA MAGISTRALE

Una nuova misura di performance per classificatori automatici con applicazione a sistemi CAD polmonari

Candidato

Francesco Marafini _____

Relatori

Prof.sa Beatrice Lazzerini _____

Prof. Francesco Marcelloni _____

Ing. Marco Cococcioni _____

Indice

Elenco delle figure

Capitolo 1

Introduzione

Una delle principali cause di morte per neoplasia al mondo è rappresentata senza dubbio dal cancro al polmone[1]. Il tumore polmonare è raramente guaribile e la percentuale globale di sopravvivenza[1] a 5 anni varia dal 10% al 15%. Se la diagnosi è effettuata tempestivamente, ovvero al I stadio, la sopravvivenza cresce notevolmente salendo al 70-80%. Ciò ha spinto i ricercatori a rendere la diagnosi il più precoce possibile. L'obiettivo è la scoperta del tumore quando è ancora operabile, ovvero nella sua fase iniziale, per aumentare la percentuale di sopravvivenza del paziente e contemporaneamente diminuire la mortalità dell'intervento chirurgico. Data la tipica assenza di sintomi nei tumori polmonari precoci e curabili, è stato necessario ricorrere al controllo sistematico della popolazione a rischio[2] tramite programmi di screening[2]. I primi screening sono stati effettuati negli anni 70-80 con l'utilizzo della radiografia toracica. I limiti diagnostici di questa tecnica risiedono principalmente nell'impossibilità di rilevare lesioni polmonari con dimensioni pari o inferiori ai 2 cm di diametro, specie se situate in posizioni sfavorevoli quali dietro al cuore o a una costola.

La *Tomografia Assiale Computerizzata* (TAC)(fig.1.1), che permette di ricostruire in 3 dimensioni l'interno del corpo umano con l'uso dei raggi X, ha aperto nuovi orizzonti nella diagnosi precoce del cancro al polmone[3]. Tra le evoluzioni della TAC tradizionale, due sono le tecniche di particolare importanza:

- La *TAC spirale*: in questa tecnica lo strumento viene fatto ruotare attorno al paziente come se questi fosse avvolto in una spirale (fig.1.2).

[1] La percentuale di sopravvivenza globale è riferita a tutti gli stadi del tumore dal I al IV.
[2] I soggetti da sottoporre al test sono di solito selezionati in base al numero di sigarette fumate e all'età.

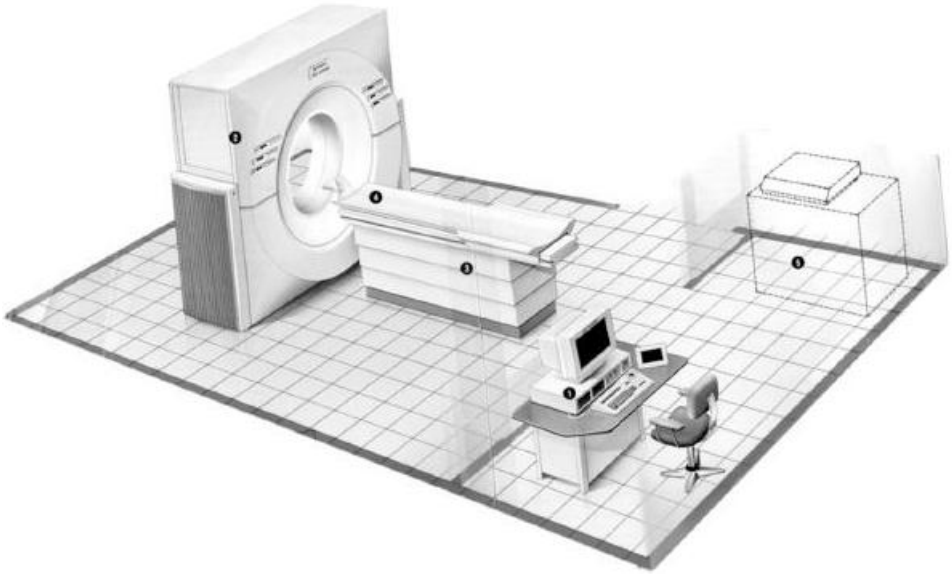


Figura 1.1: Schema di una tipica apparecchiatura TAC, formata da (1) console di controllo, (2) piattaforma di emissione, (3) tavolo per il paziente, (4) poggiatesta e (5) imager laser.

- La *TAC multistrato*: in questa tecnica l'organo da esaminare viene scansionato in sezioni sottilissime. Lo strumento fornisce fino a 8 immagini al secondo.

Il grande vantaggio della TAC spirale risiede nella possibilità di ottenere una acquisizione volumetrica durante una singola scansione. All'interno del tubo radiogeno è presente una corona composta da un migliaio di rivelatori. Il tubo ruota attorno al paziente e l'intensità dei raggi X viene misurata dai sensori dopo l'attraversamento dei tessuti. I dati sono poi inviati ad un calcolatore che calcola la densità dei tessuti e, da questa, produce immagini in scala di grigio delle sezioni del corpo. Le immagini delle sezioni, dette "fette", possono essere sottili fino ad 1mm. Questo livello di dettagli permette di rilevare lesioni fino a 2-3mm nelle immagini, bidimensionali o tridimensionali, ricostruite tramite l'uso di software dedicati. Ciò è fondamentale perchè i piccoli noduli sono molto spesso l'espressione di un tumore ad uno stadio precoce e, in quanto tale, più curabile.

La durata di un esame TAC è di 20 secondi, periodo durante il quale il paziente deve rimanere in apnea. Un esame TAC secondo i protocolli standard non è utilizzabile in un programma di screening in quanto il livello di radiazioni previsto è molto elevato. A questo scopo è però possibile utilizzare la TAC spirale a bassa dose di radiazioni essendo stato dimostrato che tale dose ridotta non

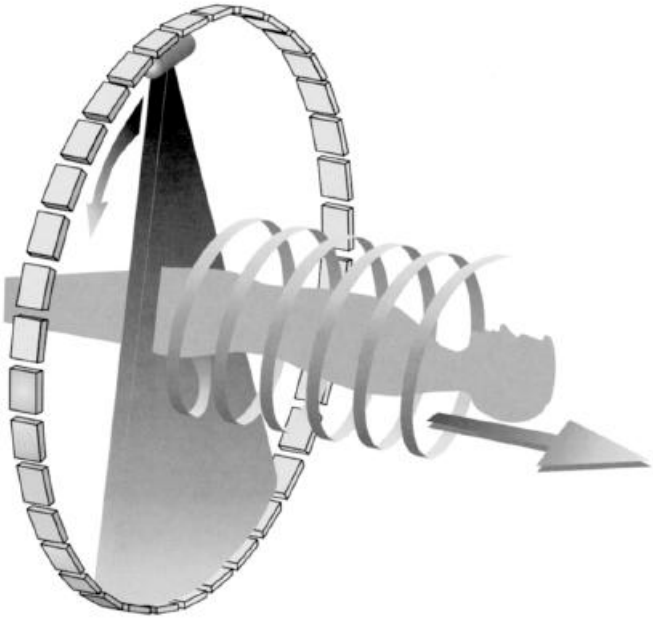


Figura 1.2: Lo scanning a spirale fa si che il punto focale segua una traiettoria elicoidale attorno al paziente

interferisce con le capacità diagnostiche dell'esame. La sensibilità nei confronti dei noduli con diametro inferiore a 5mm, micronoduli, non viene diminuita e l'esame è perciò adatto per effettuare diagnosi precoci.

Per garantire la reale efficacia di un programma di screening va ridotto il rischio di *sovradiagnosi*, ovvero il riconoscimento di falsi positivi, e soprattutto si deve evitare di incorrere nella *sottodiagnosi*, cioè la mancata rivelazione di noduli. Nello specifico, un adeguato test di screening deve possedere le seguenti caratteristiche:

- *Specificità:* questa carattestica è definita come la proporzione, rispetto al totale dei soggetti, dei pazienti sani correttamente classificati come negativi dal test.
- *Sensibilità:* questa caratteristica è definita come la proporzione, rispetto al totale dei soggetti, dei pazienti malati correttamente classificati come positivi dal test.

L'analisi delle immagini prodotte da un esame TAC presenta alcune difficoltà.

Il numero di immagini per ciascun paziente che il radiologo deve esaminare è

molto elevato. Una TAC multistrato a collimazione sottile produce infatti circa 300 immagini molto spesso rumorose[4].

Il rischio di riconoscimento di falsi negativi e falsi positivi è molto elevato. Bronchi, vasi sanguigni e noduli hanno caratteristiche molto simili per quanto riguarda dimensione, densità e forma. Le strutture di interesse da riconoscere all'interno delle singole immagini si confondono perciò con le normali strutture anatomiche.

Per far fronte a queste difficoltà, soprattutto al rischio di sbagliare diagnosi, ci sono alcune alternative.

Una di queste è l'impiego di più radiologi per analizzare indipendentemente le immagini[5].

Un'altra efficace alternativa è costituita dall'utilizzo di un sistema C.A.D. come ausilio al radiologo per una diagnosi veloce e accurata[2]. A seconda dello scopo del sistema l'acronimo *C.A.D.* assume due differenti significati:

1. *Computer Aided Detection.* E' la modalità più diffusa nella quale il sistema è considerato come "secondo lettore". Il sistema CAD ha il compito di evidenziare nelle immagini digitalizzate le aree con sospetta presenza di noduli, senza discriminare tra noduli maligni e benigni. Il vantaggio principale consiste nella possibilità di individuare noduli che il radiologo non avrebbe visto dato lo spessore sottile delle "fette" della scansione TAC e la facilità di confusione del nodulo con i vasi sanguigni adiacenti. Lo svantaggio principale consiste nell'aumento del numero di candidati noduli da analizzare e quindi nell'aumento del rischio di diagnosticare falsi positivi.
2. *Computer Aided Diagnosis.* Il sistema CAD, oltre a rilevare automaticamente le aree in cui sono presenti i noduli, fornisce indicazioni e stime sulla probabilità che tali noduli rappresentino tumori maligni.

Questi sistemi sono però visti come uno strumento di supporto e di aiuto alla diagnosi del radiologo. La vera innovazione consiste nel passare dal concetto di diagnosi assistita al concetto di *Diagnosi Computerizzata* nell'ambito del CAD polmonare.

La struttura di un sistema CAD prevede le seguenti funzioni:

1. Estrazione delle regioni polmonari.
2. Rilevazione delle *regioni di interesse* (ROI).
3. Classificazione delle ROI in *noduli/vasi sanguigni.*

4. Classificazione dei noduli in *maligni/benigni*

Il passo 1, chiamato *segmentazione*, ha lo scopo di limitare la ricerca dei noduli all'interno del polmone, tramite la selezione, all'interno di ciascuna immagine, delle aree relative ai lobi polmonari.

Il passo 2 consiste nella ricerca delle regioni in cui è presente un nodulo, caratterizzate di solito da determinate caratteristiche grafiche quali il livello di grigio.

Il passo 3 consiste nell'analisi delle caratteristiche delle ROI al fine di classificare la struttura anatomica, presente all'interno della ROI, e discriminare le ROI contenenti vasi sanguigni da quelle contenenti noduli, le quali verrano sottoposte ad ulteriore analisi nel passo 4.

Il passo 4 consiste nella classificazione dei noduli presenti nelle ROI in benigni e maligni, tenendo conto delle varie caratteristiche estratte dalla regione quali, ad esempio, la forma della lesione e il grado di "smoothness".

Presso il Dipartimento di Ingegneria Informatica dell'Università di Pisa è in realizzazione un prototipo di sistema per la *Computerised Diagnosis* del tumore al polmone.

Nell'ambito di tale complesso progetto, sono stati dati contributi in due aree: il *clustering robusto*, e sugli *indici di performace delle curve ROC* ottenute nei passi 2, 3 e 4 del sistema CAD.

Il presente lavoro di tesi è stato organizzato come segue.

Nel capitolo 2 verrà descritto il problema della *segmentazione delle immagini TAC* e verrà introdotto un algoritmo per il clustering robusto.

Il capitolo 3 sarà dedicato alla valutazione delle performance dei classificatori; in particolare verranno presentate la ROC Analysis, le metriche di bontà di un classificatore con particolare attenzione all'AUC e sarà descritta e valutata un'estensione di questa metrica, da noi chiamata tAUC, che ci ha permesso di introdurre una nuova tecnica di analisi, da noi chiamata *tROC Analysis*, di cui è stata valutata l'efficacia e l'applicabilità.

Nel capitolo 4 saranno descritti i risultati sperimentali dell'applicazione della tROC Analysis ai dati relativi alla rilevazione dei noduli. In questo capitolo saranno anche presentate alcune delle scelte implementative relative al progetto del sistema CAD polmonare.

Nel capitolo 5, infine, saranno illustrate le conclusioni e gli sviluppi futuri.

Bibliografia

[1] S. Sone, et al., "Mass screening for lung cancer with mobile spiral computed tomography scanner", The Lancet, 351, April 1998, pp. 1242-1245.

[2] S.G. Armato III, "CAD dissects growing volume of data from lung CT exams", http://www.diagnosticimaging.com/advancedct2003/.

[3] K. Awai, K. Murao, A. Ozawa, M. Komi, H. Hayakawa, S. Hori, Y. Nishimura, "Pulmonary nodules at chest CT: effect of computer-aided diagnosis on radiologists' detection performance", Radiology, vol. 230, n. 2, February 2004, pp. 347-352.

[4] S.J. Swensen et al., "Screening for lung cancer with low-dose spiral computed tomography", American Journal of Respiratory and Critical Care Medicine, vol. 165, n. 4, February 2002, pp. 508-513.

[5] M.J.R. Dalrymple-Hay, N.E. Drury, "Screening for lung cancer", Journal of the Royal Society of Medicine, vol. 94, January 2001, pp. 2-5.

Capitolo 2

Segmentazione e Estrazione delle ROI

2.1 Introduzione

La segmentazione, termine con il quale ci si riferisce alla estrazione delle regioni polmonari, è la prima operazione compiuta da un sistema CAD polmonare. Essa consiste nell'individuazione, in ciascuna delle immagini relative alle "fette" della TAC, del *parenchyma*[1] polmonare ovvero delle regioni relative ai lobi polmonari, attraverso l'utilizzo di tecniche di elaborazione delle immagini quali il thresholding e opportuni operatori morfologici. Questo approccio si basa principalmente sulla differenza tra l'attenuazione dei raggi X dovuta ai lobi e quella dovuta alle strutture anatomiche circostanti: i lobi polmonari sono caratterizzati da una densità minore rispetto alle strutture che li circondano, quali la cassa toracica, la colonna vertebrale e i tessuti muscolari ed epiteliali del torace, e risultano perciò in zone dell'immagine con un livello di grigio più basso.

La segmentazione svolge l'importante compito di ridurre le aree dell'immagine da analizzare selezionando le parti significative all'analisi e alla ricerca dei noduli. L'accuratezza di questa operazione è cruciale per l'individuazione corretta di eventuali noduli situati nelle zone periferiche del polmone[2]. Se la segmentazione è svolta in maniera scorretta, tali noduli potrebbero essere interpretati come facenti parte della gabbia toracica ed eliminati, in quanto caratterizzati da un livello di grigio molto simile a quello della gabbia stessa.

[1] Il *parenchyma* di un organo è la parte che contiene gli elementi funzionali dell'organo. Ad esempio, nei reni il parenchyma è costituito dai nefroni e nel polmone dagli alveoli. Esso è contrapposto allo *stroma* che è composto dai tessuti di supporto.

[2] Tali noduli sono detti *noduli pleurici*

L'operazione successiva alla segmentazione è l'estrazione delle Regioni di Interesse (ROI). In questa fase, il sistema CAD ricerca all'interno delle aree di immagine dei lobi polmonari, ottenute tramite il procedimento di segmentazione, delle regioni contenenti noduli, alveoli e vasi sanguigni. Una volta individuate e selezionate le ROI all'interno delle immagini, queste vengono analizzate e separate in *noduli* e *non noduli*.

Questa operazione può essere svolta sottoponendo il volume relativo ai lobi polmonari, ottenuto combinando in un agglomerato 3D le sezioni polmonari delle singole "fette", ad un processo di clustering. In particolare, tale processo è volto a separare gli agglomerati in due cluster:

1. il *cluster delle ROI*. Questo cluster contiene i noduli e i vasi sanguigni che sono caratterizzati da pixel scuri.
2. il *cluster del "rumore"*. Questo cluster contiene le strutture anatomiche dei bronchi e l'aria, i quali sono caratterizzati da pixel più chiari.

La scelta di un algoritmo di clustering adatto e specifico per le applicazioni di imaging, e soprattutto robusto nei confronti del rumore, è fondamentale per l'individuazione corretta delle regioni in cui ricercare i noduli. Una estrazione scorretta delle ROI inficierebbe pesantemente la funzionalità e le performance del sistema CAD, in quanto risulterebbe nella possibile perdita di regioni contenenti noduli maligni.

Tra i numerosi algoritmi di clustering robusti, si è scelto di impiegare il *Robust Fuzzy C-Means (RFCM)* descritto in [1]. Tale algoritmo adotta una strategia meno raffinata rispetto ad altri algoritmi di clustering fuzzy, quali l'algoritmo di R.Davè descritto in [2], ma molto più adatta alla nostra applicazione. Tale algoritmo, infatti, incorpora all'interno della funzione obiettivo le informazioni spaziali relative ai pixel dell'immagine.

In questo capitolo andremo quindi a descrivere le caratteristiche dello RFCM nei confronti del Fuzzy C-Means standard. Verrà inoltre presentata l'implementazione MATLAB di tale algoritmo e saranno mostrati i risultati dell'applicazione dell'algoritmo ad un'immagine reale relativa ad una "fetta" di TAC spirale.

2.2 L'algoritmo FCM nel trattamento delle immagini

Nell'ambito delle applicazioni di imaging, l'algoritmo *Fuzzy C-Means (FCM)* classico presenta lo svantaggio di essere particolarmente sensibile al rumore e ad altri artefatti presenti nelle immagini. Ciò è dovuto principalmente al fatto che lo FCM non tiene conto delle informazioni relative al contesto spaziale. Riportiamo di seguito la formulazione matematica dell'algoritmo:

$$\begin{cases} J_{FCM} = \sum_{j\in\Omega} \sum_{k=1}^{C} u_{jk}^{q} \left\| \boldsymbol{y}_j - \boldsymbol{v}_k \right\|^2 \\ \sum_{k=1}^{C} u_{jk} = 1 \end{cases} \tag{2.1}$$

dove $\boldsymbol{y}_j$ è l'osservazione al pixel j, $\boldsymbol{v}_k$ è il centroide del cluster k, u_{jk} è la funzione di appartenenza del pixel j al cluster k, C è il numero dei cluster, Ω è il dominio dell'immagine e q è il grado di "fuzziness".

Come si può vedere nell'equazione (2.1), la funzione obiettivo dello FCM non tiene in considerazione nessun tipo di dipendenza spaziale tra le osservazioni. In questo modo le funzioni di appartenenza calcolate possono mostrare una certa sensibilità al rumore presente nell'immagine osservata. Un tipico approccio per la compensazione di tale sensibilità consiste nel sottoporre l'immagine ad un'operazione di "smoothing" prima di applicare l'algoritmo FCM. I filtri di smoothing standard possono però dar luogo alla perdita di importanti dettagli dell'immagine e, soprattutto, non c'è modo di controllare rigorosamente il trade-off tra lo smoothing e il risultato del clustering.

2.3 L'algoritmo RFCM

Un approccio più classico consiste nell'incorporare le informazioni spaziali penalizzando la funzione obiettivo per vincolare le funzioni di appartenenza e scoraggiare configurazioni improbabili o indesiderabili, quali ad esempio il caso di un alto grado di appartenenza ad una classe circondato da valori bassi relativi alla stessa classe.

L'algoritmo *Robust Fuzzy C-Means (RFCM)*[1] segue tale approccio e introduce un termine di penalità che tiene conto del contesto spaziale ed è consistente sia con il comportamento regolato dal parametro q sia con il comportamento desiderato in termini di smoothness.

Il vantaggio di usare una funzione di penalità sta nel fatto che solo il calcolo

che coinvolge i gradi di appartenenza viene modificato, mentre il calcolo dei centroidi rimane lo stesso del FCM standard.

L'algoritmo RFCM è caratterizzato da alcune proprietà interessanti. Possiede infatti una robustezza al rumore maggiore dello FCM standard. Inoltre la sua formulazione matematica è solo leggermente differente dalla formulazione originale dello FCM, e ciò permette una più facile implementazione nell'estensione di applicazioni che fanno uso dell'algoritmo FCM originale.

Andiamo ora a descrivere nel dettaglio la formulazione matematica e le proprietà dell'algoritmo RFCM.

