

*A Luciano e Angela*



# Introduzione

I sistemi di recupero di immagini fanno parte di un settore di ricerca molto attivo già dagli ultimi decenni del '900, conosciuto con il nome di *image retrieval*.

In particolare, a partire dal 1990 si è iniziato a dare sviluppo ai cosiddetti sistemi CBIR, sistemi di recupero di immagini in grado di lavorare su grandi raccolte perché basati sul contenuto visivo delle immagini stesse .

Il melanoma cutaneo è un tumore maligno della pelle che rientra, ad oggi, tra le patologie oncologiche più diffuse in Italia. Le probabilità di sopravvivenza sono molto elevate nei casi di diagnosi precoce, ma purtroppo, statisticamente, l'accuratezza della diagnosi è piuttosto bassa tra i medici generici e addirittura tra i dermatologi privi di esperienza specifica. Per questi motivi l'analisi automatica e computerizzata può rivelarsi un valido strumento di supporto.

Da qualche anno è in fase di sviluppo un progetto a cura del Dott. Ivan Tomba e dell'Ing. Andrea Visotti, con la supervisione del Prof. Massimo Ferri e la collaborazione del Dott. Ignazio Stanganelli, per la creazione di un sistema CBIR per il recupero di immagini dermatologiche di nevi e melanomi.

In questa tesi viene effettuata una valutazione di questo sistema di recupero con lo scopo di determinare dei parametri che permettano di ottimizzarlo quanto più possibile.

Nel primo capitolo si introducono i concetti fondamentali relativi ai sistemi di recupero e alla disciplina dell'*information retrieval*, concentrando in

particolare l'attenzione sul successivo sviluppo che si è avuto nel campo delle immagini fino ad arrivare ai sistemi CBIR.

Nel secondo capitolo si riportano i metodi di valutazione più utilizzati per i sistemi di recupero di immagini CBIR.

Nel terzo capitolo viene illustrato il melanoma cutaneo. Si riferisce di una prima esperienza nello sviluppo di sistemi automatici per il riconoscimento dei melanomi concretizzatasi nei primi anni 2000 nel progetto europeo ADAM e si descrivere l'attuale sistema di recupero sui cui si lavorerà in questa tesi.

Nel quarto capitolo si parla del problema dei minimi quadrati e della sua risoluzione.

Nel quinto capitolo vengono illustrati i modelli proposti, grazie alla collaborazione con il Prof. Massimo Ferri, la Prof.ssa Elena Loli Piccolomini e il Dott. Ivan Tomba, per effettuare la stima dei parametri.

Nel sesto ed ultimo capitolo si riporta la fase sperimentale e la valutazione dei risultati ottenuti.

# Indice

<b>Introduzione</b>	<b>i</b>
<b>1 Sistemi di recupero</b>	<b>1</b>
1.1 Cos'è un motore di ricerca . . . . .	1
1.2 Recupero di informazioni . . . . .	2
1.3 Recupero di immagini . . . . .	3
1.3.1 Recupero di immagini basato sul testo . . . . .	3
1.3.2 Recupero di immagini basato sul contenuto (CBIR) . . . . .	4
<b>2 Valutazione di un sistema di recupero</b>	<b>5</b>
2.1 Valutazione di un sistema CBIR . . . . .	5
<b>3 Analisi automatica di immagini dermatologiche</b>	<b>9</b>
3.1 Il melanoma cutaneo . . . . .	9
3.2 Progetto ADAM . . . . .	13
3.3 Progetto attuale . . . . .	15
<b>4 Problema dei minimi quadrati</b>	<b>19</b>
4.1 Ottimizzazione . . . . .	19
4.1.1 Ottimizzazione vincolata . . . . .	21
4.1.2 Ottimizzazione non vincolata . . . . .	22
4.2 Problema dei minimi quadrati . . . . .	24
4.2.1 Metodo del gradiente proiettato . . . . .	24
4.2.2 Metodo di Gauss-Newton . . . . .	27

---

4.3	Problema lineare dei minimi quadrati . . . . .	30
<b>5</b>	<b>Stima dei parametri</b>	<b>33</b>
5.1	Obiettivo di questo lavoro . . . . .	33
5.2	Modello vettoriale . . . . .	35
5.3	Modello matriciale . . . . .	40
<b>6</b>	<b>Sperimentazione e valutazione risultati</b>	<b>43</b>
6.1	Metodi di risoluzione . . . . .	43
6.2	Metodi di valutazione . . . . .	46
6.3	Database 107 . . . . .	49
6.3.1	Matrici 107 4x - 5 funzioni misuranti . . . . .	52
6.3.2	Matrici 107 8x - 5 funzioni misuranti . . . . .	56
6.3.3	Matrici 107 - 9 descrittori . . . . .	60
6.4	Osservazioni . . . . .	64
	<b>Conclusioni</b>	<b>67</b>
<b>A</b>	<b>Omologia persistente</b>	<b>69</b>
A.1	Distanze di taglia . . . . .	70
A.2	Funzioni di taglia . . . . .	71
A.3	Punti angolari e distanza bottleneck . . . . .	73
<b>B</b>	<b>Grafici</b>	<b>77</b>
B.1	Matrici 107 4x - 5 funzioni misuranti . . . . .	78
B.2	Matrici 107 8x - 5 funzioni misuranti . . . . .	81
B.3	Matrici 107 - 9 descrittori . . . . .	84
<b>C</b>	<b>Codici MATLAB</b>	<b>87</b>
	<b>Bibliografia</b>	<b>99</b>