2.3.1 Formulazione matematica

Per restringere la funzione di appartenenza ad essere spazialmente smooth, viene introdotta la seguente funzione obiettivo:

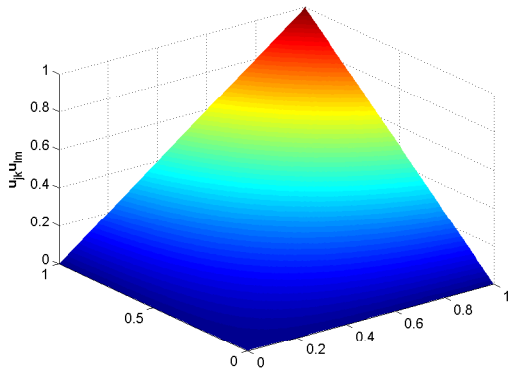
$$J_{RFCM} = \sum_{j\in\Omega}\sum_{k=1}^{C} u_{jk}^{q} \left\|\boldsymbol{y}_j - \boldsymbol{v}_k\right\|^2 + \frac{\beta}{2}\sum_{j\in\Omega}\sum_{k=1}^{C} u_{jk}^{q} \sum_{l\in N_j}\sum_{m\in M_k} u_{lm}^{q} \tag{2.2}$$

$$\sum_{k=1}^{C} u_{jk} = 1 \tag{2.3}$$

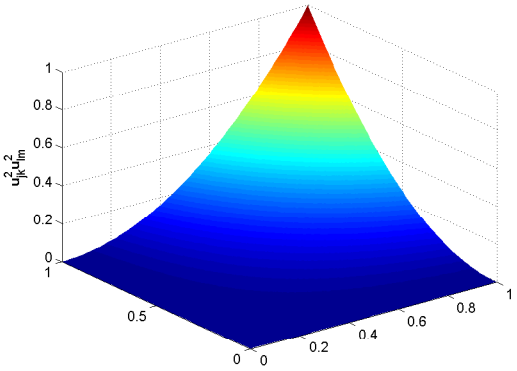
dove N_j è l'insieme dei vicini del pixel j e $M_k = \{1,\ldots,C\} - \{k\}$. Il parametro β controlla il trade-off tra minimizzare la funzione obiettivo standard dello FCM e ottenere delle funzioni di appartenenza smooth.

Il nuovo termine di penalizzazione è minimizzato quando il grado di appartenenza ad un determinato cluster è alto e i gradi di appartenenza agli altri cluster dei pixel vicini è piccolo (e viceversa). In altri termini, esso costringe i gradi di appartenenza ad un cluster a essere *correlati negativamente* con i gradi di appartenenza agli altri cluster dei pixel vicini.

La figura (2.1) mostra il ruolo della funzione di penalità per differenti valori di q. In entrambi i grafici, la penalità di una singola interazione è minimizzata quando una o entrambe le funzioni di appartenenza valgono 0, mentre è massimizzata quando entrambi valgono 1. Le configurazioni in cui pixel adiacenti hanno alti gradi di appartenenza a cluster diversi sono perciò scoraggiate. Quando $q = 2$ si ottiene una più ampia regione in cui la penalità è relativamente bassa. Questo permette l'esistenza di gradi di appartenenza con valori non necessariamente binari, ed è consistente con il ruolo di q nella formulazione FCM standard. Il parametro q regola dunque anche la "fuzziness" dell'interazione del vicinato.



(a)



(b)

Figura 2.1: Intensità della funzione di penalizzazione per differenti valori di q: (a) plot di $u_{jk}u_{lm}$, (b) plot di $u_{jk}^2u_{lm}^2$

2.3.2 Derivazione dell'algoritmo

Un algoritmo iterativo per minimizzare la (2.2) può essere derivato imponendo l'azzeramento del gradiente e calcolando i centroidi e le funzioni di appartenenza. Usando i moltiplicatori di Lagrange per rinforzare il vincolo in (2.3) e calcolando la derivata parziale di J_{RFCM} (2.2) rispetto a u_{jk} si ottiene

$$\frac{\partial \left(J_{RFCM} + \sum_{j\in\Omega} \lambda_j \left(1 - \sum_{k=1}^{C} u_{jk} \right) \right)}{\partial u_{jk}} = q u_{jk}^{q-1} \left(\left\| \boldsymbol{y}_j - \boldsymbol{v}_k \right\|^2 + \beta \sum_{l\in N_j} \sum_{m\in M_k} u_{lm}^q \right) - \lambda_j \quad (2.4)$$

dove il fattore $\frac{1}{2}$ di β scompare perché la derivata risulta composto da un termine $\frac{\beta}{2}$ corrispondente al prodotto fra u_{jk} e i suoi vicini, più un termine $\frac{\beta}{2}$ corrispondente al prodotto inverso tra i vicini e u_{jk}. Ponendo la derivata parziale a zero si ottiene:

$$u_{jk} = \left(\frac{q \left(\left\| \boldsymbol{y}_j - \boldsymbol{v}_k \right\|^2 + \beta \sum_{l\in N_j} \sum_{m\in M_k} u_{lm}^q \right)}{\lambda_j} \right)^{-\frac{1}{q-1}}. \quad (2.5)$$

Per risolvere rispetto a λ_j, sfruttiamo il vincolo (2.3) ottenendo:

$$\sum_{k=1}^{C} \left(\frac{q \left(\left\| \boldsymbol{y}_j - \boldsymbol{v}_k \right\|^2 + \beta \sum_{l\in N_j} \sum_{m\in M_k} u_{lm}^q \right)}{\lambda_j} \right)^{-\frac{1}{q-1}} = 1. \quad (2.6)$$

Poiché λ_j non dipende da k, può essere portato fuori dalla sommatoria:

$$\lambda_j^{-\frac{1}{q-1}} = q \sum_{k=1}^{C} \left(\left\| \boldsymbol{y}_j - \boldsymbol{v}_k \right\|^2 + \beta \sum_{l\in N_j} \sum_{m\in M_k} u_{lm}^q \right)^{-\frac{1}{q-1}}. \quad (2.7)$$

Combinando le equazioni (2.5) e (2.7) si ottiene la seguente condizione affinché u_{jk} si trovi in un minimo locale per J_{RFCM}:

$$u_{jk} = \frac{\left(\left\| \boldsymbol{y}_j - \boldsymbol{v}_k \right\|^2 + \beta \sum_{l\in N_j} \sum_{m\in M_k} u_{lm}^q \right)^{-\frac{1}{q-1}}}{\sum_{i=1}^{C} \left(\left\| \boldsymbol{y}_j - \boldsymbol{v}_i \right\|^2 + \beta \sum_{l\in N_j} \sum_{m\in M_k} u_{lm}^q \right)^{-\frac{1}{q-1}}}. \quad (2.8)$$

Quando $\beta = 0$ l'equazione (2.8) è identica all'equazione dello FCM standard per il calcolo dei gradi di appartenenza. Quando $\beta > 0$, la dipendenza con i vicini fa

si che u_{jk} sia elevato quando i gradi di appartenenza dei vicini agli altri cluster sono bassi. Viceversa, se i gradi di appartenenza dei vicini sono alti, il grado di appartenenza u_{jk} sarà diminuito. Il risultato è un effetto di smoothing che porta i gradi di appartenenza alla stessa classe di pixel nello stesso vicinato a essere simili uno all'altro. L'equazione (2.8) soddisfa la condizione di positività per ciascun grado di appartenenza.

Poiché la funzione di penalità non dipende da v_k, le condizioni necessarie sui centroidi affinché la (2.2) sia minimizzata sono le stesse dello FCM. Le iterazioni di queste condizioni necessarie risultano in un algoritmo per minimizzare la funzione obiettivo che è pressoché identico allo FCM standard. I passi sono i seguenti:

1. Calcolare una stima iniziale dei centroidi v_k.
2. Calcolare le funzioni di appartenenza utilizzando la (2.8).
3. Calcolare i centroidi:
$$\boldsymbol{v}_k = \frac{\sum_{j\in\Omega} u_{jk}^q \boldsymbol{y}_j}{\sum_{j\in\Omega} u_{jk}^q} \tag{2.9}$$
4. Ripartire dal passo 2 e ripetere fino alla convergenza.

La condizione per la convergenza è in genere ottenuta quando il cambiamento nella funzione obiettivo, o nelle funzioni di appartenenza, scende al di sotto di una certa soglia.

2.3.3 Correttezza dell'algoritmo

Essendo la funzione di penalizzazione indipendente dai centroidi, lo step 3 garantisce la diminuzione della funzione obiettivo come nello FCM standard. Per garantire che anche l'esecuzione del passo 2 diminuisca la funzione obiettivo, è sufficiente dimostrare che la matrice Hessiana rispetto a $u_{j1}, \ldots, u_{jC}$ sia definita positiva per ciascun pixel j. Questa matrice $C\,x\,C$ sarà diagonale, con gli elementi d_k della diagonale pari a:

$$d_k = q\,(q-1)\,u_{jk}^{q-2}\left(\left\|\boldsymbol{y}_j - \boldsymbol{v}_k\right\|^2 + \beta \sum_{l\in N_j}\sum_{m\in M_k} u_{lm}^q\right) \tag{2.10}$$

Sostituendo u_{jk} con la (2.8) e osservando che $\left\|\boldsymbol{y}_j - \boldsymbol{v}_k\right\| > 0$, segue che ciascun d_k sarà strettamente positivo quindi la matrice Hessiana sarà definita positiva. Insieme, queste due proprietà garantiscono che la funzione obiettivo J_{RFCM} decrescerà ad ogni iterazione dell'algoritmo.

2.3.4 Selezione di β

La selezione del parametro β è molto importante, in quanto influenza le performance ottenute con l'applicazione dell'algoritmo. La determinazione del valore appropriato per β dipende dall'immagine che viene sottoposta al clustering. Un valore molto alto aumenta il livello di smoothing applicato all'immagine. Questo effetto deve essere tenuto in particolare considerazione nella scelta del valore per evitare di perdere i dettagli necessari all'individuazione di piccoli noduli (circa 2mm di diametro).

Un possibile approccio per la determinazione di β consiste nell'applicare il criterio di cross-validation come suggerito in [1]. In questo approccio si estrae dall'immagine un sottoinsieme di pixel che viene utilizzato come validation-set. Dopodichè si applica iterativamente l'algoritmo RFCM sui rimanenti pixel dell'immagine fino a minimizzare l'errore sul validation-set.

Nel nostro caso si è seguito un approccio meno generale della cross-validation e più mirato ad adattare il parametro β alle caratteristiche delle immagini impiegate nel sistema di CAD polmonare[4]. Il valore di β , determinato tramite test sperimentali, è stato scelto pari a 100. Questo valore comporta un buon trade-off per ottenere una corretta classificazione delle ROI e non perdere regioni con un'area minima di 4 pixel. Tale area corrisponde a noduli candidati con un diametro di circa 2.5mm.

2.4 Implementazione

L'algoritmo RFCM è stato implementato in MATLAB, rendendone possibile l'applicazione sia per immagini bidimensionali che per immagini tridimensionali. L'impostazione utilizzata ricalca quella dello FCM standard contenuto nel FUZZY LOGIC Toolbox nella quale sono presenti: un modulo principale che inizializza le strutture dati e implementa il ciclo principale, un modulo che implementa i passi dell'algoritmo, alcuni moduli secondari che si occupano dell'effettiva inizializzazione delle singole strutture dati. L'implementazione si compone dei seguenti moduli:

- **rfcm.m.** Questo modulo implementa il ciclo principale dell'algoritmo RFCM. Il codice di questo modulo è riportato nell'appendice A.1. Si occupa delle seguenti operazioni:
  - inizializza la struttura dati contenente gli indici dei membri del vicinato di ciascun pixel chiamando la funzione *init_neigh()*. Tale

operazione è svolta una volta sola, in quanto il vicinato di ciascun pixel non cambia tra un passo e l'altro del ciclo principale.

- inizializza il vettore iniziale dei centroidi, richiamando la funzione *initfcm()* dello FCM standard essendo questa parte dell'algoritmo identica per entrambi.
- esegue i passi dell'algoritmo, richiamando la funzione *steprfcm()* e aggiornando ad ogni passo i centroidi, le funzioni di appartenenza e la funzione obiettivo. Controlla il raggiungimento della convergenza, verificando che la variazione della funzione obiettivo sia scesa al di sotto della soglia specificata.

- **steprfcm.m.** Questo modulo implementa le operazioni necessarie al calcolo iterativo dei centroidi e delle funzioni di appartenenza. Il codice di questo modulo è riportato nell'appendice A.3. Nel dettaglio, vengono compiute le seguenti attività:

  - calcola i nuovi centroidi tramite l'equazione (2.9). Nel listato A.3 si può notare come il calcolo sia effettuato come un'unica operazione di prodotto matriciale. Programmando in MATLAB si cerca di usare questo accorgimento per velocizzare il codice, evitando cicli annidati che rallentano l'esecuzione della funzione perchè devono essere interpretati ad ogni iterazione.
  - calcola le distanze tra le osservazioni e i nuovi centroidi tramite la funzione *distfcm()*. Calcola poi le penalità per ciascuna osservazione richiamando la funzione *penalty()*. Infine combina i dati ottenuti con le precedenti funzioni tramite l'equazione (2.8). Anche questo passaggio, come il calcolo dei centroidi, è implementato con un unico prodotto matriciale per velocizzare l'esecuzione.
  - calcola il valore della funzione obiettivo per il passo corrente.

- **init_neigh.m.** Questo modulo si occupa di precalcolare gli indici delle osservazioni che compongono il vicinato di ciascuna osservazione facente parte del dominio dell'immagine. Il vicinato di ciascuna osservazione è rappresentato da un array contenente l'enumerazione degli indici delle osservazioni che compongono il vicinato. Gli array contenenti ciascun vicinato vengono memorizzati in un array di celle[3], il quale viene utilizzato come argomento delle funzioni chiamate da *steprfcm()*. Il calcolo dell'array con il vicinato relativo ad un'osservazione è effettuato richiamando la

[3]Struttura dati MATLAB che permette di memorizzare in ciascun elemento una variabile di qualsiasi tipo. Il vantaggio nell'uso di questa struttura dati risiede nella mancanza di vincoli tra i tipi di ciascun elemento.

funzione *neigh()*. Il codice di questo modulo è riportato nell'appendice A.4.

- **penalty.m.** Questo modulo si occupa di calcolare il valore della penalità per ciascuna osservazione. Tale operazione non è altro che la somma dei gradi di appartenenza delle osservazioni elencate in ciascun vicinato.
  Nel listato A.5 è mostrato il codice MATLAB che corrisponde alle operazioni compiute da questo modulo. In realtà, la funzione vera e propria è stata implementata in C come mex-file[4] per rendere ragionevole il tempo di esecuzione dei due cicli annidati di cui è composta.

- **neigh.m.** Questo modulo si occupa di enumerare gli indici delle osservazioni che fanno parte del vicinato dell'osservazione, specificata come input. Il raggio del vicinato è specificato come argomento della funzione, assieme alle dimensioni dell'immagine (altezza, larghezza e profondità, se l'immagine è 3D). Il codice di questo modulo è riportato nell'appendice A.2.

2.5 Risultati

Verranno di seguito mostrati i risultati del clustering effettuato su una delle 300 "fette" di TAC spirale che compongono l'input del sistema CAD. L'immagine in scala di grigio su cui è stato eseguito il clustering è mostrata in figura 2.2.

Per mostrare le differenze derivanti dall'introduzione della penalità spaziale, sono stati eseguiti sia l'algoritmo FCM standard che l'algoritmo RFCM, entrambi con un numero di cluster pari a 4.

Per ciascuno dei cluster ottenuti sono mostrate: l'immagine con i pixel dell'immagine originale con il più alto grado di appartenenza al cluster considerato e l'immagine con i gradi di appartenenza di ciascun pixel al cluster considerato.

Per rendere effettivo il confronto, vengono affiancate le coppie <*immagine, gradi di appartenenza*> relative a cluster "simili" per ciascuno dei due algoritmi.

Come si può vedere nelle figure 2.3, 2.4, 2.5 e 2.6, l'algoritmo RFCM seleziona regioni più omogenee e smooth, mentre le regioni ottenute con lo FCM sono più frammentate.

In particolare la figura 2.5, mostra come il cluster ottenuto dallo RFCM sia relativo ai contorni dei lobi e delle strutture anatomiche presenti al loro interno,

[4] Un mex-file è un pezzo di codice scritto in un linguaggio di programmazione compilato, quali ad esempio C e FORTRAN, e opportunamente strutturato per essere richiamato da MATLAB come una funzione normale. Il mex-file viene compilato in una DLL, dopodiché viene richiamato da MATLAB nell'esecuzione degli script.

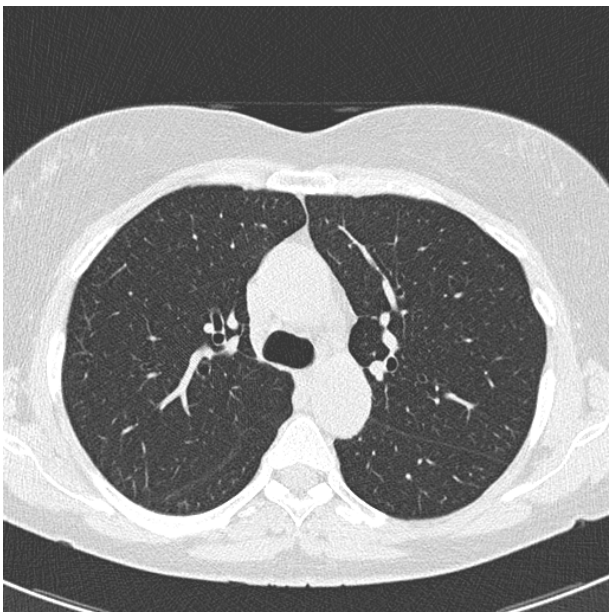
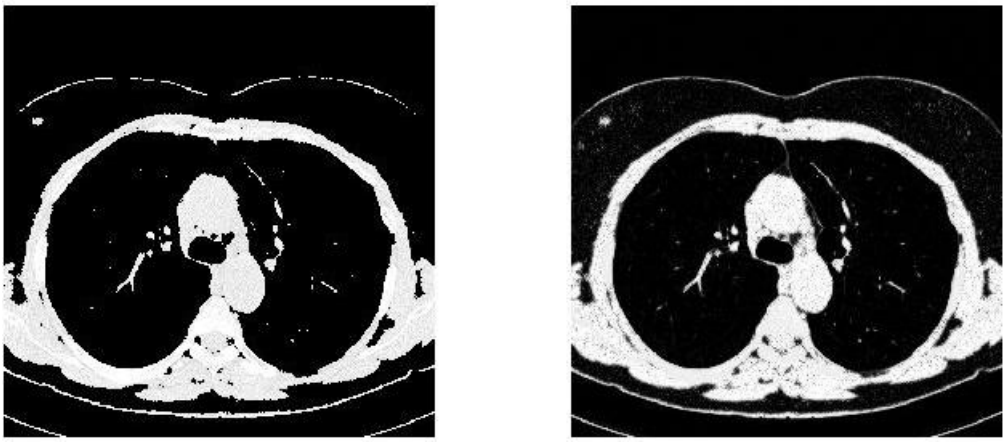


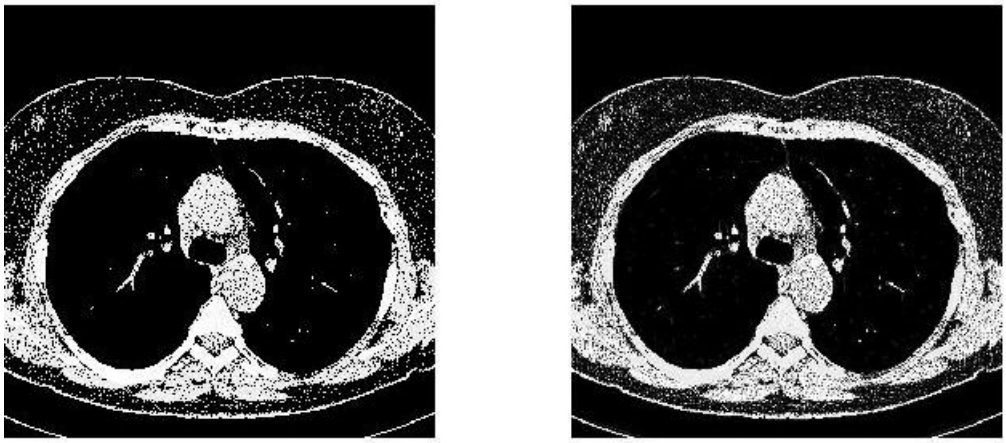
Figura 2.2: Immagine di una "fetta" di TAC spirale del polmone

mentre il cluster ottenuto con lo FCM contenga i pixel non selezionati per il cluster di figura 2.4.

Va menzionato che i risultati ottenuti in questo esempio di applicazione non verranno utilizzati per l'effettiva operazione di estrazione delle ROI. Il clustering RFCM viene infatti applicato alla zona dell'immagine contenente i soli lobi polmonari, ottenuta tramite il processo di segmentazione.