# Elenco delle figure

B.1	Matrici 107 4x - mod.vettoriale - gradiente_proiettato_01 . . .	78
B.2	Matrici 107 4x - mod.vettoriale - TrovaAlfa . . . . .	78
B.3	Matrici 107 4x - mod.vettoriale - non vincolato . . . . .	79
B.4	Matrici 107 4x - mod.matriciale - gradiente_proiettato_01 . . .	79
B.5	Matrici 107 4x - mod.matriciale - TrovaAlfa . . . . .	80
B.6	Matrici 107 4x - mod.matriciale - non vincolato . . . . .	80
B.7	Matrici 107 8x - mod.vettoriale - gradiente_proiettato_01 . . .	81
B.8	Matrici 107 8x - mod.vettoriale - TrovaAlfa . . . . .	81
B.9	Matrici 107 8x - mod.vettoriale - non vincolato . . . . .	82
B.10	Matrici 107 8x - mod.matriciale - gradiente_proiettato_01 . . .	82
B.11	Matrici 107 8x - mod.matriciale - TrovaAlfa . . . . .	83
B.12	Matrici 107 8x - mod.matriciale - non vincolato . . . . .	83
B.13	Matrici 107 9desc - mod.vettoriale - gradiente_proiettato_01 .	84
B.14	Matrici 107 9desc - mod.vettoriale - TrovaAlfa . . . . .	84
B.15	Matrici 107 9desc - mod.vettoriale - non vincolato . . . . .	85
B.16	Matrici 107 9desc - mod.matriciale - gradiente_proiettato_01 .	85
B.17	Matrici 107 9desc - mod.matriciale - TrovaAlfa . . . . .	86
B.18	Matrici 107 9desc - mod.matriciale - non vincolato . . . . .	86



# Elenco delle tabelle

6.1	Matrici 107 4x - $\alpha_k = \frac{1}{5}$ . . . . .	52
6.2	Matrici 107 4x - modello vettoriale . . . . .	53
6.3	Matrici 107 4x - modello vettoriale - percentuali . . . . .	53
6.4	Matrici 107 4x - modello vettoriale - precision e recall . . . . .	53
6.5	Matrici 107 4x - modello matriciale . . . . .	55
6.6	Matrici 107 4x - modello matriciale - percentuali . . . . .	55
6.7	Matrici 107 4x - modello matriciale - precision e recall . . . . .	55
6.8	Matrici 107 8x - $\alpha_k = \frac{1}{5}$ . . . . .	56
6.9	Matrici 107 8x - modello vettoriale . . . . .	57
6.10	Matrici 107 8x - modello vettoriale - percentuali . . . . .	57
6.11	Matrici 107 8x - modello vettoriale - precision e recall . . . . .	57
6.12	Matrici 107 8x - modello matriciale . . . . .	59
6.13	Matrici 107 8x - modello matriciale - percentuali . . . . .	59
6.14	Matrici 107 8x - modello matriciale - precision e recall . . . . .	59
6.15	Matrici 107 9desc - $\alpha_k = \frac{1}{9}$ . . . . .	60
6.16	Matrici 107 9desc - modello vettoriale . . . . .	61
6.17	Matrici 107 9desc - modello vettoriale - percentuali . . . . .	61
6.18	Matrici 107 9desc - modello vettoriale - precision e recall . . . . .	62
6.19	Matrici 107 9desc - modello matriciale . . . . .	63
6.20	Matrici 107 9desc - modello matriciale - percentuali . . . . .	63
6.21	Matrici 107 9desc - modello matriciale - precision e recall . . . . .	64



# Capitolo 1

## Sistemi di recupero

### 1.1 Cos'è un motore di ricerca

Un motore di ricerca è un sistema automatico che, data una determinata chiave di ricerca, analizza un insieme di dati e restituisce un indice dei contenuti disponibili, classificati in base a formule matematiche che ne indicano il grado di rilevanza.

Uno dei campi in cui i motori di ricerca trovano maggiore utilizzo è il web. Infatti, il principale problema di un utente della rete è trovare, in maniera efficace ed efficiente, le informazioni di cui ha bisogno all'interno di un immenso oceano di notizie e di dati. Di conseguenza, per cercare di rispondere a questa esigenza, sono stati creati i grandi motori di ricerca sul web, come Google (il più utilizzato su scala mondiale), Bing e Yahoo, grandi archivi di dati che contengono informazioni dettagliate su un gran numero di siti.

In generale, i motori di ricerca consentono di inserire in un'apposita maschera parole o frasi per definire l'argomento che l'oggetto cercato (ovvero l'oggetto di interesse dell'utente) dovrà trattare. La ricerca viene poi effettuata all'interno del database del motore, creato e catalogato seguendo particolari algoritmi che cambiano a seconda del motore di ricerca.

Il lavoro di un motore di ricerca è strettamente collegato al concetto

di rilevanza. Infatti, rispondere alla richiesta di un utente porta con sé la necessità di elencare gli elementi del database in ordine di rilevanza rispetto alla richiesta ricevuta. In questo modo il motore di ricerca sarà in grado di fornire una lista di documenti, presentando in primo luogo quelli che meglio dovrebbero rispondere alle esigenze dell'utente, ovvero i più rilevanti.

## 1.2 Recupero di informazioni

Nella sezione precedente si è detto che i motori di ricerca sono sistemi automatici largamente utilizzati sul web. Quello che molti però non sanno è che essi rappresentano solo la punta dell'iceberg di una disciplina fondata alla fine degli anni sessanta, *l'information retrieval (IR)*(recupero di informazioni).

L'information retrieval è, per definizione, la disciplina informatica che si occupa della memorizzazione e del reperimento di documenti; il suo obiettivo è la realizzazione di sistemi software che permettano la memorizzazione di ingenti quantità di documenti in un archivio, in modo tale da permettere poi un efficiente reperimento di quelli più rilevanti alle necessità informative degli utenti [4]. Molte università e biblioteche pubbliche utilizzano tutt'oggi sistemi di information retrieval per fornire l'accesso a pubblicazioni, libri ed altri documenti.

Concetti fondamentali in un sistema IR sono la domanda, più conosciuta come *query*, e il meccanismo del confronto. La query è un insieme di parole chiave rappresentanti l'informazione richiesta. Vengono digitate dall'utente all'interno del sistema di recupero e rappresentano la concretizzazione di quello che è il suo reale bisogno informativo. Il meccanismo del confronto rappresenta invece il modo con cui si confrontano le rappresentazioni dei documenti archiviati e la query.

Il concetto però più importante, legato al funzionamento di un sistema di information retrieval, è la definizione di documento rilevante e non rilevante per una data query. Infatti, a partire da questa definizione si è successivamente in grado di valutare la validità di un sistema IR, in quanto chiaramente

è sempre auspicabile che i documenti recuperati siano i più rilevanti per la richiesta ricevuta.

Questa definizione è importante a tal punto che in essa risiedono anche i principali problemi legati ad un sistema di questo tipo, ovvero la soggettività del concetto di rilevanza e la conseguente incertezza sulla correttezza e sulla valutazione del risultato del recupero.

## 1.3 Recupero di immagini

Un sistema di recupero di immagini, più comunemente conosciuto con il nome di sistema di *image retrieval*, è un sistema informatico per la ricerca e il recupero di immagini digitali da un ampio database.

L'*image retrieval* è un settore di ricerca molto attivo che si è sviluppato a partire dal 1970, grazie alla spinta di due importanti comunità di ricerca, una basata sullo studio del recupero di immagini a partire dal testo (*text-based*) e l'altra basata invece sullo studio dell'immagine vera e propria (*content-based*) [1, 2, 3].

### 1.3.1 Recupero di immagini basato sul testo

Il recupero di immagini basato sul testo (*text-based image retrieval*) risale proprio al 1970. Si tratta di un sistema di recupero molto comune e tradizionale che consiste nell'annotare le immagini con delle parti testuali, come didascalie, parole chiave e descrizioni, e utilizzare poi sistemi di gestione di dati basati sul testo per eseguire il recupero di un'immagine, sfruttando quindi in un certo senso le conoscenze già acquisite grazie all'*information retrieval*.

Nel corso degli anni, molti progressi sono stati compiuti in questa direzione di ricerca. Tuttavia, esistono due grandi difficoltà connesse a questo modello, che si manifestano soprattutto quando la dimensione del database di immagini aumenta notevolmente (decine o centinaia di migliaia). Una è la grande quantità di lavoro umano necessario per l'annotazione manuale di

tutte le immagini. L'altra, che è più sostanziale, è legata al contenuto delle immagini e alla soggettività della percezione umana. Ovvero, persone diverse possono percepire in modo diverso il contenuto della stessa immagine e riportare quindi annotazioni differenti, che possono successivamente influenzare in maniera notevole il processo di recupero.

### 1.3.2 Recupero di immagini basato sul contenuto (CBIR)

Nei primi anni '90, a causa della comparsa di grandi raccolte di immagini, le due difficoltà illustrate sopra relative all'utilizzo di annotazioni manuali divennero sempre più evidenti. Per superarle, fu proposto il recupero di immagini basato sul contenuto (*content-based image retrieval*)(CBIR). Cioè, invece di essere annotate manualmente con parole chiave, le immagini vengono indicizzate a partire dal loro contenuto visivo, ovvero da qualunque informazione che possa essere estrapolata dall'immagine stessa.

Il termine *content-based* significa, infatti, che la ricerca analizza il contenuto dell'immagine invece che le parole chiave o le descrizioni associate, dove il contenuto in questo contesto si riferisce ai colori, alla forma, o alla struttura. Per questo motivo, in questo metodo di recupero immagini, gli utenti forniscono in input al sistema immagini o figure abbozzate.

Dopo il 1990, sono stati fatti molti passi in questa direzione di ricerca e sono stati sviluppati molti sistemi di recupero CBIR, anche a livello commerciale. Tuttavia, ci sono ancora problemi aperti da risolvere come, ad esempio, la definizione di successo o insuccesso del retrieval, che rappresenta una delle sfide più complesse nell'utilizzo di un sistema CBIR.

Un vero e proprio esempio di sistema di questo tipo si può trovare nei sistemi automatici di riconoscimento facciale che usano le forze di polizia. Infatti, un'immagine catturata da una telecamera viene confrontata con l'intero database a disposizione della polizia per cercare di trovare le immagini strettamente più simili ad essa.

## Capitolo 2

# Valutazione di un sistema di recupero

La valutazione delle prestazioni di un sistema di recupero è da sempre un problema cruciale oggetto di studio di molti ricercatori. Già nel 1950 infatti, coloro che si occupavano di information retrieval iniziarono a discutere su questo argomento.

Dato lo sviluppo temporalmente più recente dei sistemi CBIR, risulta quindi evidente che i ricercatori in materia di information retrieval possiedono decenni di vantaggio su questo problema. Per questo motivo, nonostante la differenza tra i due sistemi, molte delle misure utilizzate per valutare un sistema CBIR provengono dall'information retrieval.

Di seguito riportiamo i metodi maggiormente utilizzati per la valutazione di un sistema CBIR [5].

### 2.1 Valutazione di un sistema CBIR

#### Precision e Recall

*Precision e Recall* (precisione e richiamo) sono misure standard utilizzate nella valutazione dell'information retrieval, sempre più utilizzate anche per la

valutazione di un sistema CBIR. In particolare, ciò che viene maggiormente studiato è il *grafico di precision e recall*, detto *grafico PR*.

Data un'immagine come query e definito un giudizio di rilevanza o non rilevanza rispetto alla query per tutti gli elementi del database su cui si vuole valutare il sistema, si definisce *precision* (precisione) il rapporto tra il numero di elementi recuperati rilevanti e il numero totale di elementi recuperati, e *recall* (richiamo) il rapporto tra il numero di elementi rilevanti recuperati e il numero totale di elementi rilevanti presenti nel database, ovvero

$$\begin{aligned} \textit{precision} &= \frac{N^\circ \textit{ immagini rilevanti recuperate}}{N^\circ \textit{ immagini recuperate}} \\ \textit{recall} &= \frac{N^\circ \textit{ immagini rilevanti recuperate}}{N^\circ \textit{ immagini rilevanti nel database}} \end{aligned}$$

Queste due misure di valutazione diventano particolarmente interessanti se studiate come funzione delle prime  $k$  immagini recuperate, in quanto questo permette poi di passare alla rappresentazione grafica. In particolare, indicando con  $P$  la funzione relativa a precision e con  $R$  la funzione relativa a recall, si ottengono le seguenti definizioni [6]:

$$P(k) = \frac{NR(k)}{k} \quad R(k) = \frac{NR(k)}{N_{rel}} \quad (2.1)$$

dove  $k$  è il numero di immagini recuperate,  $N_{rel}$  è il numero di immagini rilevanti presenti nel database e  $NR(k)$  è il numero di immagini rilevanti tra le prime  $k$  recuperate. Chiaramente  $P(N_{rel}) = R(N_{rel})$ .