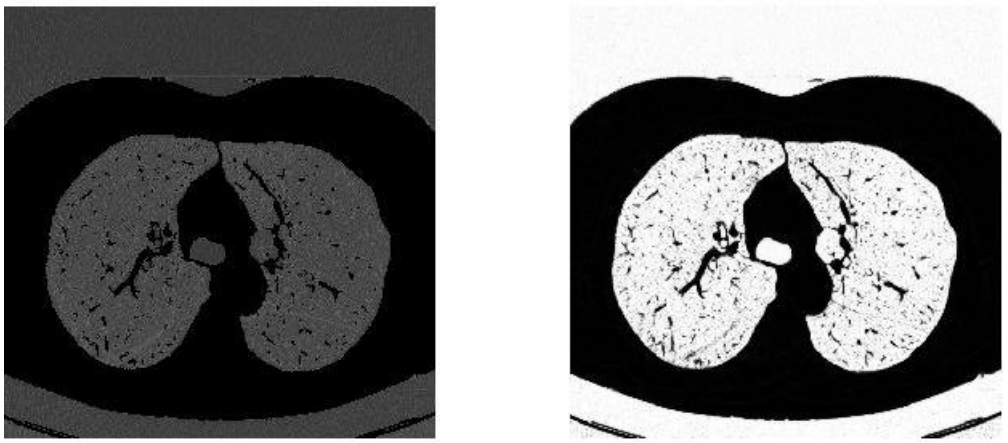


(a)

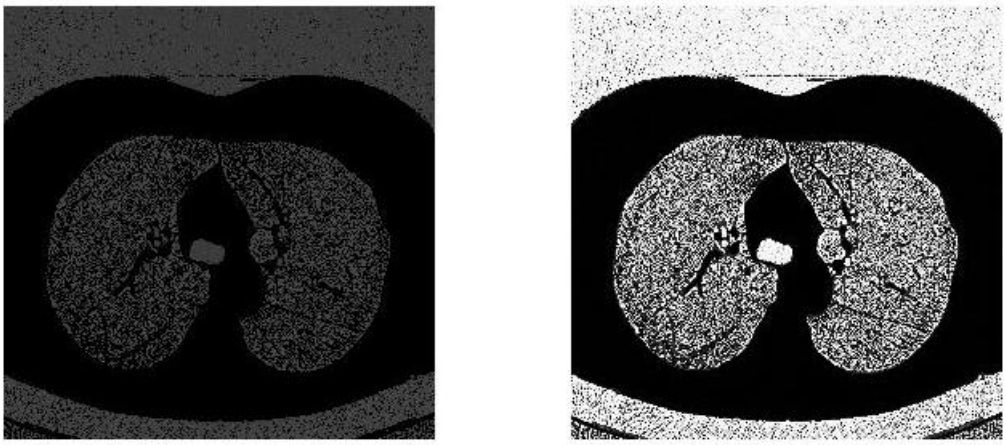


(b)

Figura 2.3: Primo cluster della gabbia toracica: (a) RFCM e (b) FCM

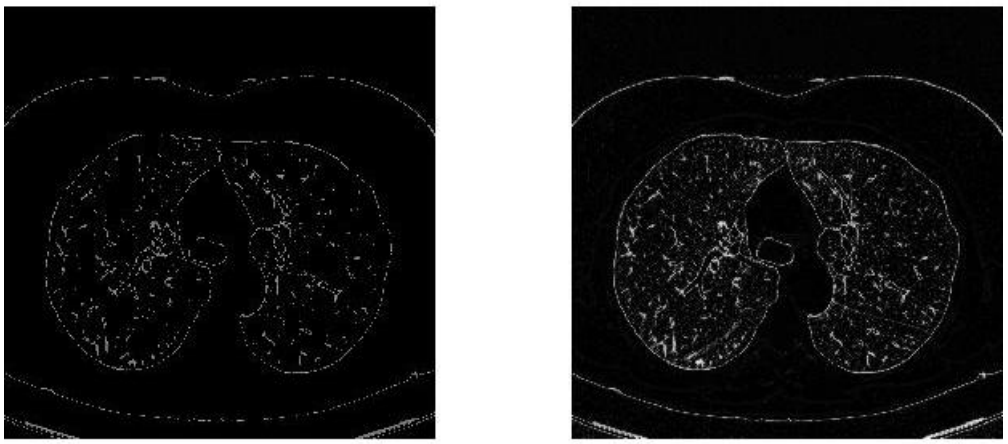


(a)

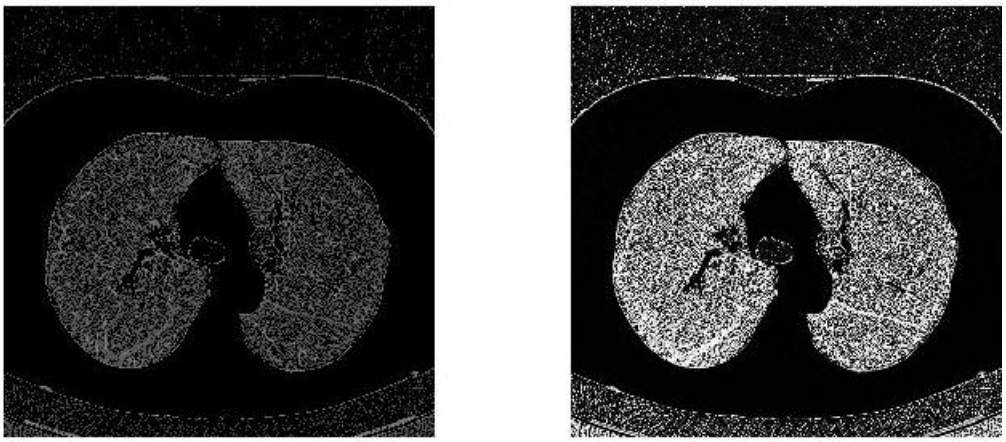


(b)

Figura 2.4: Primo cluster dei lobi polmonari: (a) RFCM e (b) FCM

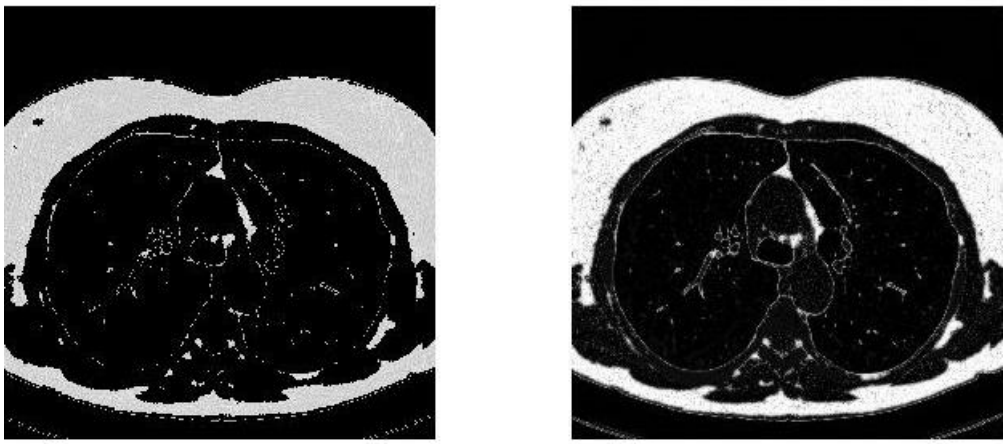


(a)

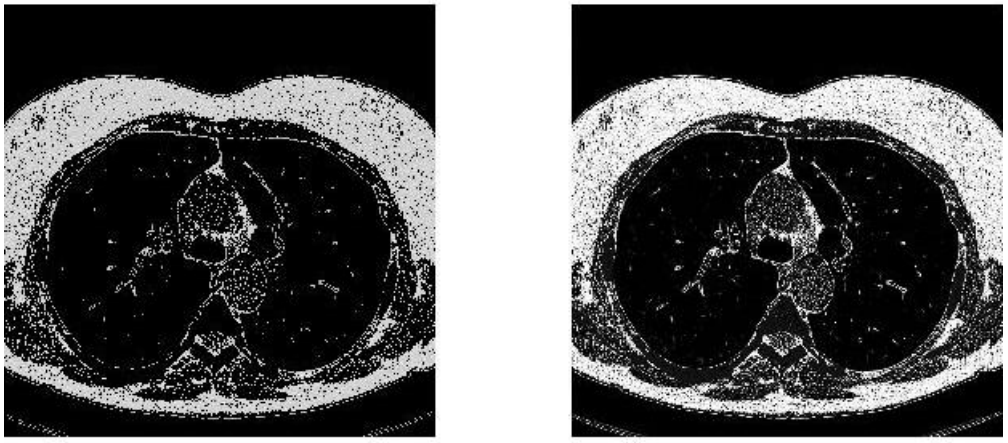


(b)

Figura 2.5: Secondo cluster dei lobi polmonari: (a) RFCM e (b) FCM



(a)



(b)

Figura 2.6: Secondo cluster della gabbia toracica: (a) RFCM e (b) FCM

Bibliografia

[1] D.L. Pham, "Spatial Models for Fuzzy Clustering", Computer Vision and Image Understanding, vol. 84, 2001, pp. 285-297.

[2] R. N. Davé, R. Krishnapuram, "Robust Clustering Methods: A Unified View", IEEE Transactions on Fuzzy Systems, vol. 5, no. 2, 1997, pp. 270-293.

[3] A. K. Jain, M. N. Murty, P. J. Flynn, "Data clustering: A review", ACM Computing Surveys, vol. 31, no. 3, 1999, pp. 265-323.

[4] M. Antonelli, G. Frosini, B. Lazzerini, F. Marcelloni, "Automated Detection of Pulmonary Nodules in CT Scans", Proc. of IEEE CIMCA-IAWTIC'05, 2005

Capitolo 3

Misure di Performance dei Classificatori

3.1 Performance dei classificatori

Andiamo ad analizzare il problema della misura della performance e della bontà di un classificatore, concentrandoci sulla *classificazione a due classi*. In questo tipo di classificazione, il problema è modellato con due classi rappresentanti, in genere, l'assenza o la presenza di una proprietà (ad esempio *benigno/maligno*), oppure due tipologie (ad esempio *nodulo/vaso sanguigno*). Gli elementi da analizzare compongono lo *spazio degli ingressi* e vengono associati ad una delle due classi secondo una o più strategie.

Formalmente, ciascuna istanza I dello spazio degli ingressi è mappata a un elemento dell'insieme $\{\boldsymbol{p}, \boldsymbol{n}\}$ delle class label positive e negative.

Un *modello di classificazione* (*classificatore*) è una funzione di mapping tra istanze e classi. Tale funzione può essere realizzata in innumerevoli modi, quali *decision tree, reti neurali, support vector machines,* etc... .

Alcuni modelli di classificazione producono un'uscita continua (ad esempio una stima della *probabilità a posteriori di appartenenza ad una classe*) al quale può essere applicata una *soglia* per predire l'appartenenza ad una classe. Altri modelli producono class label discrete indicando solamente la classe predetta dell'istanza. Per distinguere tra classi effettive e classi predette useremo le label $\{\boldsymbol{Y}, \boldsymbol{N}\}$ per la predizione prodotta da un modello.

Applicando un classificatore a un istanza, ci sono quattro possibili risultati:

1. l'istanza è positiva e viene classificata positiva: il risultato viene contato come *vero positivo (true positive)*, $\Pr(\boldsymbol{Y} \mid \boldsymbol{p})$.
2. l'istanza è positiva e viene classificata negativa: il risultato viene contato come *falso negativo (false negative)*, $\Pr(\boldsymbol{N} \mid \boldsymbol{p})$.
3. l'istanza è negativa e viene classificata positiva: il risultato viene contato come *falso positivo (false positive)*, $\Pr(\boldsymbol{Y} \mid \boldsymbol{n})$.
4. l'istanza è negativa e viene classificata negativa: il risultato viene contato come *vero negativo (true negative)*, $\Pr(\boldsymbol{N} \mid \boldsymbol{n})$.

3.1.1 Confusion Matrix

Il testing di uno schema di classificazione viene effettuato applicando lo schema ad un insieme di istanze, il *test set*, di cui si conoscono le classi. I "dati grezzi" prodotti dal test sono i conteggi, per ciascuna classe, delle classificazioni corrette e scorrette. Queste informazioni vengono poi organizzate in una matrice 2x2 detta *confusion matrix* o *tabella di contingenza*. La tabella di contingenza mostra le differenze tra classi predette e classi effettive per il test set.

	p	**n**	**total predicted**
Y	T_p	F_p	R_p
N	F_n	T_n	R_n
total	C_p	C_n	N

Tabella 3.1: Tabella di contingenza

La tabella 3.1 mostra un esempio di tabella di contingenza. Con T_p e T_n viene indicato, rispettivamente, il numero dei veri positivi e dei veri negativi[1], e con F_p e F_n il numero dei falsi positivi e dei falsi negativi. I totali di colonna, C_p e C_n, rappresentano il totale delle istanze positive e delle istanze negative, i totali di riga invece, rappresentano il totale delle istanze predette positive e predette negative. N è il numero totale di istanze. Le quantità sulla diagonale rappresentano , dunque, le decisioni corrette, mentre le quantità al di fuori della diagonale rappresentano gli errori–la confusione– tra le varie classi.

3.1.2 Misure di performance

Nonostante la confusion matrix mostri tutte le informazioni relative alla performance di un classificatore, i dati contenuti al suo interno vengono utilizza-

[1] Per chiarezza di notazione le quantità come T_p e T_n, che indicano conteggi puri, verranno indicate con le lettere maiuscole, mentre le quantità come il *true positive rate*, che indicano dei rapporti, verranno indicate con le lettere minuscole.

ti per calcolare alcune misure significative per illustrare determinati criteri di performance.

Accuracy	=	$\frac{T_p+T_n}{C_p+C_n}$	Positive predicted value	=	$\frac{T_p}{R_p}$
Sensitivity	=	$\frac{T_p}{C_p}$	Negative predicted value	=	$\frac{T_n}{R_n}$
Specificity	=	$\frac{T_n}{C_n}$	Precision	=	$\frac{T_p}{T_p+F_p}$

Tabella 3.2: Alcune metriche di performance calcolate dalla tabella di contingenza

Nella tabella 3.2 sono riportate alcune delle metriche che si possono calcolare a partire dalla tabella di contingenza. Assieme a queste, altre due sono le metriche di particolare interesse nell'analisi delle performance di un classificatore: il *true positive rate* e il *false positive rate*.

Il **true positive rate** rappresenta la percentuale di istanze positive correttamente classificate come positive. Ad esso ci si riferisce anche come *hit rate* o *recall*, e viene calcolato come:

$$tpr = \frac{\text{positivi classificati correttamente}}{\text{totale positivi}} = \frac{T_p}{C_p}$$

Il **false positive rate** rappresenta la percentuale di istanze negative incorrettamente classificate come positive. Ad esso ci si riferisce anche come *false alarm rate*, e viene calcolato come:

$$fpr = \frac{\text{negativi classificati incorrettamente}}{\text{totale negativi}} = \frac{F_p}{C_n}$$

L'importanza di queste due metriche risiede nel fatto che esse sono coinvolte nel calcolo del *misclassification cost* di uno schema di classificazione, sul quale ritorneremo in seguito nella sezione 3.3.3.

Lo fpr e il tpr, così come tutte le altre metriche di performance descritte, sono però valide per una particolare *condizione operativa*, o *operating point*. La condizione operativa è rappresentata dalla soglia di decisione in base alla quale si decide la classe predetta dal classificatore per l'istanza.

Al variare della soglia, variano anche le quantità della confusion matrix e di conseguenza le misure di performance del classificatore. Una tecnica di analisi consiste nel plottare il true positive rate in funzione del false positive rate al variare della soglia. Tale curva è chiamata *Receiver Operating Characteristic (ROC)*.

3.2 Receiver Operating Characteristic (ROC) Analysis

Il grafico della *Receiver Operating Characteristic* (ROC) è una tecnica per visualizzare, organizzare e selezionare i classificatori basata sulle loro performance. I grafici ROC vengono da lungo tempo utilizzati nella teoria del rilevamento dei segnali per rappresentare il trade-off tra lo *hit rate* e il *false alarm rate* dei classificatori. La *ROC Analysis* è stata estesa per essere usata nella visualizzazione e nell'analisi del comportamento dei sistemi di diagnosi. La comunità medica ha una estesa letteratura sull'uso dei grafici ROC per il testing diagnostico.

Oltre ad essere un utile metodo grafico di rappresentazione delle performance, i grafici ROC hanno delle proprietà che li rendono particolarmente utili per domini con class distribution asimmetriche e costi di errata classificazione diseguali. Come vedremo più avanti, queste caratteristiche sono diventate sempre più importanti man mano che continua la ricerca nelle aree dell'apprendimento cost-sensitive e dell'apprendimento in presenza di classi sbilanciate[1].

3.2.1 Lo spazio ROC

Le curve ROC sono grafici bidimensionali in cui il true positive rate *tpr* è plottato sull'asse Y e il false positive rate *fpr* è plottato sull'asse X. Un grafico ROC mostra i trade-off relativi tra i benefici (i veri positivi) e costi (falsi positivi). Nella figura 3.1 è mostrato un grafico ROC con cinque classificatori, i quali sono stati etichettati da A ad E.

Un classificatore *discreto* produce come output solo una class label. Ciascun classificatore discreto produce una coppia (fpr, tpr) che corrisponde ad un singolo punto nello spazio ROC. Tutti i classificatori in figura 3.1 sono perciò dei classificatori discreti.

Alcuni punti nello spazio ROC sono importanti da descrivere. Il punto $(0, 0)$ in basso a destra, l'origine degli assi, rappresenta la strategia che consiste nel non eseguire mai una classificazione positiva; un classificatore che adotta tale strategia non commette errori in termini di falsi positivi, ma contemporaneamente non colleziona nessun vero positivo. La strategia opposta, consistente nel classificare incondizionatamente tutte le istanze come positive, è rappresentata dal punto $(1, 1)$ nell'angolo superiore destro. Un classificatore che adotta tale strategia non commette errori in termini di falsi negativi, ma contemporaneamente colleziona il massimo numero di falsi positivi.

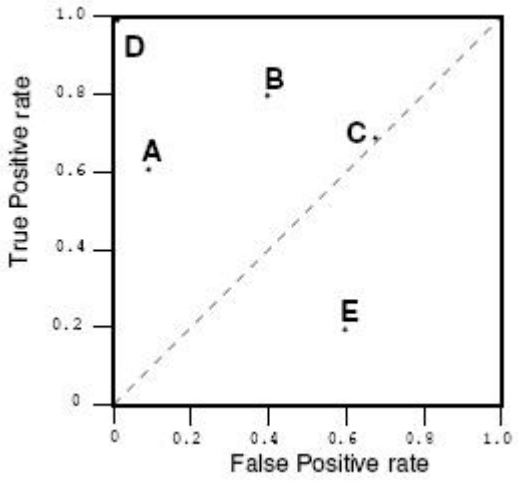


Figura 3.1: Grafico ROC elementare che mostra 5 classificatori discreti.

Il punto $(0, 1)$ rappresenta la *classificazione perfetta*. Il classificatore D ha perciò una performance ottimale.

Informalmente, un punto nello spazio ROC è migliore di un altro se si trova in alto a sinistra rispetto a quest'ultimo. Ciò significa che il *tpr* del primo è più alto, lo *fpr* è più basso, o entrambe.

I classificatori che appaiono nella parte sinistra del grafico ROC, vicino all'asse X, possono essere pensati come "conservativi": la classificazione positiva è effettuata solo se è presente una forte evidenza, perciò l'errore in termini di falsi positivi è basso ma spesso è basso anche il true positive rate.

I classificatori nella parte superiore destra del grafico ROC possono invece essere pensati come "liberali": la classificazione positiva è effettuata con una bassa evidenza cosicché praticamente tutte le istanze positive vengono classificate correttamente. Spesso, però, questo porta ad avere un alto false positive rate.

Molti domini reali di applicazione, come ad esempio la rilevazione di noduli polmonari, sono dominati da un numero molto elevato di istanze negative, perciò le performance nella parte sinistra del grafico ROC sono molto più interessanti.

3.2.2 Classificazione casuale

La linea diagonale $y = x$ rappresenta la strategia di scelta casuale della classe. Ad esempio, se un classificatore sceglie casualmente la classe positiva metà delle volte, ci si aspetta che ottenga correttamente metà dei positivi e metà dei negativi, venendo perciò rappresentato dal punto (0.5,0.5) nello spazio ROC.

Se invece il classificatore sceglie la classe positiva il 90% delle volte, ci si aspetta che ottenga correttamente il 90% dei positivi; contestualmente anche il false positive rate salirà al 90%, facendo si che il classificatore sia rappresentato dal punto $(0.9, 0.9)$ nello spazio ROC.

Perciò un classificatore casuale produce un punto ROC che "si muove" lungo la diagonale in funzione della frequenza con cui il classificatore sceglie la classe positiva.

Qualsiasi classificatore che appaia al di sotto della diagonale ha una performance peggiore di quella del classificatore casuale. Quindi questa zona dei grafici ROC è di solito vuota. Va comunque notato che lo spazio delle decisioni è simmetrico rispetto alla diagonale. Se neghiamo un classificatore, ossia invertiamo le sue classificazioni per ciascuna istanza, i suoi veri positivi diventeranno falsi negativi, e i suoi falsi positivi diventeranno veri negativi. Perciò, ciascun classificatore che produce un punto al di sotto della diagonale può essere negato per produrre un punto al di sopra della diagonale, nello specifico tale punto sarà il simmetrico del precedente rispetto alla diagonale.

3.2.3 Curve ROC

I classificatori continui producono uno *score* per ciascuna istanza, ovvero un valore numerico che rappresenta il grado di appartenenza dell'istanza ad una determinata classe. Questi valori possono essere delle probabilità, in particolare *probabilità a posteriori*, nel qual caso rispettano i teoremi della probabilità; oppure possono essere degli score generici e non calibrati, nel qual caso l'unica proprietà che vale è che uno score più alto rappresenta una probabilità più alta.