Il grafico PR si ottiene rappresentando in ascissa i valori della funzione recall e in ordinata i valori della funzione precision. Tipicamente un buon sistema di recupero corrisponde a un grafico decrescente.

Le funzioni relative a precision e recall possono essere rappresentate graficamente anche in modo separato, riportando sull'asse delle ascisse il numero di elementi recuperati. In questo caso, il grafico relativo a precision dovrebbe essere decrescente, mentre il grafico relativo a recall crescente, in quanto recuperando poche immagini ci si aspetta un valore alto di precision, mentre

recuperando tutte le immagini presenti nel database si ottiene un valore di recall uguale a uno.

L'analisi di questi grafici può essere usata anche per valutare quale potrebbe essere un buon numero di immagini da recuperare.

Oltre all'analisi dei grafici, si tende spesso a valutare coppie specifiche di valori di precision e recall, come ad esempio il valore di precision corrispondente ad un fissato valore di recall o i valori di precision e recall ad un fissato numero di immagini recuperate. Questo tipo di analisi "in coppia" è particolarmente importante in quanto, per le definizioni date di precision e recall, l'analisi di un singolo parametro, considerato in modo indipendente dell'altro, potrebbe fornire un'informazione insufficiente al fine di esprimere una valutazione sul sistema di recupero.

### Retrieval Efficiency

Il parametro di valutazione *retrieval efficiency* (efficienza del recupero) è fortemente connesso ai parametri standard precision e recall descritti nel paragrafo precedente. Infatti, data come sempre una query e definito un giudizio di rilevanza o non rilevanza rispetto ad essa per tutti gli elementi del database, se il numero di immagini recuperate è minore o uguale al numero di immagini rilevanti questo valore è uguale a precision, altrimenti è uguale a recall. Ovvero, utilizzando la stessa notazione introdotta in (2.1) si ottiene:

$$retrieval\ efficiency(k) = \begin{cases} \frac{NR(k)}{k} & \text{se } k \leq N_{rel} \\ \frac{NR(k)}{N_{rel}} & \text{altrimenti} \end{cases} \quad (2.2)$$

### Error Rate

Il parametro di valutazione *error rate* (tasso di errore) è strettamente connesso a precision, in quanto può essere interpretato come il suo complementare. Infatti, data come sempre una query e definito un giudizio di rilevanza o non rilevanza rispetto ad essa per tutti gli elementi del database,

si definisce *error rate* il valore seguente:

$$error\ rate = \frac{N^\circ\ immagini\ non\ rilevanti\ recuperate}{N^\circ\ immagini\ recuperate}$$

Chiaramente, anche per questo parametro può essere interessante valutarne il comportamento al variare di  $k$ , dove  $k$  è il numero di immagini recuperate.

### Rank e Normalized Average Rank

Il parametro *rank* (rango) rappresenta la posizione in cui è recuperata la prima immagine rilevante per una data query.

Spesso, questo singolo parametro si rivela però insufficiente al fine di esprimere una valutazione attendibile, in quanto può essere necessario tenere conto anche della dimensione del database e del numero di immagini rilevanti per la query. Per questo motivo, è stato proposto l'utilizzo di un nuovo parametro di valutazione, il *normalized average rank* ( $\widetilde{rank}$ ) (posizione media normalizzata) definito come segue:

$$\widetilde{rank} = \frac{1}{NN_{rel}} \left( \sum_{i=1}^{N_{rel}} R_i - \frac{N_{rel}(N_{rel} + 1)}{2} \right) \quad (2.3)$$

dove  $N$  è il numero degli elementi del database,  $N_{rel}$  è il numero di elementi rilevanti per la query, e  $R_i$  è la posizione (rank) in cui la  $i$ -esima immagine rilevante è recuperata. Un valore di  $\widetilde{rank} = 0$  corrisponde ad una prestazione perfetta da parte del sistema di recupero, mentre un valore prossimo a 1 rappresenta una prestazione pessima.

# Capitolo 3

## Analisi automatica di immagini dermatologiche

### 3.1 Il melanoma cutaneo

Per definizione, un *tumore* o *neoplasia* è una massa anomala di tessuto che si forma in seguito alla proliferazione patologica delle cellule di una zona del corpo. È detto *benigno* quando rimane circoscritto alla sede del suo sviluppo, mentre si definisce *maligno* quando si diffonde rapidamente per invasione dei tessuti circostanti e per metastasi, ovvero tramite cellule maligne che si staccano dal tumore originario e si diffondono in altri organi dove possono riprodursi e generare nuovi tumori.

Il *melanoma cutaneo* è un tumore maligno che origina dai melanociti, cellule che formano la pelle, responsabili della produzione della melanina, un pigmento che protegge dagli effetti dannosi dei raggi solari. In condizioni normali, i melanociti possono dar luogo ad agglomerati scuri visibili sulla superficie della pelle, definiti clinicamente *nevi* e più comunemente conosciuti come nei.

Il melanoma può insorgere su una pelle integra, oppure da nevi preesistenti, che possono essere congeniti (presenti fin dalla nascita o dalla prima infanzia) o acquisiti (comparsi durante il corso della vita).

Il melanoma cutaneo colpisce soprattutto attorno ai 45-50 anni, anche se l'età media alla diagnosi si sta abbassando notevolmente. Infatti, fino a pochi anni fa era considerato una neoplasia rara, o addirittura rarissima fino all'adolescenza, mentre negli ultimi 20 anni l'incidenza è aumentata di oltre il 4% annuo. In Italia i dati AIRTUM (Associazione italiana registri tumori) parlano di circa 13 casi ogni 100000 persone, con una stima che si aggira attorno a 3150 nuovi casi ogni anno tra gli uomini e 2850 tra le donne, con un'incidenza che, nel giro di dieci anni, è addirittura raddoppiata.

### Tipologie

Dal punto di vista clinico si distinguono quattro tipologie principali di melanoma cutaneo:

- melanoma piano a diffusione superficiale;
- melanoma nodulare;
- lentigo maligna melanoma;
- melanoma lentiginoso acrale.

Il *melanoma piano a diffusione superficiale* (SSM) rappresenta circa il 70% di tutti i melanomi ed ha spesso un decorso bifasico. Si manifesta dapprima con una lesione maculare che successivamente può evolvere in placca, con aree policromatiche o talvolta chiare. Tale andamento clinico corrisponde ad una fase di crescita orizzontale, che evolve stabilmente in un tempo generalmente lento, seguita da una fase di crescita verticale, che corrisponde ad un'invasione in profondità.

Il *melanoma nodulare* si presenta spesso con una lesione di colorito nero-marrone, talvolta sanguinolenta. Esso è caratterizzato sin dall'inizio da una fase di crescita verticale, molto aggressiva, accompagnata da una fase orizzontale scarsa o del tutto assente, e per questo è molto più aggressivo del precedente.

Il *lentigo maligna melanoma* è una variante di melanoma che deriva dal risultato di una mutazione di un melanoma in situ che tende a comparire in aree fotoesposte e danneggiate da una esposizione solare cronica, come ad esempio il volto. Colpisce più frequentemente gli individui anziani. Clinicamente appare come una chiazza asimmetrica, piana, non palpabile e policromatica (dal marrone al nero). Ha una lenta evoluzione e solo tardivamente possono comparire dei noduli sulla superficie che fanno presupporre una chiara trasformazione in un melanoma invasivo.

Il *melanoma lentiginoso acrale* è tipico delle sedi palmo-plantari e subungueali. Il melanoma ungueale rappresenta una sottoclasse meno frequente dei melanomi acrali, più frequente tra i 50-70 anni. Queste tipologie di melanoma hanno usualmente una fase intraepidermica rapida. Anche in questo caso, la comparsa di una regione nodulare riflette l'inizio di una crescita verticale del tumore. Pertanto, anche in questo caso, è generalmente rispettato un andamento bifasico come nel caso del melanoma superficiale.

### Fattori di rischio

Il rischio di sviluppare un melanoma è legato sia a fattori di carattere ambientale (esogeni), sia a fattori di tipo genetico legati all'individuo (endogeni). Tra quelli esogeni, il principale fattore di rischio è rappresentato dall'eccessiva esposizione alla luce ultravioletta (UV), principalmente rappresentata dai raggi solari, che arriva a noi sotto forma di raggi UVA e UVB. I raggi UVA sono la componente predominante dei raggi solari e costituiscono il 95% dei raggi UV che raggiungono la superficie terrestre. Nonostante la loro energia sia bassa, penetrano nella cute in profondità; inducono l'abbronzatura velocemente, provocano un invecchiamento cutaneo e generano un'ampia serie di reattivi dell'ossigeno che danneggiano il DNA. I raggi UVB costituiscono invece il restante 5% dei raggi che raggiungono la superficie terrestre e inducono infiammazione, apoptosi cellulare, immunosoppressione. La troppa esposizione al sole o ad altre sorgenti di raggi UV, come lampade e lettini solari, rappresentano quindi un potenziale pericolo perché possono

danneggiare il DNA delle cellule della pelle e innescare la trasformazione tumorale.

Tra i fattori endogeni, uno di quelli che aumenta il rischio di sviluppare un melanoma è l'appartenenza al fototipo 1-2, ovvero carnagione chiara, capelli e occhi chiari, presenza di lentiggini. Infatti, i soggetti con fototipo 1-2 hanno un rischio di melanoma doppio rispetto ai soggetti con fototipo 4, ovvero pelle scura/olivastra, occhi e capelli scuri o neri.

Tra gli altri fattori di tipo genetico è importante sottolineare la familiarità (il 10% dei pazienti affetti da melanoma riferisce almeno un altro caso nell'ambito della propria famiglia), la presenza di un numero significativo di nevi acquisiti o di nevi atipici (diametro superiore a 6 mm, bordo irregolare, colore variegato) e la recidività.

### Sintomi

Il sintomo principale del melanoma cutaneo è il cambiamento nell'aspetto di un nevo o la comparsa di uno nuovo. Le caratteristiche di un nevo che possono indicare segnali dall'allarme per lo sviluppo di un melanoma si trovano nella regola dell' ABCDE:

- **A**simmetria della lesione;
- **B**ordi irregolari e frastagliati;
- **C**olore disomogeneo policromo (nero, rosso-bruno, rosa) o nero molto intenso;
- **D**iametro superiore a 6 mm;
- **E**voluzione progressiva in un tempo piuttosto breve, con cambiamenti cromatici.

La diagnosi di un melanoma risulta agevolata quando sono presenti tutte le caratteristiche. Può invece risultare complessa quando il melanoma è nelle fasi iniziali o quando alcuni caratteri clinici dell' ABCDE vengono riscontrati

in un nevo. La difficoltà della valutazione clinica può provocare sia un alto tasso di escissioni inutili sia una sottostima della diagnosi.

Per questa ragione, al fine di incrementare la sensibilità diagnostica, lo specialista può ricorrere all'utilizzo di strumenti che consentono osservazioni migliori dell'osservazione diretta ad occhio nudo.

### **Diagnosi strumentale**

Uno degli strumenti più utilizzati in prima analisi dai dermatologi è il dermatoscopio manuale, uno strumento portatile dotato di una lente d'ingrandimento fisso a 10x che viene appoggiata sulla cute da esaminare, in modo da poter osservare le strutture anatomiche.