Questi classificatori continui (probabilistici o di *ranking/scoring)* vengono usati insieme ad una soglia per produrre una classificazione discreta[2]: se l'output del classificatore è al di sopra della soglia, il classificatore produce $\boldsymbol{Y}$ come class label, altrimenti produce $\boldsymbol{N}$. Ogni valore della soglia produce un diverso punto nello spazio ROC. Variando la soglia da $-\infty$ a $+\infty$ si traccia la curva ROC associata ad un classificatore. Quando la soglia è $-\infty$, la curva ROC si trova nel punto $(1, 1)$. A mano a mano che la soglia aumenta, il punto corrispondente sulla curva ROC si sposta sempre più in basso e verso sinistra, fino ad arrivare al punto $(0, 0)$ corrispondente ad una soglia pari a $+\infty$. Aumentare la soglia equivale a portarsi da una condizione operativa "liberale" ad una condizione

[2]Le uscite del classificatore probabilistico, per poter essere utilizzate correttamente con la soglia, vanno combinate in un'unica uscita continua. Tale uscita è espressa come $y(\mathbf{x}) = -1 \cdot \hat{\Pr}(\boldsymbol{N} \mid \mathbf{x}) + 1 \cdot \hat{\Pr}(\boldsymbol{Y} \mid \mathbf{x})$, ossia come combinazione convessa di -1 e 1 in base alle probabilità a posteriori $\hat{\Pr}(\boldsymbol{N} \mid \mathbf{x})$ e $\hat{\Pr}(\boldsymbol{Y} \mid \mathbf{x})$, che rappresentano le uscite del classificatore.

operativa più "conservativa". La figura 3.2 mostra l'effetto della variazione della soglia sul punto corrispondente della curva ROC.

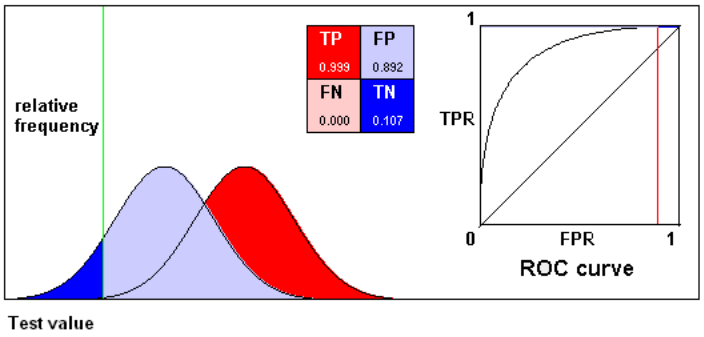
Nella figura 3.3 sono mostrate le curve ROC di nove classificatori di varia natura, relativi all'applicazione degli stessi su due database: phoneme data del database ELENA[16], e pima indians diabetes del UCI ML Repository[15].

Per quanto riguarda il database dei fonemi, il classificatore con le migliori performance in tutte le aree dello spazio ROC è risultato essere il *K-Nearest Neighbour*. Nel caso del database pima, invece, osserviamo come nella parte "conservativa" dello spazio ROC prevalgano il classificatore *Logistico*, il classificatore *Probabilistico Lineare* e il classificatore di *Fisher*, mentre nella parte "liberale" dello spazio ROC prevalga il classificatore *Naive Bayes*.

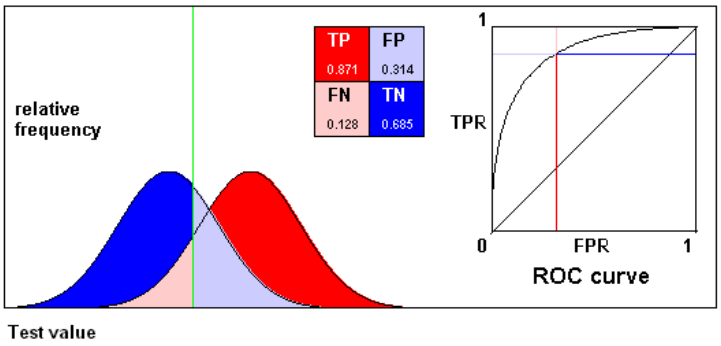
3.2.4 Proprietà delle curve ROC

Le curve ROC hanno una proprietà molto attraente: sono *insensibili* ai cambiamenti nella distribuzione delle classi. Se la proporzione tra le istanze positive e le istanze negative nel test set cambia, le curve ROC rimarranno immutate. Per capire come ciò accada, consideriamo la tabella 3.1. Va notato che la distribuzione delle classi, cioè la proporzione tra istanze positive e negative, è la relazione tra la colonna sinistra (istanze positive) e la colonna destra (istanze negative). Qualsiasi metrica di performance che usi valori da entrambe le colonne sarà inerentemente sensibile allo sbilanciamento tra le classi, detto anche *class skew*. Le quantità plottate nella curva ROC, invece, sono calcolate usando per ciascuna una singola colonna della tabella. Nello specifico, il *true positive rate* è calcolato utilizzando il conto dei veri positivi (T_p) e il totale delle istanze positive (C_p) ed entrambi questi valori appartengono alla stesa colonna. Lo stesso discorso vale per il *false positive rate*, il quale è calcolato utilizzando il conto dei falsi positivi (F_p) e il totale delle istanze negative (C_n). Se ad esempio cambiasse il totale C_p delle istanze positive, cambierebbe anche il conto $T_p$dei veri positivi, ma il loro rapporto rimarrebbe immutato, in quanto esso dipende esclusivamente dalla strategia di classificazione, e di conseguenza anche il *true positive rate* e la curva ROC non subirebbero alterazioni. Un discorso analogo può essere fatto per il *false positive rate*.

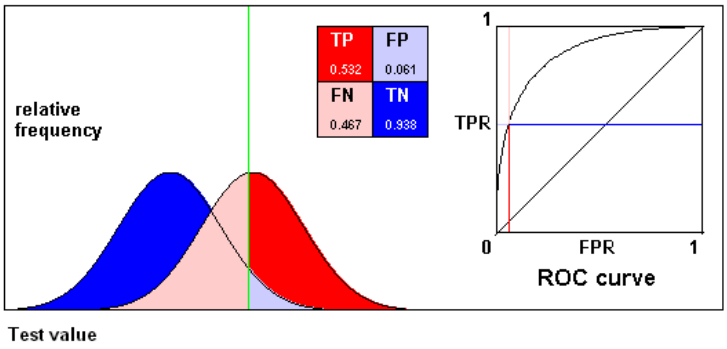
Tale proprietà è la base del successo della ROC Analysis, in quanto nei domini reali di applicazione sono molto comuni class skew dell'ordine di 10^1 e 10^2, fino ad arrivare a 10^6 per alcune applicazioni particolari. Non è inoltre irrealistico osservare un cambiamento sostanziale nella distribuzione delle classi. Ad esempio, nel campo del decision making medico, le epidemie possono far si che l'incidenza di una malattia aumenti nel tempo.



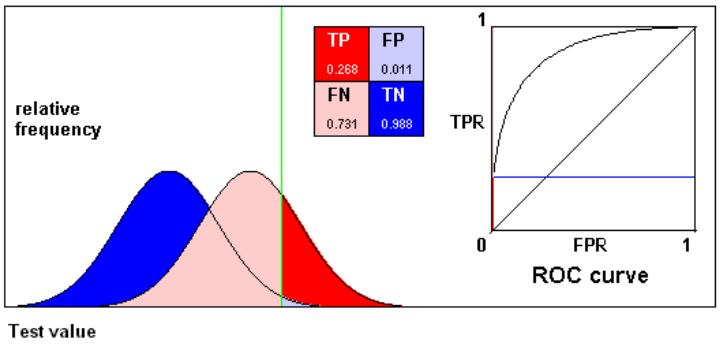
(a)



(b)

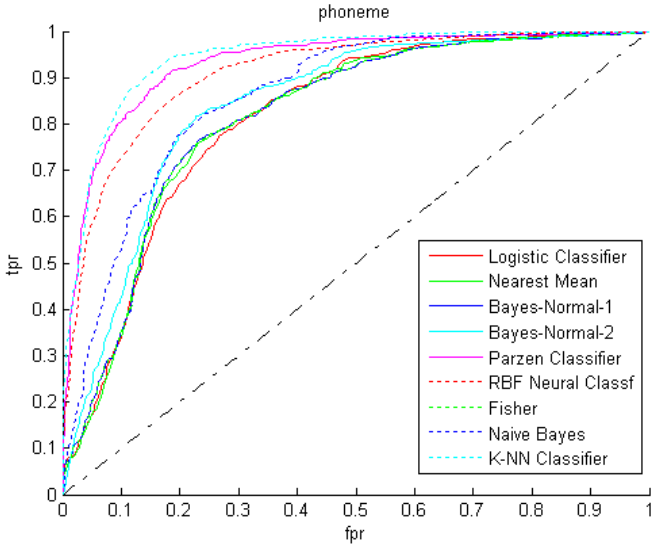


(c)

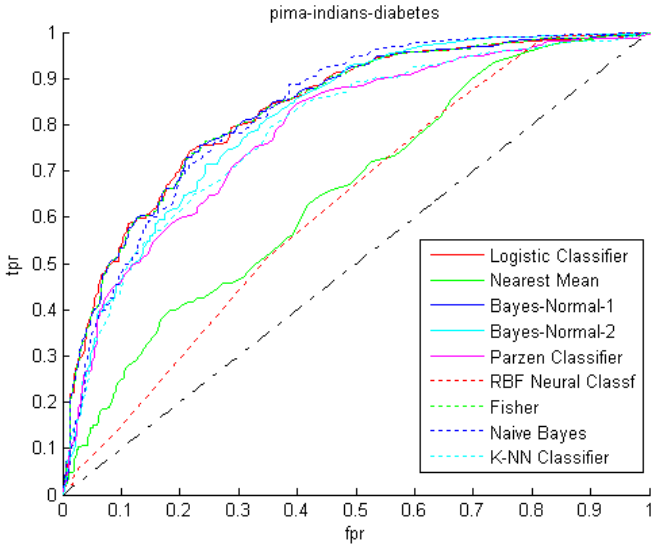


(d)

Figura 3.2: Andamento del punto sulla curva ROC al variare della soglia



(a)



(b)

Figura 3.3: Grafici ROC di alcuni classificatori applicati a due database: (a) phoneme data e (b) pima indians diabetes.

3.3 Area Under the ROC Curve (AUC)

La curva ROC è una rappresentazione *bidimensionale* delle performance di un classificatore e permette un efficace confronto visuale di un insieme di classificatori. Nonostante sia una validissima tecnica di visualizzazione, l'uso del grafico ROC è di scarso aiuto nella scelta dei classificatori. Essendo una tecnica bidimensionale, il confronto tra due o più classificatori può risultare ambiguo, e solo quando si riscontra una dominanza sull'intero spazio ROC un classificatore può essere dichiarato migliore.

C'è bisogno, perciò, di ricavare una metrica *monodimensionale* che riassuma le caratteristiche contenute nel grafico ROC e le riduca ad un singolo valore rappresentante la performance attesa. Tale metrica deve essere invariante nei confronti del criterio decisionale selezionato, delle probabilità a priori per ciascuna classe, e deve poter essere facilmente estesa per incorporare l'analisi costi/benefici. Va però sottolineato che un qualsiasi tentativo di riassumere la curva ROC in un unico valore porta alla perdita di informazioni relative al pattern dei trade-off relativi a ciascun modello di classificazione.

Tra le numerose statistiche riassuntive ideate a questo scopo, una delle più usate è senza dubbio l'*Area Under the ROC Curve* o AUC, ovvero l'area sottesa alla curva ROC.

3.3.1 Proprietà dell'AUC

Come evidenziato in [3], l'AUC gode di un'interessante proprietà statistica: l'AUC di un classificatore equivale alla probabilità che il rank prodotto dal classificatore prodotto per un'istanza positiva scelta casualmente sia maggiore del rank prodotto per un'istanza negativa scelta casualmente. Tale proprietà rende l'AUC equivalente alla statistica di Wilcoxon-Mann-Whitney per i rank. Formalmente avremo:

$$AUC = \Pr(x_p > y_n) = W$$

$$W = \frac{1}{C_p C_n} \sum_{j=1}^{C_n} \sum_{i=1}^{C_p} I(x_i, y_j)$$

$$I(x, y) = \begin{cases} 1 & x > y \\ 0 & x < y \\ \frac{1}{2} & x = y (\text{solo per valori discreti}) \end{cases} \tag{3.1}$$

Classificatore	AUC (trapz)	AUC (Wilcox)
Logistic	0.8121	0.8121
Nearest Mean	0.8139	0.8138
LDC	0.8158	0.8159
QDC	0.8409	0.8441
Parzen	0.9314	0.9330
RBF	0.9103	0.9108
Fisher	0.8156	0.8159
Naive Bayes	0.8595	0.8603
K-NN	0.9437	0.9438

Tabella 3.3: Tabella delle AUC relative al Phoneme dataset

dove x_p è il rank di un'istanza positiva, y_n è il rank di un'istanza negativa, C_p e C_n sono i totali di tabella 3.1, e W è la statistica di Wilcoxon-Mann-Whitney. Nella sezione 3.4.5 riporteremo una possibile dimostrazione di tale proprietà.

L'AUC può essere facilmente calcolata tramite integrazione numerica con il metodo dei trapezi. Come è stato osservato in [3], tale metodo porta però ad una sottostima sistematica dell'area della curva. Un metodo alternativo consiste nello sfruttare l'equivalenza (3.1), e calcolare l'AUC tramite l'indicatore della statistica di Wilcoxon-Mann-Whitney.

Poiché l'AUC è una porzione del quadrato unitario, il suo valore sarà sempre compreso tra 0 e 1. Va notato però che, poiché al classificatore casuale è associata la diagonale da $(0,0)$ a $(1,1)$, la quale ha un'area pari a $0,5$, nessun classificatore realistico dovrebbe avere un' AUC inferiore a $0,5$, perché le sue performance sarebbero inferiori a quelle del classificatore casuale.

L'AUC è inoltre strettamente legata al coefficiente di GINI[3] secondo la relazione $GINI + 1 = 2 \times AUC$.

3.3.2 Alcune osservazioni

Nella tabelle 3.3 e 3.4 sono riportate, rispettivamente, le misure dell'AUC relative ai classificatori applicati al Phoneme dataset e al Pima Indians Diabetes dataset.

Per quanto riguarda il Phoneme dataset, avevamo osservato nel grafico ROC di figura 3.3 (a) una chiara predominanza del classificatore K-NN che è stata

[3] Il coefficiente di GINI viene usato nella *statistica univariata* come misura della *eterogeneità di una distribuzione*. Esso è definito come il rapporto tra l'area compresa tra la curva di Lorenz e la curva della distribuzione uniforme e l'area sottesa alla curva della distribuzione uniforme. La curva di Lorenz è il grafico delle funzioni di ripartizione di due distribuzioni. L'AUC è una curva di Lorenz per le distribuzioni dei falsi positivi e dei veri positivi.

Classificatore	AUC (trapz)	AUC (Wilcox)
Logistic	0.8326	0.8329
Nearest Mean	0.6539	0.6562
LDC	0.8323	0.8330
QDC	0.8140	0.8131
Parzen	0.7867	0.7917
RBF	0.6260	0.6812
Fisher	0.8317	0.8331
Naive Bayes	0.8289	0.8272
K-NN	0.7889	0.7907

Tabella 3.4: Tabella delle AUC relative al Pima dataset

confermata da un valore AUC di 94.37%, vedi tabella 3.3, superiore a quello ottenuto per gli altri classificatori.

Per quanto riguarda il Pima dataset, invece, il grafico ROC di figura 3.3 (b) non ci permetteva di selezionare un classificatore come migliore in termini di performance. Tramite l'AUC, troviamo conferma della relativa superiorità dei classificatori Logistico, LDC e Fisher e riusciamo a selezionare il classificatore Fisher come migliore in termini di performance complessive.

3.3.3 Performance Isometrics

Per comprendere meglio la geometria dello spazio ROC e fornire uno strumento per la selezione della configurazione operativa, o operating point, del classificatore selezionato, andiamo ad introdurre le *isoperformance lines*.

Queste rappresentano, nello spazio ROC, le curve isometriche relative alle misure di performance che abbiamo in parte accennato nella sezione 3.1.2.

Una volta che si è selezionato un modello di classificazione attraverso la valutazione dell'AUC, il passo successivo consiste nel selezionare l'operating point, e quindi la relativa soglia decisionale, che minimizza la metrica o le metriche di performance richieste. Per fare ciò si individua l'intersezione tra la curva ROC del classificatore e la *linea di isoperformance* relativa alla metrica di interesse.

Passiamo ora in rassegna le principali metriche, e le relative isoperformance lines, utilizzate nella valutazione dei classificatori.

In [1] viene introdotto il *misclassification cost*, ovvero il costo cui si deve far fronte nel caso di un errore di classificazione. Un errore di classificazione consiste nel classificare come negativa un'istanza positiva, generare cioè un *falso negativo*, oppure nel classificare come positiva un'istanza negativa, generare cioè un *falso positivo*. Ciascuno di questi tipi di errore avrà un costo, in generale diverso a

seconda del tipo di errore. Formalmente il *misclassification cost* è perciò definito come:

$$\begin{aligned} Cost &= \Pr(\boldsymbol{p})\, c(\boldsymbol{N}, \boldsymbol{p}) \Pr(\boldsymbol{N} \mid \boldsymbol{p}) + \Pr(\boldsymbol{n})\, c(\boldsymbol{Y}, \boldsymbol{n}) \Pr(\boldsymbol{Y} \mid \boldsymbol{n}) \\ &= \Pr(\boldsymbol{p})\, c(\boldsymbol{N}, \boldsymbol{p})\,(1 - tpr) + \Pr(\boldsymbol{n})\, c(\boldsymbol{Y}, \boldsymbol{n})\, fpr \end{aligned} \tag{3.2}$$

dove $c(\boldsymbol{N}, \boldsymbol{p})$ è il costo di un falso negativo, $c(\boldsymbol{Y}, \boldsymbol{n})$ è il costo di un falso positivo, $\Pr(\boldsymbol{p})$ è la probabilità a priori di avere una istanza positiva, $\Pr(\boldsymbol{n})$ è la probabilità a priori di avere una istanza negativa e $\Pr(\boldsymbol{N} \mid \boldsymbol{p})$ e $\Pr(\boldsymbol{Y} \mid \boldsymbol{n})$ sono, rispettivamente, le probabilità di avere un falso negativo e di avere un falso positivo.

A seconda del dominio di applicazione del problema, varieranno i costi relativi ai falsi positivi e negativi e le probabilità a priori per le due classi, e di conseguenza la geometria delle curve di *misclassification cost*.

In [10] vengono invece introdotte una serie di metriche che dipendono dal class skew, ovvero dal rapporto tra le classi del problema. Alcune di queste riprendono le quantità calcolate a partire dalla confusion matrix, elencate in tabella 3.2, le altre invece sono la controparte nello spazio ROC di differenti quantità calcolate sempre a partire dai dati della confusion matrix.

Definiamo per prima cosa il class skew c come:

$$c = \frac{C_n}{C_p} \tag{3.3}$$

La prima metrica è l'*accuracy*, che corrisponde alla misura di accuratezza, ed è definita come:

$$Accuracy = \frac{tpr + c\,(1 - fpr)}{1 + c} \tag{3.4}$$

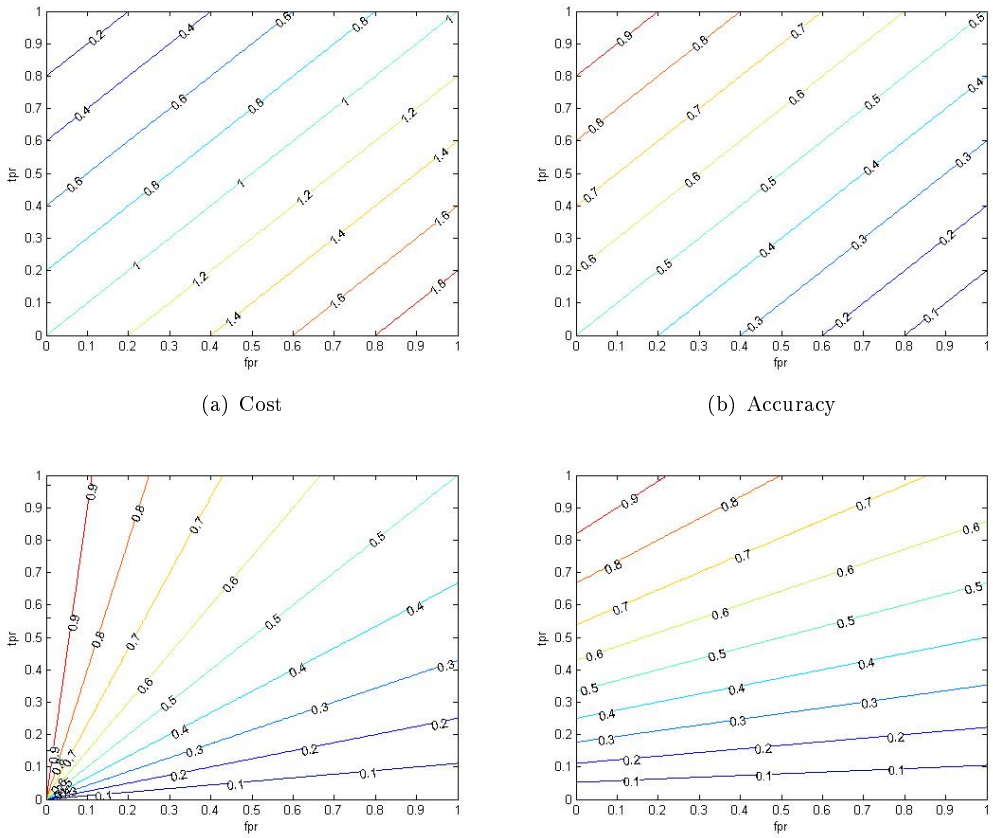
L'altra metrica è la *precision*, corrispondente alla precisione, e definita come:

$$Precision = \frac{tpr}{tpr + c \cdot fpr} \tag{3.5}$$

Infine introduciamo la *F-Measure*, che corrisponde alla misura ottenuta mediando la precisione e il richiamo, ed è definita come:

$$F\_measure = \frac{2tpr}{tpr + c \cdot fpr + 1} \tag{3.6}$$

Nella figura 3.4 sono mostrate, dalla (a) alla (d), le precedenti misure nell'ordine in cui sono state descritte.



(a) Cost

(b) Accuracy

(c) Precision

(d) F-Measure

Figura 3.4: Curve di isoperformance relative alle metriche specificate

Descriviamo ora in maggiori dettagli l'effetto sulle curve di livello di ciascuna metrica della variazione dei parametri.

Per quanto riguarda il *misclassification cost*, una variazione dei costi degli errori e delle probabilità a priori, porta ad una variazione della pendenza delle rette rappresentanti le curve di livello. Nello specifico, una prevalenza del costo dell'errore in termini di falsi positivi e/o uno sbilanciamento verso le istanze negative in termini di probabilità a priori, porta ad avere rette con pendenza elevata poichè il prodotto di tali termini è al numeratore nelle equazioni delle curve di livello. Viceversa, una prevalenza del costo dell'errore in termini di falsi negativi e/o uno sbilanciamento verso le istanze positive in termini di probabilità a priori, porta ad avere rette con pendenza bassa poichè il prodotto di tali termini si trova al denominatore.

Lo stesso discorso vale per la *accuracy*, dove un basso valore del class skew, corrispondente ad una prevalenza del numero di istanze positive nei confronti del numero di istanze negative, porta ad avere delle rette rappresentanti le curve di livello con pendenza bassa. Viceversa, un alto valore del class skew, corrispondente ad una prevalenza del numero di istanze negative nei confronti del numero di istanze positive, corrisponde a rette con pendenza elevata.

Nel caso della *precision*, la variazione del class skew corrisponde ad una rotazione delle rette centrata nell'origine.

Differente è l'effetto del class skew sulle curve di livello della *F-Measure*. Queste infatti sono tutte originate in un unico punto, le cui coordinate dipendono dal class skew e sono pari a:

$$(fpr = -\frac{1}{c},\ tpr = 0)$$

Un aumento del class skew porta dunque l'origine del fascio di rette ad avvicinarsi all'origine degli assi. Se lo sbilanciamento verso le istanze negative è molto marcato, e quindi il class skew è molto alto, le curve della f-measure tenderanno a coincidere con le curve dell *precision*.

3.4 Una nuova misura di performance

3.4.1 Il criterio di Neyman-Pearson e i limiti dell'AUC

L'AUC, come d'altronde tutte le statistiche *single-valued* riassuntive costruite sulla curva ROC, presenta delle limitazioni. Per capire meglio quali sono, andiamo ad analizzare il *criterio di Neyman-Pearson* per la selezione dei classificatori.

La discriminazione tra classificatori in base al criterio di Neyman-Pearson avviene nel seguente modo:

1. Si stabilisce il massimo *false positive rate,* fpr_{max}, tollerabile dal sistema per l'applicazione considerata.
2. Si seleziona il classificatore che, a parità di fpr_{max}, presenta il *true positive rate* più alto.

In termini di curve di isoperformance, il criterio di Neyman-Pearson equivale a linee verticali perpendicolari all'asse x, corrispondenti ai valori del *false positive rate*.

In molti reali casi applicativi, si è interessati a minimizzare il più possibile il numero dei falsi positivi ottenuti dal sistema, ovvero si va a preferire una strategia "conservativa". Di conseguenza l'attenzione è concentrata sulle performance dei classificatori nella zona dello spazio ROC vicina all'origine, cioè l'area in cui i valori di *fpr* sono vicini allo 0.

L'uso dell'AUC come metrica di valutazione comporta, alla luce di tale considerazione, una perdita della "località" delle informazioni relative alle performance. L'AUC è infatti calcolata integrando i vari valori del *true positive rate* lungo tutto l'asse x, ovvero per tutti i valori del *false positive rate* senza discriminazione.

L'ideale sarebbe ottenere una statistica riassuntiva globale, come l'AUC, ma che tenga in maggiore considerazione le performance dei classificatori per bassi valori del *false positive rate*.

3.4.2 Le possibili estensioni dell'AUC

L'idea che abbiamo seguito è stata quella di estendere l'AUC tenendo presenti le precedenti considerazioni sulle performance per bassi *fpr*. A tale scopo abbiamo focalizzato la nostra attenzione sugli indici di performance utilizzati nella *Teoria dei Controlli Automatici* per la progettazione di controllori ottimi.

In tale campo, i parametri dei controllori vengono selezionati in modo da minimizzare l'errore compiuto dal sistema alla risposta a ingressi standard, quali il gradino unitario.

L'indicatore più semplice è l'*IAE*, Integral of Absolute Error, ovvero l'integrale del modulo dell'errore, calcolato come:

$$IAE = \int |e(t)|\, dt$$

Poiché il progettista è in genere interessato maggiormente a minimizzare l'errore a regime rispetto all'errore commesso nel transitorio, sono stati ideati altri indici che pesano in maniera differente l'errore, a seconda dell'istante temporale in cui esso viene compiuto. Tra questi riportiamo l'$ITAE$ e l'IT^2AE, definiti come segue:

$$ITAE = \int t\,|e(t)|\,dt$$

$$IT^2AE = \int t\,|e(t)|\,dt$$

L'ITAE va a pesare l'errore commesso dal sistema moltiplicandolo per l'istante temporale, discriminando in questo modo l'errore commesso al transitorio dall'errore commesso a regime, privilegiando la riduzione dell'errore a regime. L'IT^2AE segue la stessa strategia ma, pesando l'errore con il quadrato del tempo, da un peso molto minore, rispetto all'ITAE, all'errore commesso al transitorio e un peso molto maggiore all'errore commesso a regime.

Prendendo spunto da queste metodologie della teoria del controllo, si possono introdurre degli equivalenti, nello spazio ROC, per gli indici precedentemente menzionati.

A tale scopo vanno però fatte delle considerazioni preliminari, relative alle caratteristiche dello spazio ROC e agli obiettivi della valutazione delle performance dei classificatori.

1. Mentre nella progettazione dei controllori l'obiettivo è minimizzare l'indice scelto, in quanto esso misura l'errore, nella valutazione delle performance l'obiettivo è massimizzare l'indice scelto, in quanto esso misura positivamente la bontà del classificatore.

2. Ancora più importante è la differente "zona" che si va ad evidenziare. Nella progettazione dei controllori si da maggior peso all'errore compiuto a regime, ovvero per istanti temporali lontani dallo 0. Nella scelta dei classificatori, invece, siamo interessati ad evidenziare le performance del classificatore nella zona vicina all'origine, che rappresenta le performance del classificatore per bassi valori del *false positive rate*.

3. Infine si deve tener conto delle caratteristiche "geometriche" dello spazio ROC. Esso infatti è localizzato nel quadrato unitario, perciò va fatta particolare attenzione nella scelta dei "pesi", al fine di evitare una situazione in cui si vadano a penalizzare le regioni che si era intenzionati a privilegiare.

Tenute presenti le precedenti considerazioni, andiamo a mostrare gli equivalenti

ROC degli indici della controllistica:

$$\begin{array}{ccc} IAE & \longleftrightarrow & AUC \\ ITAE & \longleftrightarrow & tAUC = \int tpr\,(1-fpr)\,dfpr \\ IT^2AE & \longleftrightarrow & t^2AUC = \int tpr\sqrt{fpr}dfpr \end{array} \tag{3.7}$$

Tali equivalenti sono stati ottenuti sostituendo al valore assoluto dell'errore, $|e\,(t)|$, il true positive rate, tpr, e modificando opportunamente i fattori di scala con i corrispettivi nello spazio ROC più adeguati alla luce delle tre considerazioni.

E' interessante osservare la corrispondenza, praticamente immediata, tra l'ITAE e l'AUC, e come essi presentino gli stessi inconvenienti in termini di perdita di località.

Alla luce della 2 e della 3, andiamo a chiarire in maggior dettaglio la scelta dei "pesi" per gli indici $tAUC$ e t^2AUC.

Per il $tAUC$ si è scelto come equivalente di t il termine $(1-fpr)$: esso è infatti lineare ed è maggiore per bassi valori di fpr e minore per alti valori di fpr, essendo fpr compreso tra 0 e 1.

Per il t^2AUC si è scelto come equivalente di t^2 il termine $\sqrt{fpr}$: poiché fpr è un numero sempre minore o uguale a 1, non si poteva utilizzare il termine fpr^2 in quanto si sarebbe penalizzato ciò che si voleva esaltare; si è perciò scelto di utilizzare $\sqrt{fpr}$ poiché la radice quadrata di un numero minore di 1 è maggiore del numero stesso.

Nonostante il t^2AUC dia maggiore risalto alla parte dello spazio ROC di nostro interesse, si è scelto di adottare il $tAUC$ per le sue interessanti proprietà.

3.4.3 Paradigmi sperimentali e significato del tAUC

Andiamo ora ad illustrare quelle che sono le caratteristiche e il significato di questa nuova metrica.Per prima cosa focalizziamo l'attenzione sull'espressione del $tAUC$:

$$tAUC = \int tpr\,(1-fpr)\,dfpr \tag{3.8}$$

Possiamo innanzitutto osservare che il termine $1-fpr$, dalla definizione di *false positive rate* e delle metriche di tabella 3.1 e 3.2, corrisponde al *true negative rate*, ovvero alla percentuale di istanze negative correttamente riconosciute. Pesando il *true positive rate* con il *true negative rate*, abbiamo in effetti calcolato la

percentuale di istanze negative e positive correttamente riconosciute se presentate contemporaneamente. Come verrà dimostrato nella sezione 3.4.5, il $tAUC$ è uguale alla probabilità che gli score di una coppia di istanze, una positiva e l'altra negativa scelte casualmente, siano correttamente ordinati rispetto ad una terza istanza negativa, anche essa scelta casualmente. Formalmente:

$$tAUC = \Pr(x_i > y_k \geq y_j) \tag{3.9}$$

dove x_i e y_j sono gli score della coppia di istanze, positiva e negativa rispettivamente, e y_k è lo score dell'istanza negativa di riferimento.

Se inoltre sviluppiamo la (3.8) notiamo un'interessante relazione:

$$\begin{aligned} tAUC &= \int tpr \cdot dfpr - \int tpr \cdot fpr \cdot dfpr \\ &= AUC - \int tpr \cdot fpr \cdot dfpr \end{aligned} \tag{3.10}$$

La (3.10) ci mostra come il $tAUC$ sia legato linearmente all'AUC, e nel dettaglio come questo legame risulti nella correzione dell'AUC rispetto ad un termine che assume un particolare significato. Il prodotto $tpr \cdot fpr$ equivale, infatti, a calcolare la percentuale di coppie *(positivo, negativo)* per le quali il classificatore ha correttamente classificato l'istanza positiva compiendo un errore nella classificazione dell'istanza negativa. Sempre nella sezione 3.4.5, verrà dimostrato come questo termine corrisponda alla probabilità che, presa una coppia di istanze, positiva e negativa scelte casualmente, lo score della positiva sia correttamente ordinato rispetto allo score di una terza istanza negativa di riferimento, scelta anche essa casualmente, e lo score dell'istanza negativa della coppia sia ordinato non correttamente. Formalmente:

$$tAUC = AUC - \Pr(x_i > y_k < y_j) \tag{3.11}$$

Ricordando il significato statistico dell'AUC, mostrato nella (3.1), otteniamo:

$$tAUC = \Pr(x_i > y_k) - \Pr(x_i > y_k < y_j) \tag{3.12}$$

La (3.12) mostra come il $tAUC$ sia pari all'AUC, che rappresenta la capacità del classificatore di separare le due classi, ovvero la capacità di effettuare un corretto ranking relativo, corretta di un fattore pari all'errore compiuto dal classificatore nel ranking delle istanze negative.

Un'altra caratteristica che rende interessante il $tAUC$, e giustifica la sua scelta, è la possibilità di esprimere tale metrica facendo uso di un indicatore, analoga-

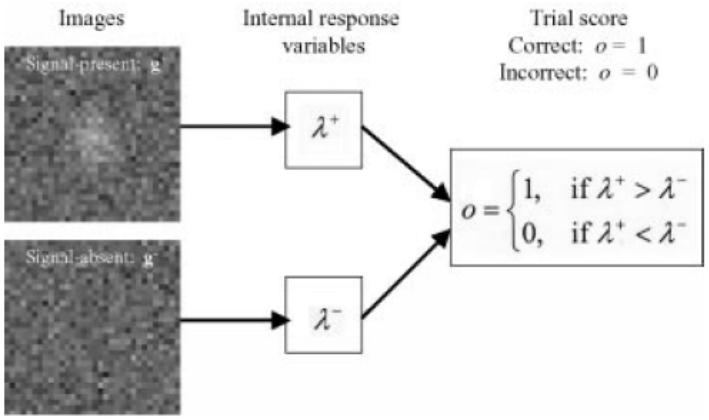


Figura 3.5: Modello di come un lettore effettua il 2AFC test

mente a quanto si fa per l'AUC. Per la dimostrazione si rimanda alla sezione 3.4.5.

Per completare l'analisi delle caratteristiche del $tAUC$, andiamo a descrivere i paradigmi sperimentali dei test diagnostici medici e il loro rapporto con le metriche ROC quali l'AUC e il nuovo $tAUC$.

La prima tipologia di test diagnostici è il cosiddetto *Yes-No method*. In questo tipo di procedura, viene presentata al lettore un'istanza alla volta, e il lettore deve decidere se l'istanza è positiva o negativa, fornendo un grado di confidenza, ovvero uno score, per l'istanza esaminata. Questo metodo corrisponde perfettamente al problema della classificazione a due classi, e la decisione del lettore corrisponde all'applicazione di un classificatore automatico ad un insieme di istanze appartenenti al dominio del problema. Per questo tipologia, dunque, l'AUC assume il significato di misura della capacità del classificatore di *separare correttamente* le istanze positive da quelle negative.

La seconda tipologia di test è il cosiddetto *Two-Alternative Forced Choice method* (TAFC o 2AFC). In questo tipo di procedura, vengono presentate al lettore due istanze alla volta, una positiva e una negativa. Il lettore deve decidere quale delle due è positiva e quale è negativa, fornendo per ciascuna immagine uno score relativo al grado di confidenza. In questo tipo di test il lettore ha a disposizione un'informazione supplementare: sa già che una delle due istanze che gli vengono presentate è positiva, mentre l'altra è negativa. La figura 3.5 mostra un modello del processo decisionale coinvolto in ciascuna delle presentazioni delle coppie $(positivo, negativo)$: a ciascuna immagine, il lettore associa un grado interno di confidenza, dopodiché confronta queste due misure e etichetta come positiva l'immagine con la confidenza più alta. In questo scenario, perciò,

l'AUC indica esattamente la accuratezza del classificatore. Ciò è legato strettamente al processo decisionale coinvolto nella classificazione: il lettore, essendo a conoscenza dell'informazione relativa alla presenza di un'istanza positiva e di un'istanza negativa, le confronta e sceglie tra le due quella che ritiene essere positiva con più probabilità. Nel 2AFC non è importante, dunque, che il classificatore produca degli score alti *in assoluto* per le istanze positive, ma che questi score siano alti in relazione a quelli delle istanze negative. Questa caratteristica farebbe si che le istanze positive, quando presentate in coppia ad istanze negative, vengano correttamente individuate. L'AUC misura proprio la probabilità che lo score di un'istanza positiva sia maggiore dello score di un'istanza negativa, e di conseguenza misura la probabilità che il classificatore discrimini correttamente l'istanza positiva fra le istanze della coppia presentata nel 2AFC.

Un discorso differente va fatto per il $tAUC$. A questa nuova metrica è associato un diverso tipo di test, che potremmo chiamare *Two-Alternative Ranking Experiment* (2ARE). Questa procedura consiste nel presentare al lettore una coppia di istanze, una positiva e una negativa, analogamente al 2AFC, ma, al contrario del 2AFC, senza specificare che una delle due istanze è positiva e l'altra è negativa. Si chiede dunque al lettore di ordinare le due istanze secondo il grado di positività da egli assegnato a ciascuna immagine, grado di positività che viene fornito dal lettore come score per ciascuna immagine. Il $tAUC$ dunque rappresenta la misura della bontà dell'ordinamento ottenuto per questo tipo di esperimento. Questa tipologia di test, rappresenta una procedura più generale rispetto al 2AFC, che può essere visto come una versione particolare del 2ARE in cui viene fornita al lettore l'informazione supplementare relativa alla presenza di un'istanza positiva e di un'istanza negativa all'interno della coppia. La maggiore generalità della procedura rende il 2ARE più "difficile" rispetto al 2AFC, e ciò giustifica anche la relazione tra il $tAUC$ e l'AUC. In base alla (3.10) e alla (3.11), infatti, si nota come $tAUC \leq AUC$.

3.4.4 Dal tAUC al tROC Space

Come abbiamo visto nella sezione 3.4.3, scalando i valori del *tpr* con il *true negative rate,* nel calcolo del $tAUC$, andiamo in effetti a calcolare la percentuale di istanze positive e di istanze negative correttamente classificate se presentate contemporaneamente. Visto l'importante significato di questa quantità, abbiamo pensato di applicare una trasformazione allo spazio ROC e plottare tale percentuale, che chiameremo *correct response rate*, in funzione del *false positive rate.*

Al contrario di quanto riportato in [7] per il $pAUC$, questa trasformazione è

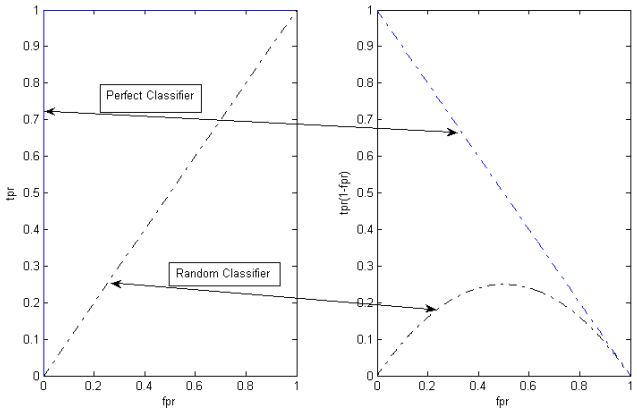


Figura 3.6: Effetti della trasformazione nello spazio tROC sulle curve ROC del classificatore perfetto e del classificatore casuale.

immediata e non richiede particolari artifici, in quanto consiste solamente nello scalare i valori del *tpr* moltiplicandoli per i corrispondenti valori di $(1 - fpr)$. Abbiamo quindi trasformato lo spazio ROC nello *spazio tROC*.

In questo modo andiamo a trasformare le curve ROC nelle equivalenti curve *tROC*. Nella figura 3.6 sono mostrate le curve tROC di due classificatori particolari: il classificatore perfetto e il classificatore casuale. Come possiamo vedere, la curva ROC del classificatore perfetto nello spazio tROC diventa la retta $y = 1 - x$. In base a tale osservazione, possiamo determinare, analogamente a quanto fatto per l'AUC, il valore massimo del tAUC. Poiché nessun classificatore può avere performance migliori del classificatore perfetto, il tAUC massimo sarà pari all'area della curva tROC del classificatore perfetto. Tale area è pari a 0.5. Segue dunque che il tAUC sarà limitato superiormente da:

$$tAUC \leq tAUC_{max} = 0.5.$$

Per quanto riguarda la curva ROC del classificatore casuale, invece, osserviamo che essa corrisponde alla curva tROC descritta dalla parabola $y = x(1 - x)$. Analogamente alla considerazione fatta per il classificatore perfetto, andiamo a calcolare il limite inferiore di performance del tAUC. Tale limite corrisponde infatti all'area della curva corrispondente al classificatore casuale, la quale è pari a $\frac{1}{6}$. Segue dunque che il valore minimo per il tAUC è pari a:

$$tAUC \geq tAUC_{min} = \frac{1}{6}.$$

Classificatore	tAUC (phoneme)	tAUC (pima)
Logistic	0.3446	0.3667
Nearest Mean	0.3467	0.2476
LDC	0.3489	0.3661
QDC	0.3660	0.3484
Parzen	0.4408	0.3349
RBF	0.4234	0.2149
Fisher	0.3487	0.3658
Naive Bayes	0.3803	0.3585
K-NN	0.4497	0.3365

Tabella 3.5: Misure tAUC per i classificatori applicati al Phoneme Dataset e al Pima Dataset

A questo punto va fatta una doverosa precisazione su tale limite. Il valore del tAUC del classificatore casuale non è effettivamente un limite inferiore, in quanto il tAUC per un classificatore può scendere al di sotto di tale valore, basta che performi peggio del classificatore casuale. Come indicato nella sezione 3.2.2 relativa al classificatore casuale, però, se un classificatore ha una curva ROC, e quindi una performance, al di sotto di quella del classificatore casuale, basta invertire le class-label dell'output per ottenere una performance accettabile. Lo stesso procedimento si può adottare per il tAUC e la relativa curva tROC, rendendo di fatto il valore del tAUC del classificatore casuale un *"limite inferiore"*.