Per rendere più accurato l'esame, si può ricorrere alla tecnica dell'epiluminescenza che consiste nell'applicazione di un mezzo di contrasto che permette l'osservazione della struttura microscopica dei primi strati di pelle sottostanti quelli visibili a occhio nudo.

Per ottenere ingrandimenti fino a 1000x si utilizza invece il videodermatoscopio, uno strumento che utilizza la luce polarizzata, costituito da una fotocamera digitale con fibre ottiche e lenti. La fotocamera è collegata con un cavo ad un computer e questo permette di fotografare ed archiviare le immagini, rendendo così possibile il confronto di lesioni dubbie a distanza di tempo.

## **3.2 Progetto ADAM**

Nei primi anni 2000 il CINECA, Consorzio Interuniversitario del Nord Est Italiano per il Calcolo Automatico, il Dipartimento di Matematica dell'Università di Bologna ed ARCES, l'Istituto Oncologico Romagnolo e la società DSMedica di Milano avevano concordato una ricerca sull'uso di metodi innovativi per la diagnosi automatica di lesioni cutanee pigmentate, avviando il progetto ADAM nell'ambito del 5° Programma Quadro dell'Unione Europea. L'obiettivo era di fornire al dermatologo o al medico di base uno strumento

che producesse in modo oggettivo un indice di rischio di supporto all'esame clinico [10].

L'idea che l'ausilio di un test diagnostico computerizzato potesse rivelarsi estremamente utile trovava le radici in alcune indagini statistiche che avevano rivelato una percentuale di diagnosi corrette effettuate dai dermatologi intorno al 60% e ancora inferiori tra i medici di base.

La tecnica utilizzata dal software di analisi di immagini ADAM consisteva nell'uso di una classe di descrittori di forma geometrici-topologici propri dell'omologia persistente, le *funzioni di taglia* (Appendice A). Una funzione di taglia dipende sempre da una funzione reale, detta *funzione misurante*, scelta in modo da catturare l'aspetto dell'immagine che interessa all'utente. In questo caso, le funzioni misuranti utilizzate riguardavano il bordo, la tessitura e la distribuzione del colore.

Per prima cosa l'immagine di una lesione acquisita dal sistema veniva segmentata, ovvero la parte vera e propria della lesione veniva separata dal resto della cute. Poi si suddivideva l'immagine così ottenuta in due metà e si confrontavano le due metà mediante le funzioni di taglia relative alle tre funzioni misuranti scelte. Si tracciavano poi tre curve che riassumevano i confronti tra le due metà e si calcolavano tre parametri caratteristici. A questo punto una SVM (Macchina a Vettori di Supporto) elaborava i parametri per produrne in output uno solo da confrontare con un valore di soglia supposto in grado di discriminare i nevi dai melanomi. L'obiettivo del sistema era determinare una risposta di alto o basso rischio, o di rischio intermedio.

Sono stati fatti esperimenti su un dataset costituito da 42 melanomi e 435 nevi; 22 melanomi e 218 nevi sorteggiati in modo indipendente costituivano il training set e i restanti nevi e melanomi appartenevano invece al test set. In questa situazione i migliori risultati ottenuti sono stati:

- sensibilità: 80%
- specificità: 79,74%
- accuratezza diagnostica: 79,75%

dove la specificità rappresenta la percentuale di nevi riconosciuti come tali dal sistema, la sensibilità rappresenta il numero di melanomi riconosciuti come tali dal sistema e l'accuratezza diagnostica la percentuale di lesioni correttamente riconosciute.

Nel confronto con il sistema ADAM, dermatologi con un'esperienza di almeno due anni nella diagnosi in epiluminescenza hanno mostrato risultati più bassi in termini di sensibilità e più elevati in specificità, sia sulla diagnosi clinica sia sulla valutazione delle immagini in epiluminescenza.

Da questi risultati il progetto ADAM ha quindi dimostrato di poter fornire un valido supporto oggettivo per il dermatologo lungo il percorso diagnostico delle lesioni pigmentate.

### 3.3 Progetto attuale

Attualmente è in fase di realizzazione e sviluppo un nuovo progetto per la costruzione di una macchina ad alta tecnologia per il supporto medico nell'individuazione di lesioni melanocitiche. Il progetto è a cura del Dott. Ivan Tomba e dell'Ing. Andrea Visotti della ditta CA-MI S.r.l., con la supervisione del Prof. Massimo Ferri, leader del gruppo di ricerca di Matematica della Visione dell'Istituto ARCES dell'Università di Bologna, e la collaborazione del Dott. Ignazio Stanganelli, direttore dell'istituto IRST dell'Ospedale di Forlì-Ravenna.

Questo progetto può essere considerato a tutti gli effetti un sistema CBIR, ovvero un sistema di recupero di immagini a partire dal contenuto. Infatti, l'obiettivo degli inventori è quello di creare una macchina contenente un database di immagini cliniche di nevi e melanomi che, una volta acquisita una nuova immagine di una lesione, recuperi le immagini più "vicine" a quella in esame tra tutte quelle contenute nel database.

È importante sottolineare che non si vuole assolutamente fornire una diagnosi che possa sostituire quella del dermatologo, ma si vuole cercare di agevolare e aiutare il lavoro di quest'ultimo attraverso l'utilizzo del recupero

di immagini. In un certo senso, per cercare una metafora, si vuole dotare il dermatologo di una grandissima enciclopedia di tutte le lesioni viste durante la sua carriera, in modo tale che possa confrontare una nuova lesione con tutte quelle viste nel suo passato, di cui chiaramente conosce già la natura clinica. La macchina si preoccupa di svolgere al suo posto tutti i confronti e gli restituisce direttamente solo le immagini più simili a quella che lui sta analizzando, facendogli quindi risparmiare moltissimo tempo.

Il confronto tra le immagini viene effettuato principalmente sfruttando la teoria dell'omologia persistente (Appendice A). In particolare, due immagini vengono confrontate utilizzando cinque funzioni misuranti: il blu, il rosso, il verde, il bordo e la variazione totale. La distanza tra due immagini rispetto a una funzione misurante viene calcolata come distanza bottleneck tra le due funzione di taglia associate. Infine, la distanza globale tra due immagini viene calcolata come media pesata delle cinque distanze bottleneck.