3.4.4.1 Alcune osservazioni

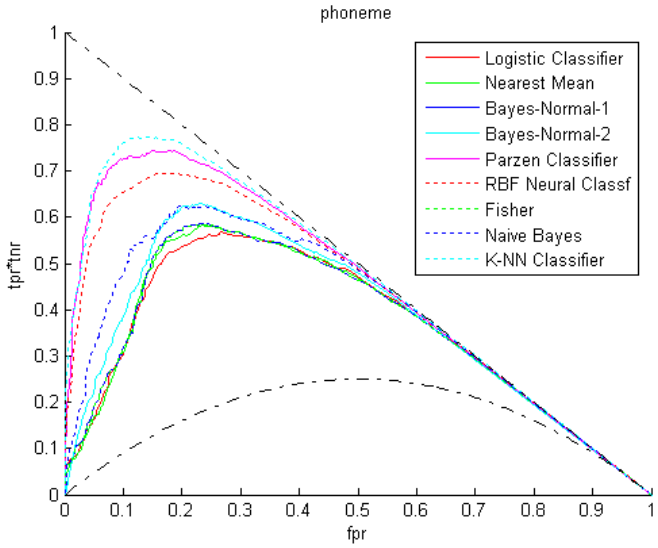
Andiamo ora a vedere l'effetto della trasformazione nel tROC space sulle curve ROC relative ai due dataset analizzati in precedenza.

Nella figura 3.7 sono mostrate, rispettivamente, le curve tROC dei classificatori applicati al phoneme dataset (a), e le curve tROC dei classificatori applicati al pima dataset. Associate a queste curve sono riportati, in tabella 3.5, i valori del tAUC di ciascun classificatore per i due dataset.

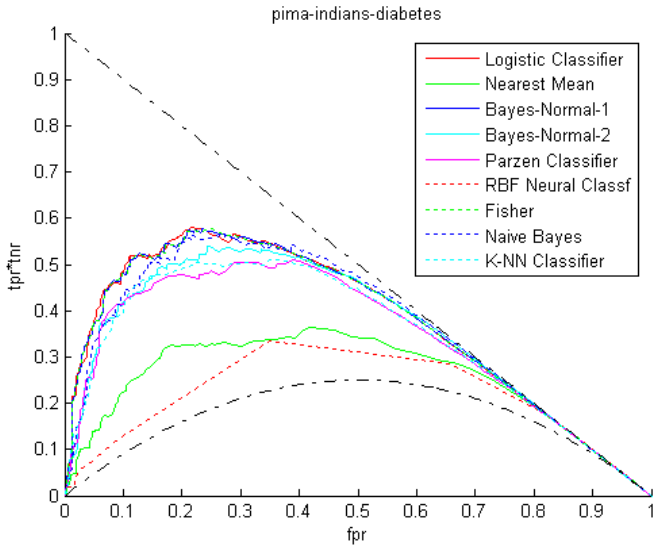
Si può subito osservare come, in entrambe le figure, tutte le curve tROC siano comprese tra le curva del classificatore perfetto e la curva del classificatore casuale. Queste due curve svolgono quindi una funzione di delimitatori geometrici per lo spazio tROC.

Andiamo ora ad analizzare le indicazioni che le curve tROC e l'indice tAUC forniscono sulle performance dei classificatori.

Nel caso del phoneme dataset, osserviamo come ci sia una netta prevalenza della curva tROC del classificatore K-NN rispetto alle curve degli altri classificatori.

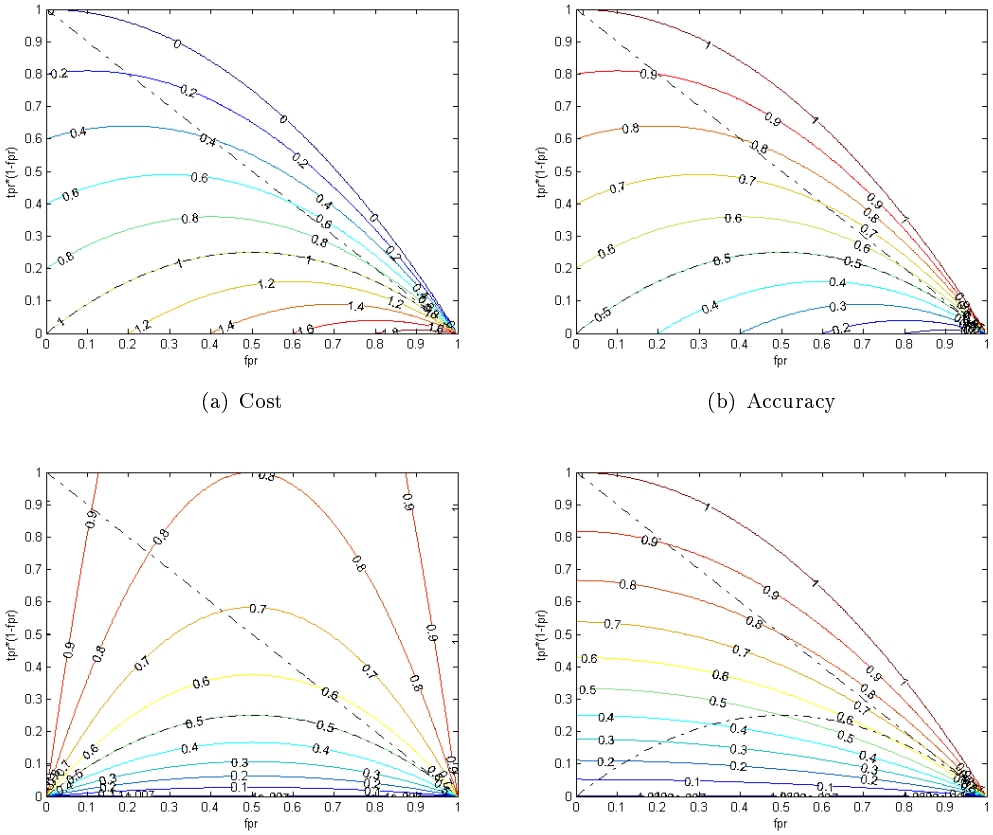


(a)



(b)

Figura 3.7: Curve tROC relative ai due dataset considerati: (a) phoneme e (b) pima indians diabetes



(a) Cost

(b) Accuracy

(c) Precision

(d) F-Measure

Figura 3.8: Curve di isoperformance nello spazio tROC

Tale superiorità è confermata anche dall'indice tAUC il cui valore è 44.97%, significativamente vicino al $tAUC_{max}$ del 50%. Tale risultato concorda con quanto ottenuto tramite la ROC analysis e l'AUC.

Nel caso del pima dataset, invece, osserviamo come sia la curva del Logistic classifier a prevalere sulle altre nella maggior parte dello spazio tROC. Tale predominanza non è netta, come confermato dal tAUC, il quale supera di misura il tAUC dei classificatori le cui curve tROC sono circa a "pari merito" con la curva tROC del Logistic classifier.

3.4.4.2 tROC Performance Isometrics

Analogamente a quanto fatto per le curve ROC, possiamo applicare la stessa trasformazione alle curve di isoperformance, completando in questo modo la caratterizzazione geometrica dello spazio tROC, e rendendo la *tROC Analysis*

uno strumento indipendente e completo per la valutazione delle performance dei classificatori.

Andiamo perciò ad esprimere le curve di isoperformance in funzione del *correct response rate* P_{cr}. Il *correct response rate* è stato definito precedentemente come:

$$p_{cr} = tpr \cdot (1 - fpr)$$

segue dunque che tpr può essere espresso, in funzione di p_{cr} e fpr, come:

$$tpr = \frac{p_{cr}}{1 - fpr} \tag{3.13}$$

A questo punto basta sostituire la (3.13) nelle equazioni (3.2), (3.4), (3.5) e (3.6), per ottenere le espressioni delle isoperformance lines delle rispettive metriche nello spazio tROC.

La notazione rimane la stessa utilizzata nella sezione 3.3.3.

Per il *misclassification cost* otterremo:

$$Cost \longleftrightarrow tCost = c(\boldsymbol{N}, \boldsymbol{p}) \left(1 - \frac{p_{cr}}{1 - fpr}\right) + c(\boldsymbol{Y}, \boldsymbol{n})\, fpr$$

Per la *accuracy* otterremo:

$$Accuracy \longleftrightarrow tAccuracy = \frac{\frac{p_{cr}}{1-fpr} + c(1 - fpr)}{1 + c}$$

Per la *precision* otterremo:

$$Precision \longleftrightarrow tPrecision = \frac{\frac{p_{cr}}{1-fpr}}{\frac{p_{cr}}{1-fpr} + c \cdot fpr}$$

Infine per la *f-measure* otterremo:

$$F\_Measure \longleftrightarrow tF\_Measure = \frac{2\frac{p_{cr}}{1-fpr}}{\frac{p_{cr}}{1-fpr} + c \cdot fpr + 1}$$

Nella figura 3.8 sono mostrate le curve di isoperformance nello spazio tROC, relative alle precedenti misure di performance. Si può osservare come tali curve, che nello spazio ROC erano delle rette, nello spazio tROC diventino delle parabole.

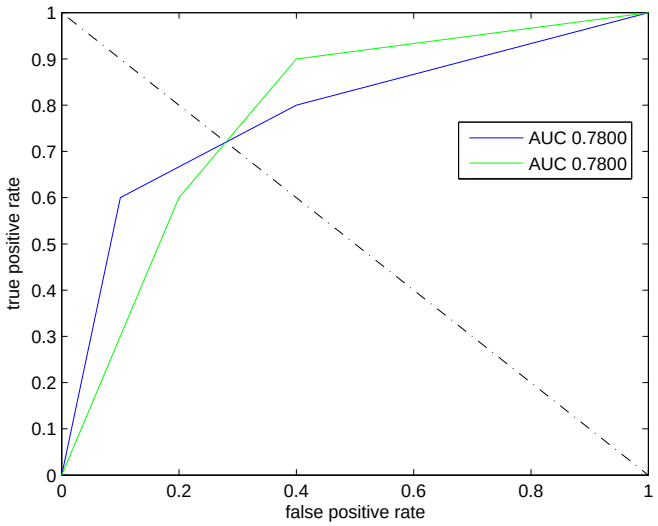


Figura 3.9: Curve ROC artificiali con la stessa AUC

3.4.4.3 tROC Analysis e Performance Tie

Oltre ad essere utilizzato come strumento indipendente di analisi, la tROC analysis e il tAUC possono essere affiancati alla ROC analysis tradizionale come ausilio decisionale nelle situazioni di *pareggio (Performance Tie)*.

Nella valutazione delle performance tramite le curve ROC e l'AUC si può verificare una situazione spiacevole. Due classificatori, infatti, possono trovarsi ad avere lo stesso valore dell'AUC pur avendo curve ROC diverse. Tale fenomeno va sotto il nome di *Performance Tie*, o pareggio. Uno dei due classificatori, però, sarà preferibile all'altro in quanto possiederà valori del *true positive rate* più alti nella parte iniziale dello spazio ROC, ossia nella zona in cui il *false positive rate* è più basso.

Un *performance tie* si verifica quando le curve ROC dei due classificatori sono in una particolare relazione, ossia le due curve sono simmetriche rispetto alla diagonale $y = 1 - x$. In questo caso, infatti, le due curve avranno la stessa area.

Nella figura 3.9 sono mostrate una curva ROC, costruita artificialmente, e la sua simmetrica rispetto alla diagonale $y = 1 - x$. Come si può vedere entrambe le curve hanno lo stesso AUC.

Nella figura 3.10, invece, è mostrato il grafico ROC del classificatore Parzen, in blu, applicato al phoneme dataset, e del suo simmetrico rispetto a $y = 1 - x$, in verde. Anche qui si può vedere come l'AUC sia la stessa.

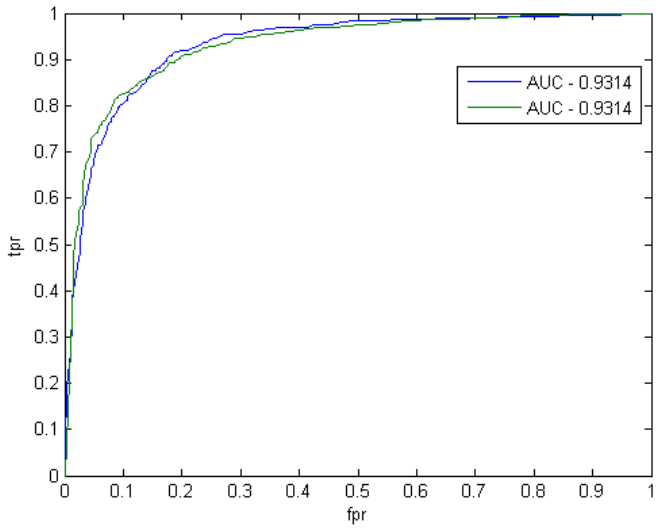


Figura 3.10: Curva ROC del Parzen classifier, applicato al phoneme dataset, e sua simmetrica

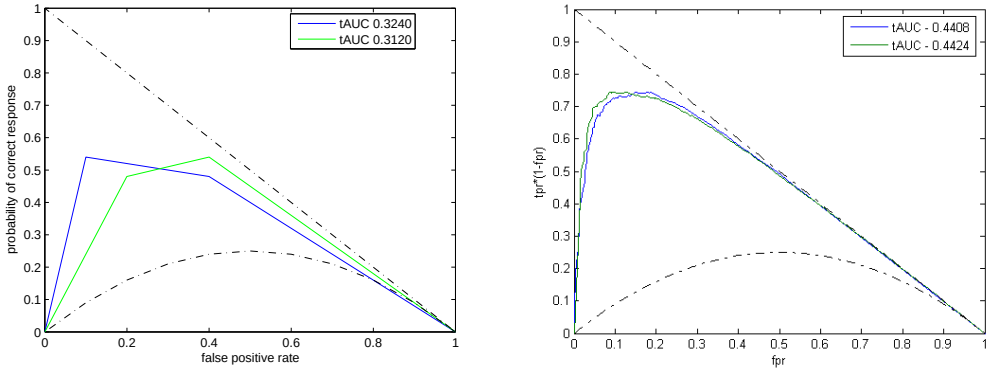
Per risolvere questa parità, scegliendo il classificatore con le performance migliori nella zona dello spazio ROC vicina all'origine, si può applicare facilmente il tAUC, il quale è stato disegnato appositamente per esaltare i classificatori che presentano performance migliori in tale zona.

Nella figura 3.11 nella pagina seguente sono mostrate le curve tROC relative alle due situazioni precedentemente presentate. Come si può vedere, sia nella figura 3.11(a) che nella figura 3.11(b), il tAUC permette di discriminare correttamente il classificatore con le performance migliori per bassi valori del *false positive rate*. Tale risultato ci permette di confermare la validità della tROC analysis e del tAUC come ausilio decisionale alla ROC analysis per risolvere efficacemente le situazioni di *performance tie*.

3.4.5 Proprietà matematiche e statistiche del tAUC

In questo paragrafo dimostreremo le proprietà e il significato statistico del *tAUC*. Per prima cosa verrà presentata la dimostrazione dell'equivalenza tra l'AUC e la statistica di Wilcoxon-Mann-Whitney. Dopodiché si procederà alla dimostrazione del significato statistico del *tAUC*.

Definiamo innanzitutto la notazione di cui faremo uso per entrambe le dimostrazioni.



(a) (b)

Figura 3.11: Curve tROC relative alle performance tie: (a) artificiali e (b) Parzen classifier

Sia I un'istanza appartenente ad un test-set etichettato $\mathbb{Z}$, e $\mathbf{x}$ il vettore di misura, nello spazio $\mathbb{X}$ delle misure, che la rappresenta.

Sia $\hat{p}(\mathbf{x}) = \Pr(\boldsymbol{Y} | \mathbf{x})$ la stima della probabilità a posteriori per un'istanza I, con vettore di misura $\mathbf{x}$, di appartenere alla classe positiva. Essa rappresenta l'output del classificatore.

Sia $f(\hat{p}) = f(\,\hat{p}(\mathbf{x}) \,|\, \boldsymbol{p})$ la densità di probabilità condizionale dei valori della stima della probabilità di appartenenza alla classe positiva per le istanze positive. Sia poi $F(t) = F(\,\hat{p}(\mathbf{x}) \leq t \,|\, \boldsymbol{p}) = \Pr\,(\hat{p}(\mathbf{x}) \leq t \mid \boldsymbol{p})$ la funzione di ripartizione associata a $f(\hat{p})$ e $\bar{F}(t) = 1 - F\,(t) = \Pr\,(\hat{p}(\mathbf{x}) > t \mid \boldsymbol{p})$. $F(t)$ e $\bar{F}(t)$ rappresentano, rispettivamente, il *false negative rate* e il *true positive rate* per il valore x della soglia.

Sia $g(\hat{p}) = g(\,\hat{p}(\mathbf{x}) \,|\, \boldsymbol{p})$ la densità di probabilità condizionale dei valori della stima della probabilità di appartenenza alla classe positiva per le istanze negative. Sia poi $G(t) = G(\,\hat{p}(\mathbf{x}) \leq t \,|\, \boldsymbol{n}) = \Pr\,(\hat{p}(\mathbf{x}) \leq t \mid \boldsymbol{n})$ la funzione di ripartizione associata a $g(\hat{p})$ e $\bar{G}(t) = 1 - G\,(t) = \Pr\,(\hat{p}(\mathbf{x}) > t \mid \boldsymbol{n})$. $G(t)$ e $\bar{G}(t)$ rappresentano, rispettivamente, il *true negative rate* e il *false positive rate* per il valore t della soglia.

3.4.5.1 Dimostrazione AUC

In base a questa notazione, la curva ROC è il plot di $\bar{F}(t)$, sull'asse y, in funzione di $\bar{G}(t)$, sull'asse x, ottenuto facendo variare la soglia t in $[\mathbf{t_{min}}, \mathbf{t_{max}}]$.

Dalla definizione di area integrale, l'AUC è pari a:

$$AUC = \int_{\bar{G}_{min}}^{\bar{G}_{max}} \bar{F}\left(\bar{G}\right) d\bar{G} \tag{3.14}$$

Per la definizione di funzione di ripartizione, $\bar{G}(t)$ è una funzione *monotona non crescente,* perciò avremo:

$$\bar{G}(t_1) \leq \bar{G}(t_2),\ t_1 \geq t_2 \tag{3.15}$$

Dalla definizione di $\bar{G}(t)$, il differenziale $d\bar{G}$ sarà pari a:

$$d\bar{G}(t) = d\left(1 - G\left(t\right)\right) = -g\left(t\right) dt \tag{3.16}$$

Ora possiamo osservare che la probabilità $\Pr\left(\hat{p}(\mathbf{x}) > t \mid \boldsymbol{p}\right)$ che un'istanza positiva sia correttamente classificata positiva quando la soglia è t, sarà pari a:

$$\Pr\left(\hat{p}(\mathbf{x}) > t \mid \boldsymbol{p}\right) = \bar{F}\left(t\right) h\left(t\right) dt \tag{3.17}$$

dove $h\left(t\right)$ è la densità di probabilità dei valori assunti da t.

La *probabilità totale*, ottenuta facendo assumere a t tutti i possibili valori, di avere un'istanza positiva correttamente classificata sarà perciò:

$$\Pr\left(\boldsymbol{Y} \mid \boldsymbol{p}\right) = \int_{t_{min}}^{t_{max}} \bar{F}\left(t\right) h\left(t\right) dt \tag{3.18}$$

Se t viene scelta in accordo con la distribuzione $G\left(t\right)$, ovvero t assume il valore di $\hat{p}(\mathbf{x})$ per un'istanza negativa scelta casualmente avremo:

$$h\left(t\right) = g\left(t\right) \tag{3.19}$$

e la probabilità in (3.18) diventerà la *probabilità che il* $\hat{p}(\mathbf{x})$ *di un'istanza positiva sia maggiore del* $\hat{p}(\mathbf{x})$ *di un'istanza negativa*

$$\Pr\left(\boldsymbol{Y} \mid \boldsymbol{p}\right) = \Pr(\hat{p}(\mathbf{x}) > \hat{p}(\mathbf{y})) = \int_{t_{min}}^{t_{max}} \bar{F}\left(t\right) g\left(t\right) dt \tag{3.20}$$

Dalla (3.15) sappiamo che

$$\bar{G}\left(t_{min}\right) = \bar{G}_{max},\ \bar{G}\left(t_{max}\right) = \bar{G}_{min} \tag{3.21}$$

Dalla (3.16) e dalla (3.21), e in base alla regola di integrazione per sostituzione otteniamo:

$$\int_{t_{min}}^{t_{max}} \bar{F}(t)\, g(t)\, dt \quad = \quad \int_{t_{max}}^{t_{min}} \bar{F}(t)\, d\bar{G}(t) \quad = \quad \int_{\bar{G}_{min}}^{\bar{G}_{max}} \bar{F}\left(\bar{G}\right) d\bar{G} \tag{3.22}$$

Combinando la (3.22) e la (3.14) otteniamo l'importante equivalenza

$$AUC = \Pr(\hat{p}(\mathbf{x}) > \hat{p}(\mathbf{y}))$$

che ci permette di calcolare l'AUC usando l'espressione della statistica di Wilcoxon-Mann-Whitney:

$$AUC = \sum_{j=1}^{N} \sum_{i=1}^{P} \frac{\mathcal{I}\left(\hat{p}\left(\mathbf{x_i}\right) > \hat{p}\left(\mathbf{y_j}\right)\right)}{N \cdot P}$$

dove N e P sono, rispettivamente, il totale delle istanze negative e il totale delle istanze positive.

3.4.5.2 Dimostrazione tAUC

Sempre in base alla notazione definita in precedenza, la curva tROC è il plot di $\bar{F}(t) \cdot G(t)$, sull'asse y, in funzione di $\bar{G}(t)$, sull'asse x, ottenuto facendo variare la soglia t in $[\mathbf{t_{min}}, \mathbf{t_{max}}]$.

Dalla definizione di area integrale, il tAUC è pari a:

$$tAUC = \int_{\bar{G}_{min}}^{\bar{G}_{max}} \bar{F}\left(\bar{G}\right) G\left(\bar{G}\right) d\bar{G} \tag{3.23}$$

Ora possiamo osservare che la probabilità $P(C)$ di avere contemporaneamente un'istanza positiva e un'istanza negativa classificate correttamente quando la soglia è t, sarà pari a:

$$P(C) \;=\; \Pr\left(\hat{p}(\mathbf{x}) > t \mid \boldsymbol{p}\right) \cdot \Pr\left(\hat{p}(\mathbf{y}) \leq t \mid \boldsymbol{n}\right) \;=\; \bar{F}(t)\, G(t)\, h(t)\, dt \tag{3.24}$$

dove $h(t)$ è la densità di probabilità dei valori assunti da t.

La *probabilità totale*, ottenuta facendo assumere a t tutti i possibili valori, di avere contemporaneamente un'istanza positiva e un'istanza negativa corretta-

mente classificate sarà perciò:

$$\Pr\left(I_1 \to \boldsymbol{Y},\, I_2 \to \boldsymbol{N} \mid I_1 \in \boldsymbol{p},\, I_2 \in \boldsymbol{n}\right) = \int_{t_{min}}^{t_{max}} \bar{F}\left(t\right) G\left(t\right) h\left(t\right) dt \tag{3.25}$$

Se t viene scelta in accordo con la distribuzione $G\left(t\right)$, ovvero t assume il valore di $\hat{p}(\mathbf{x})$ per un'istanza negativa scelta casualmente avremo:

$$h\left(t\right) = g\left(t\right) \tag{3.26}$$

e la probabilità in (3.25) diventerà la *probabilità che il* $\hat{p}(\mathbf{x})$ *di un'istanza positiva sia e il* $\hat{p}(\mathbf{x})$ *di un'istanza negativa siano correttamente ordinati rispetto al* $\hat{p}(\mathbf{x})$ *di una terza istanza negativa*

$$\begin{aligned}\Pr\left(I_1 \to \boldsymbol{Y},\, I_2 \to \boldsymbol{N} \mid I_1 \in \boldsymbol{p},\, I_2 \in \boldsymbol{n}\right) &= \Pr(\hat{p}(\mathbf{x}) > \hat{p}(\mathbf{z}) \geq \hat{p}(\mathbf{y})) \\ &= \int_{t_{min}}^{t_{max}} \bar{F}\left(t\right) G\left(t\right) g\left(t\right) dt \end{aligned} \tag{3.27}$$

Dalla (3.15) sappiamo che

$$\bar{G}\left(t_{min}\right) = \bar{G}_{max},\ \bar{G}\left(t_{max}\right) = \bar{G}_{min} \tag{3.28}$$

Dalla (3.16) e dalla (3.28), e in base alla regola di integrazione per sostituzione otteniamo:

$$\int_{t_{min}}^{t_{max}} \bar{F}\left(t\right) G\left(t\right) g\left(t\right) dt = \int_{t_{max}}^{t_{min}} \bar{F}\left(t\right) G\left(t\right) d\bar{G}\left(t\right) = \int_{\bar{G}_{min}}^{\bar{G}_{max}} \bar{F}\left(\bar{G}\right) G\left(\bar{G}\right) d\bar{G} \tag{3.29}$$

Combinando la (3.29) e la (3.23) otteniamo l'importante equivalenza

$$tAUC = \Pr(\hat{p}(\mathbf{x}) > \hat{p}(\mathbf{z}) \geq \hat{p}(\mathbf{y}))$$

che ci permette di calcolare il tAUC usando un indicatore, analogamente a quanto si fa per l'AUC:

$$tAUC = \sum_{k=1}^{N} \sum_{j=1}^{N} \sum_{i=1}^{P} \frac{\mathcal{I}\left(\hat{p}\left(\mathbf{x_i}\right) > \hat{p}\left(\mathbf{z_k}\right) \geq \hat{p}\left(\mathbf{y_j}\right)\right)}{N \cdot N \cdot P}$$

dove N e P sono, rispettivamente, il totale delle istanze negative e il totale delle

istanze positive.

Bibliografia

[1] F. Provost, T. Fawcett, "Robust classification for imprecise environments", Machine Learning, vol. 42, n. 3, 2001, p. 203-231.

[2] P. Flach and S. Wu, "A scored AUC Metric for Classifier Evaluation and Selection", Proc. ICML 2005 Workshop on ROC Analysis in Machine Learning, 2005.

[3] J.A. Hanley and B.J. McNeil, "The Meaning and Use of the Area Under a Receiver Operating Characteristic (ROC) Curve", Radiology, vol. 143, pp. 29-36, 1982.

[4] D.J. Hand and R.J. Till, "A simple generalisation of the Area Under the ROC Curve for Multiple-Class Classification Problems", Machine Learning, vol. 45, no. 2, pp. 171-186, 2001.

[5] A.P. Bradley, "The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms", Pattern Recognition, vol. 30, no. 7 pp.1145-1159, 1997.

[6] T. Fawcett, "ROC Graphs: Notes and Practical Considerations for Researchers", Machine Learning, 2004.

[7] C. Ferri, P. Flach, J. Hernández-Orallo, A. Senad, "Modifying ROC Curves to Incorporate Predicted Probabilities", Proc. ICML 2005 Workshop on ROC Analysis in Machine Learning, 2005.

[8] S. J. Mason and N. E. Graham , "Conditional Probabilities, Relative Operating Characteristics, and Relative Operating Levels", Weather and Forecasting, vol. 14, pp. 713-725, 1999.

[9] E. F. Schisterman, D. Faraggi, B. Reiser, M. Trevisan, "Statistical Inference for the Area under the Receiver Operating Characteristic Curve in the Presence of Random Measurement Error", American Journal of Epidemiology, Vol. 154, No. 2, pp. 174-179, 2001.

[10] P. A. Flach, "The Geometry of ROC Space: Understanding Machine Learning Metrics through ROC Isometrics", Proc. 20th Int. Conf. on Machine Learning ICML-03, 2003.

[11] D. R. Lovell, C. R. Dance, M. Niranjan, R. W. Prager, K. J. Dalton, "Ranking the effect of different features on the classification of discrete valued data," in Engineering Applications of Neural Networks, pp. 487–494, 1996.

[12] C. K. Abbey, M. P. Eckstein, "Optimal Shifted Estimates of Human-Observer Templates in Two-Alternative Forced-Choice Experiments", IEEE Trans. on Medical Imaging, vol. 21, no. 5, 2002.

[13] A. I. Schulman, R. R. Mitchell, "Operating Characteristics from Yes-No and Forced-Choice Procedures", Journal of the Acoustical Society of America, vol. 40, no. 2, pp. 473-477, 1966.

[14] C. K. Abbey, M. P. Eckstein, "Classification image analysis: Estimation and statistical inference for two-alternative forced-choice experiments", Journal of Vision, vol. 2, no. 1, pp. 66-78, 2002.

[15] Newman, D.J. & Hettich, S. & Blake, C.L. & Merz, C.J. (1998). UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

[16] C. Aviles-Cruz, A. Guérin-Dugué, J.L. Voz, D. Van Cappel, Enhanced Learning for Evolutive Neural Architecture [http://www.dice.ucl.ac.be/mlg/?page=Elena]

Capitolo 4

Risultati Sperimentali

4.1 Rilevatori di noduli

Il compito di un rilevatore di noduli è quello di classificare le ROI 3D, rispettivamente, in noduli e vasi sanguigni.

Un rilevatore di noduli è definito dalla scelta di un classificatore e dall'insieme di caratteristiche su cui opera, oltre che, nel caso di classificatori parametrici, dagli eventuali parametri specifici di inizializzazione.

Nell'ambito della realizzazione di un sistema CAD, siamo interessati a realizzare un insieme di classificatori diversi (che chiameremo *classificatori di primo livello*) e a combinare opportunamente le loro uscite (attraverso un *classificatore di secondo livello*, che costituirà il generatore di noduli) in modo da ottenere prestazioni migliori di quelle del miglior singolo classificatore. Allo scopo, si può utilizzare lo stesso classificatore di primo livello su insiemi di caratteristiche diversi, oppure classificatori di primo livello distinti che lavorano sullo stesso insieme di caratteristiche, ovvero approcci ibridi. Per la realizzazione del nostro sistema è stato seguito il secondo approccio, il quale fa uso della metodologia del *ROC Convex Hull* descritta da Provost et al, in [1].

Per distinguere noduli da vasi sanguigni si considerano, per ciascuna ROI 3D, la forma e il valore di grigio. I noduli tendono, infatti, ad essere più circolari dei vasi sanguigni; d'altra parte, i valori di grigio dei vasi sanguigni che scorrono verticalmente rispetto all'immagine 2D sono di solito più alti di quelli di un nodulo della stessa dimensione. Di conseguenza, sono state estratte quattro caratteristiche da ciascuna ROI 3D in modo da rappresentare le ROI stesse come vettori in $\Re^4$. Con riferimento alla generica ROI 3D (sia R_i) le caratteristiche scelte sono le seguenti:

- *sfericità*

$$s = \frac{V(R_i \cap S_i)}{V(R_i)}$$

dove $V()$, R_i, S_i sono, rispettivamente, il volume, la ROI 3D e la sfera equivalente a tale ROI (ovvero la sfera con centro nel baricentro di R_i e raggio uguale a quello della sfera con volume uguale a $V(R_i)$);

- *livello medio di grigio* g_m, calcolato come media dei livelli di grigio dei voxel che compongono R_i;

- *deviazione standard del livello medio di grigio* $dev(g_m)$;

- *inverso dell'elongazione*

$$\frac{1}{el} = \frac{8\left[\max_{j \in J(R_i)}(d_j)\right]^2}{\sum_{j \in J(R_i)} A(R_j)}$$

dove $A()$ è l'area, $max()$ è il massimo, e d_j è lo spessore della j-esima ROI 2D costituente la ROI 3D, ovvero il numero di passi di erosione necessari per far sparire tale ROI 2D.

Nell'ambito del progetto di realizzazione del sistema CAD polmonare, sono stati considerati i seguenti cinque classificatori (di primo livello): a discriminazione lineare (LDC), a discriminazione quadratica (QDC), logistico (LOGLC), albero di decisione (TREEC) e radial basis function network (RBNC). I primi quattro classificatori sono non parametrici, mentre il quinto è parametrico. Sono state inoltre considerate varie tecniche di combinazione per la realizzazione del classificatore di secondo livello che costituisce il generatore di noduli.

Per quanto riguarda questo lavoro di tesi, invece, ci siamo focalizzati sulla analisi delle performance dei classificatori di primo livello, approfondendo le valutazioni relative al progetto principale tramite l'applicazione della tecnica della *tROC Analysis* a un numero maggiore di classificatori.

4.2 Pianificazione degli esperimenti

Al fine di valutare le performance dei classificatori da selezionare per la realizzazione dei rilevatori di primo livello, sono stati effettuati tre esperimenti. Si è utilizzato un dataset $X = X_n \cup X_v$ composto da 1984 ROI 3D, suddivise in 64 noduli, che compongono l'insieme X_n e corrispondono a circa il 3.2% del totale, e 1920 vasi sanguigni, che compongono l'insieme X_v e corrispondono a circa il 96.8% del totale. Tutti i dati utilizzati nelle prove sono rappresentati come

punti in $\Re^4$, dopo aver estratto dalle singole ROI 3D le quattro caratteristiche definite precedentemente in sezione 4.1.

In tutti e tre gli esperimenti ci si è basati sulla *cross-validation*, perché è una tecnica ampiamente utilizzata che ha il vantaggio, rispetto ad altre tecniche, di usare, in momenti diversi, tutti i punti del dataset sia in fase di addestramento che in fase di test. Poiché la cross-validation prevede un numero di prove al suo interno, le performance finali sono state ottenute come media delle performance sul test set di ogni singola prova. In particolare, si è utilizzata una 5-fold cross-validation in cui i cinque fold sono gli stessi per tutti e tre gli esperimenti.

4.2.1 Esperimento 1

Per tenere in considerazione il fatto che i dati reali hanno una distribuzione fortemente sbilanciata a favore dei negativi (vasi sanguigni), abbiamo effettuato il test su un dataset sbilanciato 1:30 (a favore dei negativi), così come è il nostro dataset iniziale. Tale rapporto positivi/negativi sul test set si riflette nello stesso rapporto sul training set, qualora si adoperi una cross-validation classica. Più precisamente, da un punto di vista operativo, abbiamo suddiviso l'insieme X in 5 fold X^i, $i = 1, \ldots 5$, ed abbiamo costruito gli insiemi di addestramento $TR^j = \bigcup_{i \neq j} X^i$ e di test $TS^j = X^j$, $j = 1, \ldots 5$.

4.2.2 Esperimento 2

La considerazione sulla quale si basa il secondo esperimento riguarda il fatto che alcuni rilevatori possono essere influenzati dalla distribuzione delle classi più di altri. Per quei rilevatori basati sulla minimizzazione dell'errore quadratico medio prodotto in uscita, infatti, la classe maggiormente rappresentata tenderà a monopolizzare le operazioni di correzione dell'errore. Per questo motivo si preferisce addestrare tali sistemi con uno stesso numero di esempi per le due classi.

Usando un procedimento basato su clustering (in particolare, si è usato l'algoritmo fuzzy c-means), gli esempi negativi di ciascuno dei cinque fold usati per il training sono stati ridotti di un fattore 30. Il test è rimasto lo stesso di prima, ossia si è mantenuto un rapporto 1:30. Più precisamente, si sono creati dapprima i 5 insiemi di addestramento $TR^1, \ldots, TR^5$ e di test $TS^1, \ldots, TS^5$ dell'esperimento precedente, poi ciascun TR^i è stato sostistuito con $\bar{TR}^i$ ottenuto applicando l'algoritmo fuzzy c-means sugli esempi negativi di TR^i, con un numero di cluster pari al numero di esempi positivi di TR^i. I punti negativi

Classificatore	AUC	tAUC
Logistic	0.9084	0.4280
Nearest Mean	0.4838	0.04562
LDC	0.9006	0.4256
QDC	0.8945	0.4235
Parzen	0.5252	0.0865
RBF	0.8543	0.4076
Fisher	0.7239	0.2249
Naive Bayes	0.8956	0.4196
K-NN	0.8591	0.3910

Tabella 4.1: Tabella dei valori AUC e tAUC dei classificatori per l'esperimento 1

più vicini ai centroidi prodotti dall'algoritmo sono stati poi inseriti come esempi negativi in $\bar{TR}^i$ insieme agli esempi positivi di TR^i.

4.2.3 Esperimento 3

Il terzo esperimento è simile al secondo esperimento, differendo da questo per il fatto che la riduzione degli esempi negativi nel training set avviene per scelta casuale invece che attraverso un processo di clustering.

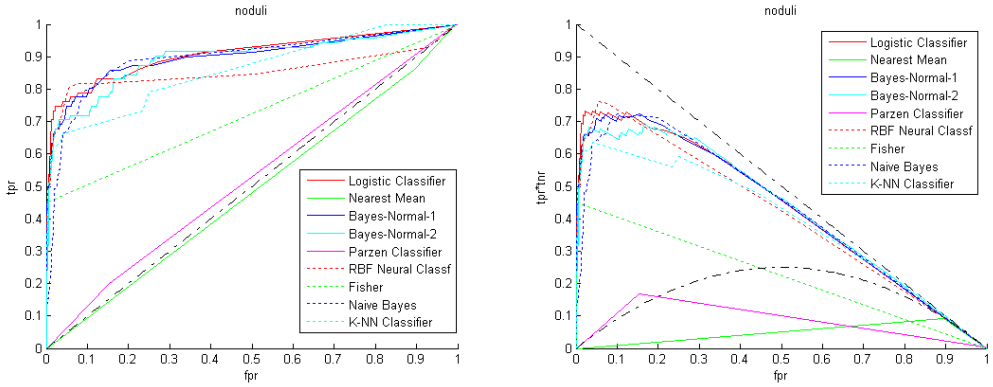
4.3 Risultati

Riportiamo ora i risultati dell'applicazione dei classificatori al dataset dei noduli per ciascun esperimento.

4.3.1 Esperimento 1

Nella figura 4.1 sono riportate le curve ROC, (a), e le curve tROC dei classificatori applicati al dataset dell'esperimento 1. Nella tabella 4.1 sono riportati i rispettivi valori dell'AUC e del tAUC di ciascun classificatore.

Possiamo innanzitutto osservare come non ci sia nessun classificatore che prevalga nettamente sugli altri. Nessuna delle curve ROC e delle curve tROC dei vari classificatori prevale nettamente, su tutto lo spazio ROC e su tutto lo spazio tROC rispettivamente. Notiamo inoltre la presenza di due classificatori con performance inferiori a quelle del classificatore casuale. Nello specifico, tali classificatori sono il Parzen, con un'AUC del 52.52%, e il Nearest Mean, con un'AUC del 48.38%. Osserviamo poi una caratteristica interessante del tROC



(a) (b)

Figura 4.1: Curve ROC e tROC dei classificatori per l'esperimento 1

Classificatore	AUC	tAUC
Logistic	0.8945	0.4183
Nearest Mean	0.4774	0.0283
LDC	0.8969	0.4225
QDC	0.8546	0.4029
Parzen	0.5726	0.1199
RBF	0.7383	0.3389
Fisher	0.6744	0.1761
Naive Bayes	0.8996	0.4226
K-NN	0.8046	0.3141

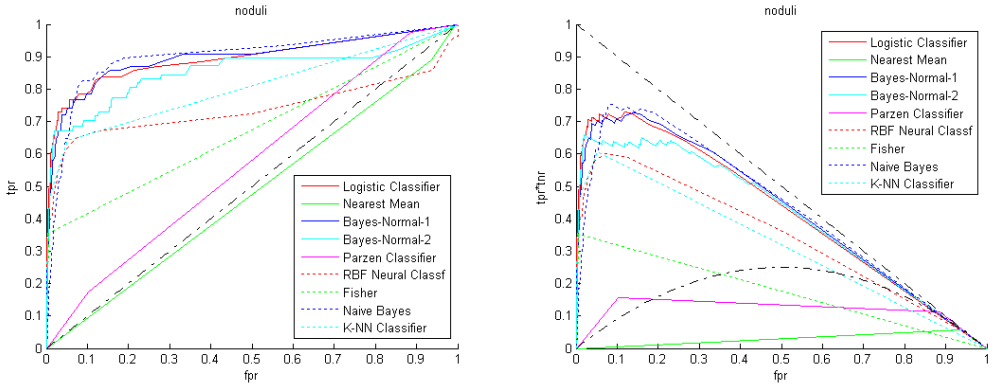
Tabella 4.2: Tabella dei valori AUC e tAUC dei classificatori per l'esperimento 2

space. Nella figura 4.1(b), nella quale sono riportate le curve tROC per l'esperimento 1, questa inferiorità di performance dei due classificatori nei confronti del classificatore casuale, è resa ancora più evidente che attraverso le curve ROC della figura 4.1(a).

Per quanto riguarda le performance generali, il miglior risultato è stato ottenuto dal classificatore Logistic con un'AUC del 90.84%, seguito dal classificatore LDC, con un'AUC del 90.06%. I risultati ottenuti in termini di tAUC concordano con quelli ottenuti in termini di AUC.

4.3.2 Esperimento 2

Nella figura 4.2 sono riportate le curve ROC, (a), e le curve tROC dei classificatori applicati al dataset dell'esperimento 2. Nella tabella 4.2 sono riportati i rispettivi valori dell'AUC e del tAUC di ciascun classificatore.



(a) (b)

Figura 4.2: Curve ROC e tROC dei classificatori per l'esperimento 2

Rispetto ai risultati ottenuti nell'esperimento 1, possiamo osservare alcuni cambiamenti relativi alla situazione ottenuta per l'esperimento 2. Per prima cosa notiamo alcuni cambiamenti fra le performance dei vari classificatori. Nella fattispecie, il classificarore rbf ha ottenuto un netto calo di prestazioni rispetto all'esperimento 1, mentre si è verificato un aumento delle performance del classificatore naive bayes, la cui curva ROC, da un punto dello spazio in poi, domina sulle altre. Rimane costante la scarsità di performance, per questa applicazione, dei classificatori Nearest Mean e Parzen, e in particolare si è ottenuto un lieve miglioramento delle prestazioni di quest'ultimo affiancato ad un peggioramento delle performance del primo.

Per quanto riguarda le performance generali, il classificatore che ha ottenuto le prestazioni migliori per questo esperimento è stato il classificatore Naive Bayes, con un'AUC del 89.96%. Il risultato in termini di AUC conferma la configurazione spaziale evidenziata dalla curva ROC. Secondo in termini di performance è stato il classificatore LDC con un'AUC del 89.69%. Anche in questo esperimento i risultati in termini di tAUC confermano i risultati ottenuti con la valutazione dell'AUC.

4.3.3 Esperimento 3

Nella figura 4.3 sono riportate le curve ROC, (a), e le curve tROC dei classificatori applicati al dataset dell'esperimento 2. Nella tabella 4.3 sono riportati i rispettivi valori dell'AUC e del tAUC di ciascun classificatore.