Più nello specifico, il lavoro svolto dall'algoritmo informatico che sarà contenuto all'interno della macchina può essere descritto attraverso alcune parti principali: depilatore, segmentatore, triangolatore, tagliatore, distanziatore e retrieval.

Dopo essere state acquisite, le immagini vengono prima di tutto private degli eventuali peli presenti attraverso il depilatore, in modo tale da eliminare alcune delle possibili fonti di rumore per i passi successivi. Il segmentatore determina poi il bordo della lesione, in modo da estrapolare la lesione vera e propria su cui andranno a lavorare i passi successivi. Il triangolatore calcola il grafo di taglia, triangolando la figura sulla base della quantità di pixel che si decide di considerare, e calcola le funzioni misuranti sui grafi. Successivamente entrano in gioco il tagliatore, che genera la lista dei punti angolari e il distanziatore che si occupa di calcolare le distanze.

L'ultimo passo, ovvero il retrieval, si differenzia a seconda che la macchina stia lavorando "*real time*" o "*off-line*".

Infatti, nel momento *real time*, cioè nel caso in cui viene inserita una immagine query nel sistema, che viene confrontata attraverso i passi precedenti

con tutte le immagini già presenti nel database, il retrieval restituisce in output le dieci immagini più vicine a quella in esame, ovvero le dieci immagini corrispondenti alle dieci distanze globali più basse.

Il momento *off-line* lavora invece su tutte le immagini contenute nel database ed in particolare prende tutti i valori calcolati attraverso i passi precedenti, che rappresentano tutte le distanze reciproche tra le immagini (confrontate due a due), e li raccoglie creando in output le cosiddette “matrici distanza”, particolari matrici aventi la seguente forma:

$$\begin{pmatrix} 0 & 0 & I_1 & I_2 & \cdots & \cdots & I_N \\ 0 & 0 & 1 & 0 & \cdots & \cdots & 0 \\ I_1 & 1 & 0 & d_{(1,2)} & \cdots & \cdots & d_{(1,N)} \\ I_2 & 0 & d_{(1,2)} & 0 & \cdots & \cdots & d_{(2,N)} \\ \vdots & \vdots & \vdots & & \ddots & & \vdots \\ \vdots & \vdots & \vdots & & & \ddots & \vdots \\ I_N & 0 & d_{(1,N)} & d_{(2,N)} & \cdots & \cdots & 0 \end{pmatrix} \quad (3.1)$$

dove  $N$  rappresenta il numero di immagini presenti nel database,  $I_i$  rappresenta l'immagine  $i$ -esima e  $d_{(i,j)}$  la distanza calcolata tra l'immagine  $i$  e l'immagine  $j$ .

Come si può osservare, si tratta di matrici simmetriche la cui diagonale è composta da tutti zeri, in quanto ogni immagine ha distanza zero da se stessa. Nella prima riga si trovano le sigle o le indicizzazioni relative alle immagini, mentre nella seconda riga le diagnosi corrispondenti, dove con lo 0 si indicano i nevi e con 1 i melanomi.

Il calcolo di queste matrici risulta particolarmente utile anche ai fini di un successivo momento *real time*. Infatti, l'aver già calcolato tutti i vari “dati” relativi alle immagini del database permette al momento *real time* di non dover ripercorrere, per queste immagini, tutti i passi dell'algoritmo, rendendo quindi molto più veloce il suo tempo di esecuzione.

Inoltre, le matrici distanza sono per noi particolarmente interessanti in quanto, a partire da esse, è possibile esprimere una valutazione sul sistema di recupero. Per questo motivo rappresentano il punto di partenza del nostro

lavoro, che verrà descritto nei prossimi capitoli di questa tesi, e che sarà poi inserito all'interno di quello che abbiamo chiamato il momento *off-line*.

# Capitolo 4

## Problema dei minimi quadrati

### Notazione

Introduciamo la seguente notazione che utilizzeremo per tutto il capitolo:

- $x \in \mathbb{R}^n$  : vettore colonna;
- $x_k$  :  $k$ -esimo vettore (o elemento) di una successione;
- $x_i$  :  $i$ -esima componente di un vettore;
- $\nabla f(x)$  : gradiente di una funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  calcolato in  $x$ ;  
è un vettore colonna con componenti  $\frac{\partial f(x)}{\partial x_j}$  per  $j = 1, \dots, n$ ;
- $\nabla^2 f(x)$  : matrice hessiana ( $n \times n$ ) di una funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$   
calcolata in  $x$ , con componenti  $\frac{\partial^2 f(x)}{\partial x_i \partial x_j}$  per  $i, j = 1, \dots, n$ .

### 4.1 Ottimizzazione

L'ottimizzazione è per definizione la minimizzazione o massimizzazione di una funzione sotto eventuali vincoli sulle variabili.

Un problema di ottimizzazione può essere visto come il problema in cui data una funzione  $f$ , detta *funzione obiettivo*, si cerca di determinare un argomento  $x^*$  tale per cui  $f$  raggiunga un minimo in  $x^*$ , ovvero si cerca di determinare un punto di minimo di  $f$  su un dato dominio.

Formalmente, data la funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  e un insieme  $S \subseteq \mathbb{R}^n$ , il problema che si vuole risolvere è

$$\min_{x \in S} f(x) \quad (4.1)$$

ovvero, si vuole determinare, se esiste,

$$x^* = \arg \min_{x \in S} f(x)$$

Il problema della ricerca di un massimo di  $f$  viene visto come il problema della ricerca di un minimo di  $-f$ .

Prima di procedere ricordiamo quanto segue.