Questo esperimento ha dato dei risultati notevoli che permettono di mettere in evidenza l'efficacia della tROC analysis, sia come strumento indipendente sia

Classificatore	AUC	tAUC
Logistic	0.8945	0.4181
Nearest Mean	0.5233	0.07562
LDC	0.9019	0.4259
QDC	0.8780	0.4135
Parzen	0.5031	0.1296
RBF	0.8515	0.3966
Fisher	0.6821	0.1847
Naive Bayes	0.9059	0.4245
K-NN	0.8344	0.3747

Tabella 4.3: Tabella dei valori AUC e tAUC dei classificatori per l'esperimento 3

affiancata alla ROC analysis tradizionale. Innanzitutto osserviamo un miglioramento delle performance, rispetto all'esperimento 2, del classificatore LDC e del classificatore logistico nei confronti del Naive Bayes. Notiamo inoltre come le curve ROC di questi ultimi due classificatori siano quasi coincidenti nella prima parte dello spazio ROC. Tale equivalenza di prestazioni si può notare anche nello spazio tROC, dove le due curve dei classificatori seguono un andamento molto simile.

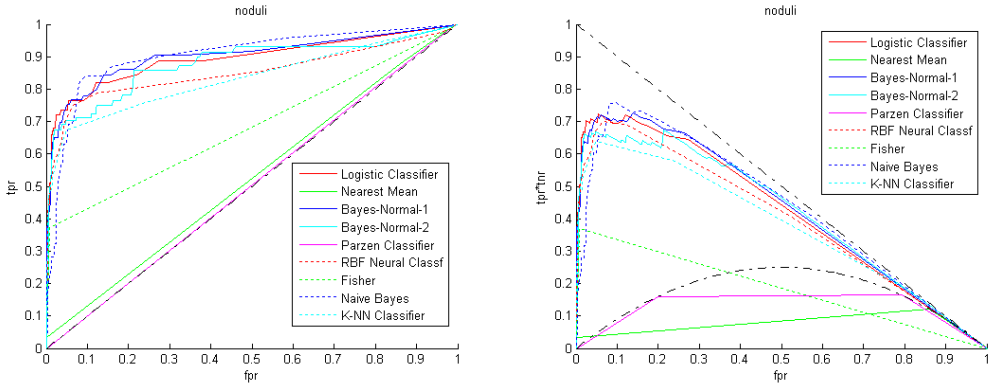
In termini di performance generali, continua a prevalere il classificatore Naive Bayes con un'AUC del 90.59%, seguito dal classificatore LDC, con un'AUC del 90.19%.

La misura delle performance in termini di tAUC, però, non concorda con i risultati ottenuti per l'AUC. Il classificatore LDC, infatti, ha un tAUC del 42.59% maggiore, seppur di poco, rispetto al tAUC del Naive Bayes il quale è pari al 42.45%. Ciò è confermato dall'andamento delle curve tROC. Per bassi valori di fpr, la curva del Naive Bayes è infatti più bassa di quella del classificatore LDC, sia nello spazio ROC sia soprattutto nello spazio tROC dove questa caratteristica è evidenziata maggiormente. tale risultato conferma l'abilità del tAUC di fornire una misura globale di performance, evidenziando però le prestazioni ottenute per bassi valori del *false positive rate*.

4.3.4 Sistema CAD

In questa sezione riportiamo, brevemente, i risultati ottenuti nella valutazione delle performance delle strategie di classificazione e di combinazione, selezionate per la realizzazione, rispettivamente, dei rilevatori di primo livello e del combinatore di secondo livello.

Per quanto riguarda i rilevatori di primo livello, si è osservata una prevalenza del classificatore LDC in termini di performance globali, misurate tramite l'AUC.



(a) (b)

Figura 4.3: Curve ROC e tROC dei classificatori per l'esperimento 3

Metodo	max $\frac{\Pr(\boldsymbol{n})c(\boldsymbol{Y},\boldsymbol{n})}{\Pr(\boldsymbol{p})c(\boldsymbol{N},\boldsymbol{p})}$	min $\frac{\Pr(\boldsymbol{n})c(\boldsymbol{Y},\boldsymbol{n})}{\Pr(\boldsymbol{p})c(\boldsymbol{N},\boldsymbol{p})}$
rbnc	0,0688	0
loglc	0,1750	0,0688
ldc	2,0555	0,1750
loglc	179,2	2,0555
ldc	$+\infty$	179,2

Tabella 4.4: Intervalli di ottimalità dei rilevatori di noduli considerati singolarmente

Come per gli esperimenti eseguiti per questo lavoro di tesi, anche qui non è emerso un classificatore che predominasse significativamente rispetto agli altri su tutto lo spazio ROC.

Come si era accennato in precedenza, si è utilizzata la strategia ibrida del ROC Convex-Hull, che permette l'estrazione, dall'insieme dei classificatori valutati, degli elementi che contribuiscono significativamente alle prestazioni globali del sistema, e la *selezione dinamica* del miglior classificatore per le reali condizioni operative del sistema.

L'ottimalità locale è misurata in termini del rapporto tra il costo dei falsi positivi e il costo dei falsi negativi, $\frac{\Pr(\boldsymbol{n})c(\boldsymbol{Y},\boldsymbol{n})}{\Pr(\boldsymbol{p})c(\boldsymbol{N},\boldsymbol{p})}$, il quale indica la pendenza della retta di *misclassification cost* tangente alla curva ROC del convex hull. La tabella 4.4 mostra, per ciascun rilevatore localmente ottimo, gli intervalli di pendenze delle retto isocosto in ciascun punto di tangenza al ROC convex hull. La figura 4.4, invece, descrive i rilevatori di noduli che danno un contributo al ROC convex hull: nello specifico, sono mostrati i punti del ROC convex hull e il classificatore associato ad ogni punto. Nella figura 4.5, infine, sono riportati il ROC convex hull ottenuto per i rilevatori, quello ottenuto per i combinatori, e il ROC convex

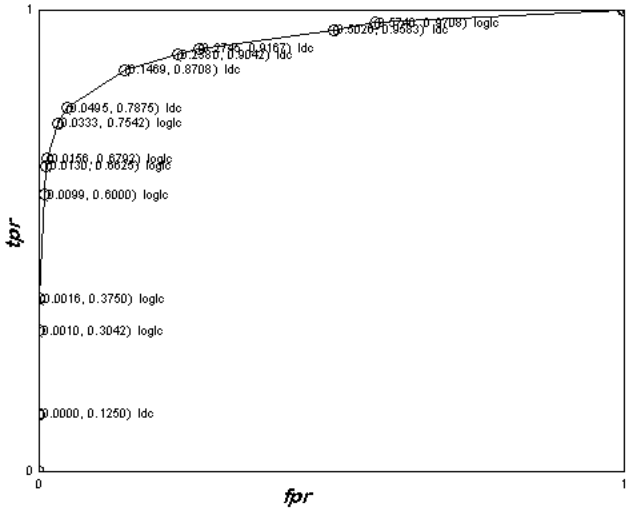


Figura 4.4: Rilevatori di noduli che danno un contributo al ROC convex hull

hull complessivo di entrambe.

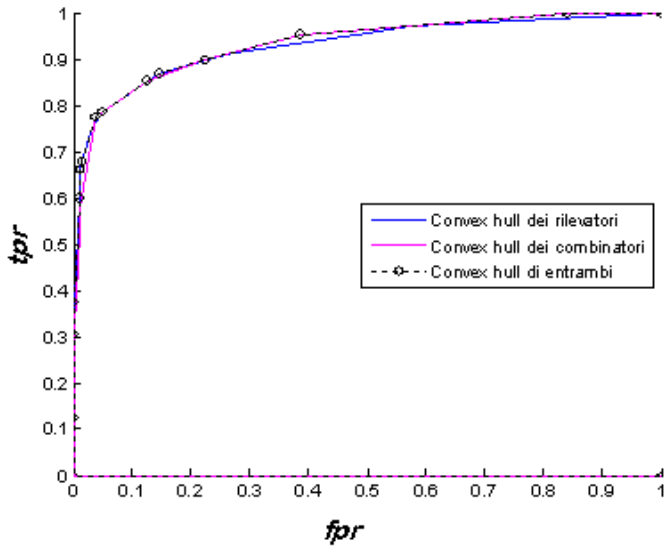


Figura 4.5: ROC convex hull complessivo dei rilevatori e dei combinatori

Bibliografia

[1] F. Provost, T. Fawcett, “Robust classification for imprecise environments”, Machine Learning, vol. 42, n. 3, 2001, p. 203-231.

Capitolo 5

Conclusioni

Vista la forte incidenza del tumore al polmone fra le cause di morte per neoplasia al mondo, e data l'alta percentuale di mortalità per questa patologia se non diagnosticata quando è ancora operabile, diventa necessario un adeguato programma di screening della popolazione a rischio.

A tale scopo sono di notevole importanza le moderne tecniche diagnostiche come la *Tomografia Assiale Computerizzata a spirale*, la quale permette una acquisizione volumetrica ad alta definizione della regione polmonare. Questa tecnica produce un numero molto elevato di immagini, o "fette", ad alta risoluzione che permettono al radiologo di individuare anche le lesioni più piccole, ma rendono il lavoro di analisi molto più lungo e faticoso.

Poichè la mole di dati da analizzare è grande, ed è necessario che non si compiano errori nella diagnosi delle lesioni maligne, sono stati progettati dei sistemi automatici di ausilio nell'individuazione e nella diagnosi dei noduli polmonari, a partire dalle immagini prodotte dalla TAC.

Tali strumenti sono chiamati *sistemi C.A.D. polmonari*, dove l'acronimo CAD assume il significato di Computer Aided Detection, se il sistema è volto alla sola individuazione dei noduli, oppure di Computer Aided Diagnosis, se il sistema compie anche una discriminazione sulla benignità e/o malignità delle lesioni individuate.

In questo settore la grande innovazione consiste nel passare da una diagnosi assistita al computer ad una *computerised diagnosis*, ovvero una diagnosi computerizzata.

Nel dipartimento di Ingegneria Informatica dell'Università di Pisa è stato realizzato un prototipo di sistema CAD polmonare per la *computerised diagnosis,* il quale prevede le seguenti funzioni:

1. Estrazione delle regioni polmonari.
2. Rilevazione delle *regioni di interesse* (ROI).
3. Classificazione delle ROI in *noduli/vasi sanguigni.*
4. Classificazione dei noduli in *maligni/benigni.*

Nell'ambito di tale complesso progetto, sono stati dati contributi in due aree: il *clustering robusto*, e sugli *indici di performace delle curve ROC* ottenute nei passi 2, 3 e 4 del sistema CAD.

Nello specifico, in questo lavoro di tesi, si è andati ad analizzare la tecnica del Fuzzy C-Means robusto descritta da Pham. Questa tecnica, pur non essendo tra le più raffinate, ha dimostrato di avere caratteristiche peculiari particolarmente adatte all'applicazione nella segmentazione delle immagini. In particolare abbiamo descritto le proprietà di questo algoritmo e ne abbiamo mostrato un'esempio di applicazione su un'immagine reale proveniente da una scansione TAC.

Dopodichè abbiamo trattato l'analisi delle performance dei classificatori automatici. Si è introdotta e descritta la metodologia e gli strumenti della *Receiver Operating Characteristic (ROC) Analysis*, con particolare attenzione alla metrica dell'*Area Under the ROC Curve (AUC).* Particolare attenzione è stata rivolta alle mancanze e ai difetti dell'AUC emersi da questo particolare ambito di applicazione. A tale scopo abbiamo ideato e analizzato una estensione di questa metrica prendendo spunto dagli indici di prestazione utilizzati nel campo del Controllo Automatico per la progettazione dei controllori ottimi. A partire da tale metrica, da noi chiamata *tAUC*, abbiamo esteso la ROC Analysis tradizionale ad una nuova metodologia denominata *tROC Analysis.* Per completare la trattazione, abbiamo applicato la nuova metodologia e la ROC Analysis ad un dataset reale, contenente i dati estratti dal sistema CAD. La nuova tecnica si è rivelata efficace sia come strumento di analisi a se stante, sia come utile ausilio da affiancare alla tecnica ROC tradizionale nella soluzione di quelle situazioni di pareggio, in cui l'AUC non permette di discriminare efficacemente quale classificatore selezionare.

Visti i risultati promettenti della tROC Analysis, potrebbe essere interessante approfondire le proprietà di questa tecnica al fine di perfezionarla e renderla uno strumento completo e robusto da affiancare agli strumenti tradizionali di analisi delle performance quali la ROC Analysis.

Appendice A

RFCM - Codice Matlab

A.1 rfcm.m

```
function [center, U, obj_fcn] = rfcm(data, cluster_n, beta, L, H, W,
flag_3d, P, options)

if nargin < 6 | nargin > 9,

error('Too many or too few input arguments!');

end

if nargin < 7,

flag_3d = 0;

end

data_n = size(data, 1);

in_n = size(data, 2);

% Change the following to set default options

default_options = [2; % exponent for the partition matrix U

100; % max.  number of iteration

1e-5; % min.  amount of improvement

1]; % info display during iteration

if (nargin < 9),

options = default_options;
```

```
else
% If "options" is not fully specified, pad it with default values.
if length(options) < 4,
tmp = default_options;
tmp(1:length(options)) = options;
options = tmp;
end
% If some entries of "options" are nan's, replace them with defaults.
nan_index = find(isnan(options)==1);
options(nan_index) = default_options(nan_index);
if options(1) <= 1,
error('The exponent should be greater than 1!');
end
end
expo = options(1); % Exponent for U
max_iter = options(2); % Max.  iteration
min_impro = options(3); % Min.  improvement
display = options(4); % Display info or not
obj_fcn = zeros(max_iter, 1); % Array for objective function
U = initfcm(cluster_n, data_n); % Initial fuzzy partition
if display,
fprintf('Initializing Neighbourhood...\t');
end
if flag_3d,
N = init_neigh(size(U,2), L, H, W, flag_3d, P); %Precalculate Neighbourhood
of each point
else
N = init_neigh(size(U,2), L, H, W); %Precalculate Neighbourhood of
each point
end
```

```
% Main loop

for i = 1:max_iter,

if flag_3d,

[U, center, obj_fcn(i)] = steprfcm(data, U, cluster_n, expo, beta,
H, W, N, L, flag_3d, P);

else

[U, center, obj_fcn(i)] = steprfcm(data, U, cluster_n, expo, beta,
H, W, N, L);

end

if display,

fprintf('Iteration count = %d, obj.  fcn = %f\n', i, obj_fcn(i));

end

% check termination condition

if i > 1,

if abs(obj_fcn(i) - obj_fcn(i-1)) < min_impro, break; end,

end

end

iter_n = i; % Actual number of iterations

obj_fcn(iter_n+1:max_iter) = [];
```

A.2 neigh.m

```
function out = neigh(j, L, H, W, flag_3d, P)

%NEIGH(j, L, H, W, flag_3d, P)

% calcola gli indici dei punti dell'immagine che appartengono al vicinato

% del punto j

%

% j - indice del punto di cui si calcola il vicinato

% L - raggio del vicinato, il vicinato è un quadrato[cubo] di (2L+1)x(2L+1)[x(2L+1)]
punti centrato sul punto j
```

```
% H - altezza in pixel dell'immagine (verticale) -> y
% W - larghezza in pixel dell'immagine (orizzontale) -> x
% P - profonditâ dell'immagine
if nargin < 5,
flag_3d = 0;
else
flag_3d = 1;
end
if flag_3d,
[y, x, z] = ind2sub([H,W,P], j);
else
[y, x] = ind2sub([H,W], j);
end
if (x - L) <= 0
x1 = 1;
else
x1 = x - L;
end
if (x + L) > W
x2 = W;
else
x2 = x + L;
end
if (y - L) <= 0
y1 = 1;
else
y1 = y - L;
end
if (y + L) > H
```

```
y2 = H;
else
y2 = y + L;
end
Nx = [x1:x2];
Ny = [y1:y2];
if flag_3d,
if (z - L) <= 0
z1 = 1;
else
z1 = z - L;
end
if (z + L) > P
z2 = P;
else
z2 = z + L;
end
Nz = [z1:z2];
end
if flag_3d,
%t = 1;
%for i=1:length(Nx),
% for k=1:length(Ny),
% for v = 1 :  length(Nz),
% if sub2ind( [H,W,P], Ny(k), Nx(i), Nz(v) ) ~= j,
% out( t ) = sub2ind( [H,W,P], Ny(k), Nx(i), Nz(v));
% t=t+1;
% end
% end
```

```
% end
%end
t = ones(length(Nx)*length(Nz),1)*Ny;
r=t(:)';
s = ones(length(Nz),1)*Nx;
c=[];
%for i=1:length(Ny),
% c = horzcat(c,s(:)');
%end
c = repmat( s(:)', 1, length(Ny) );
p=[];
%for i=1:(length(Nx)*length(Ny)),
% p = horzcat(p,Nz);
%end
p = repmat(Nz, length(Nx)*length(Ny));
i = sub2ind( [H,W,P], r, c, p);
k = find( i == j );
out = i( [1:(k-1), (k+1):length(i)] );
else
%t = 1;
%for i=1:length(Nx),
% for k=1:length(Ny),
% if sub2ind( [H,W], Ny(k), Nx(i) ) ~= j,
% out( t ) = sub2ind( [H,W], Ny(k), Nx(i) );
% t=t+1;
% end
% end
%end
t = ones(length(Nx),1)*Ny;
```

```
r=t(:)';

c=[];

%for i=1:length(Ny),

% c = horzcat(c,Nx);

%end

c = repmat( Nx, 1, length(Ny) );

i = sub2ind( [H,W], r, c);

k = find( i == j );

out = i([1:(k-1), (k+1):length(i)]);

end

end
```

A.3 steprfcm.m

```
function [U_new, center, obj_fcn] = steprfcm(data, U, cluster_n, expo,
beta, H, W, N, L, flag_3d, P)

%STEPRFCM One step in robust fuzzy c-mean clustering.

% [U_NEW, CENTER, ERR] = STEPFCM(DATA, U, CLUSTER_N, EXPO, beta, H,
W, N, L, flag_3d, P)

% performs one iteration of fuzzy c-mean clustering, where

%

% DATA: matrix of data to be clustered.  (Each row is a data point.)

% U: partition matrix.  (U(i,j) is the MF value of data j in cluster
j.)

% CLUSTER_N: number of clusters.

% EXPO: exponent (> 1) for the partition matrix.

% BETA: penalty coefficent

% H: image height

% W: image width

% N: cell array with neighbourhood of each point
```

```
% L: neighbourhood radius
% flag_3d:  1 if image is 3D, 0 otherwise
% P: image depth
% U_NEW: new partition matrix.
% CENTER: center of clusters.  (Each row is a center.)
% ERR: objective function for partition U.
%
% Note that the situation of "singularity" (one of the data points
is
% exactly the same as one of the cluster centers) is not checked.
% However, it hardly occurs in practice.
%
% See also DISTFCM, INITFCM, IRISFCM, FCMDEMO, FCM.
% Francesco Marafini, 04-17-06.
if (nargin < 10),
flag_3d = 0;
else
flag_3d = 1;
end
mf = U.^expo; % MF matrix after exponential modification
center = mf*data./((ones(size(data, 2), 1)*sum(mf'))'); % new center
dist = distfcm(center, data); % fill the distance matrix
if flag_3d,
pen = penalty(mf, N, H, W, L, flag_3d, P); % fill the penalty matrix
else
pen = penalty(mf, N, H, W, L); % fill the penalty matrix
end
obj_fcn = sum(sum((dist.^2).*mf)) + (beta/2)*sum(sum(mf.*pen)); % objective
function
tmp = ((dist.^2) + beta*pen).^(-1/(expo-1)); % calculate new U, suppose
expo != 1
U_new = tmp./(ones(cluster_n, 1)*sum(tmp));
```

A.4 init_neigh.m

```
function N = init_neigh(nelem, L, H, W, flag_3d, P)
% INIT_NEIGH(nelem, L, H, W, flag_3d, P)
% Precalcola il vicinato di ciascun punto dell'immagine
% nelem - numero dei punti
% L - raggio del vicinato
% H - altezza dell'immagine
% W - larghezza dell'immagine
% flag_3d - flag che indica se l'immagine è tridimensionale
% P - profondità dell'immagine
%
if nargin < 5,
flag_3d = 0;
else
flag_3d = 1;
end
a = 1:(nelem/10):nelem;
b = numel(find(1 >= a));
for j = 1:nelem,
if j == 1,
fprintf('[');
end
c = numel(find( j >= a));
if flag_3d,
N{j} = neigh(j,L,H,W,flag_3d,P);
else
N{j} = neigh(j,L,H,W);
end
if c > b,
```

```
fprintf('.');
end
if j == nelem,
fprintf(']\n');
end
b = c;
end
end
```

A.5 penalty.m

```
function out = penalty(mf, N, H, W, L, flag_3d, P)
%PENALTY(mf, N, H, W, flag_3d, P)
% Penalty measure in robust fuzzy c-mean clustering.
% returns a
% penalty matrix OUT of size M by N, where M and N are
% dimensions of MF, respectively, and OUT(I, J) is
% penalty of MF(I, J).
if nargin < 6,
flag_3d = 0;
else
flag_3d = 1;
end
out = zeros(size(mf, 1), size(mf, 2));
% fill the output matrix
for i = 1:size(mf, 1),
for j = 1:size(mf,2),
out(i, j) = sum(sum(mf([1:(i-1),(i+1):(size(mf, 1))],N{j})'));
end
end
end
```