**Definizione 4.1.** Sia  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  e sia  $S \subseteq \mathbb{R}^n$ . Un vettore  $x^* \in S$  si dice:

- un punto di *minimo globale* di  $f$  su  $S$  se:

$$f(x^*) \leq f(x) \quad \forall x \in S$$

- un punto di *minimo globale stretto* di  $f$  su  $S$  se:

$$f(x^*) < f(x) \quad \forall x \in S$$

- un punto di *minimo locale* di  $f$  su  $S$  se esiste un intorno  $U(x^*)$  di  $x^*$  tale che:

$$f(x^*) \leq f(x) \quad \forall x \in S \cap U(x^*)$$

- un punto di *minimo locale stretto* di  $f$  su  $S$  se esiste un intorno  $U(x^*)$  di  $x^*$  tale che:

$$f(x^*) < f(x) \quad \forall x \in S \cap U(x^*), x \neq x^*$$

**Proposizione 4.1.1.**

Sia  $S \subseteq \mathbb{R}^n$  un insieme convesso e  $f$  una funzione convessa su  $S$ . Allora ogni punto di minimo locale di  $f$  su  $S$  è anche punto di minimo globale di  $f$  su  $S$ .

### 4.1.1 Ottimizzazione vincolata

Un problema di ottimizzazione vincolata è un problema della forma

$$\begin{aligned} \min \quad & f(x) \\ & x \in S, \quad S \subset \mathbb{R}^n \end{aligned} \tag{4.2}$$

Inoltre, assumeremo sempre che:

- $S$  sia un sottoinsieme non vuoto, convesso e chiuso di  $\mathbb{R}^n$ ;
- la funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  sia continua e differenziabile su  $S$ .

Un vettore  $x$  che soddisfa il vincolo di un dato problema è chiamato *vettore ammissibile* per il problema. L'insieme  $S$  è detto *insieme di ammissibilità*.

**Definizione 4.2.** Sia  $S \subset \mathbb{R}^n$  e sia  $\bar{x} \in S$ . Si dice che un vettore  $d \in \mathbb{R}^n$ ,  $d \neq 0$ , è una *direzione ammissibile* per  $S$  in  $\bar{x}$  se esiste  $\bar{\lambda} > 0$  tale che:

$$\bar{x} + \lambda d \in S, \quad \forall \lambda \in (0, \bar{\lambda})$$

Vale la seguente proposizione per le direzioni ammissibili in un insieme convesso.

**Proposizione 4.1.2.**

*Sia  $S \subset \mathbb{R}^n$  un insieme convesso e sia  $\bar{x}$  un qualsiasi punto di  $S$ . Allora, comunque si fissi  $x \in S$  tale che  $x \neq \bar{x}$ , la direzione  $d = x - \bar{x}$  è una direzione ammissibile per  $S$  in  $\bar{x}$ .*

*Dimostrazione.* Sia  $\bar{x} \in S$ . Comunque sia preso  $x \in S$ , con  $x \neq \bar{x}$ , per la convessità di  $S$  si ha che  $(1 - \beta)\bar{x} + \beta x \in S$  per ogni  $\beta \in [0, 1]$  e quindi  $\bar{x} + \beta(x - \bar{x}) \in S$  per ogni  $\beta \in [0, 1]$ .

Ne viene quindi che la direzione  $d = x - \bar{x}$  è una direzione ammissibile per  $S$  in  $\bar{x}$ .

□

Per le ipotesi fatte su  $S$  ed  $f$  valgono inoltre le seguenti *condizioni di ottimalità*, dove per condizione di ottimalità si intende una condizione affinché un punto  $x^*$  risulti una soluzione (locale o globale) del problema di minimo.

**Teorema 4.1.3.**

1. Sia  $x^* \in S$  un punto di minimo locale di  $f$  su  $S$ . Allora risulta

$$\nabla f(x^*)^T(x - x^*) \geq 0, \quad \forall x \in S \quad (4.3)$$

2. Se  $f$  è convessa su  $S$ , condizione necessaria e sufficiente affinché  $x^*$  sia minimo globale è che risulti

$$\nabla f(x^*)^T(x - x^*) \geq 0, \quad \forall x \in S \quad (4.4)$$

Il lettore interessato alla dimostrazione di questi risultati può fare riferimento a [15].

Un vettore  $x^*$  che soddisfa la condizione di ottimalità (4.3) si dice *punto stazionario*.

### 4.1.2 Ottimizzazione non vincolata

Un problema di ottimizzazione non vincolata è un problema della forma

$$\begin{aligned} \min \quad & f(x) \\ & x \in \mathbb{R}^n \end{aligned} \quad (4.5)$$

dove  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  è una funzione data.

In questo caso vale il seguente risultato che ci tornerà utile nel seguito.

**Definizione 4.3.** Sia  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ . Un vettore  $d \in \mathbb{R}^n$ ,  $d \neq 0$ , si dice *direzione di discesa* per  $f$  in  $x$  se esiste  $\bar{\lambda} > 0$  tale che

$$f(x + \lambda d) < f(x), \quad \forall \lambda \in (0, \bar{\lambda}) \quad (4.6)$$

**Proposizione 4.1.4.**

Sia  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  continuamente differenziabile in un punto  $x \in \mathbb{R}^n$  e sia  $p \in \mathbb{R}^n$  un vettore non nullo. Allora se risulta

$$p^T \nabla f(x) < 0$$

la direzione  $p$  è una direzione di discesa per  $f$  in  $x$ .

Valgono inoltre le seguenti condizioni di ottimalità.

**Teorema 4.1.5** (Condizione necessaria del primo ordine).

Sia  $x^*$  un punto di minimo locale di  $f$  e sia  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  differenziabile con continuità in un intorno aperto di  $x^*$ . Allora:

$$\nabla f(x^*) = 0 \tag{4.7}$$

Un punto  $x^*$  che soddisfa (4.7) è detto *punto stazionario*.

**Teorema 4.1.6** (Condizione necessaria del secondo ordine).

Sia  $x^* \in \mathbb{R}^n$  e sia  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  due volte differenziabile con continuità in un intorno aperto di  $x^*$ . Condizione necessaria affinché  $x^*$  sia un punto di minimo locale di  $f$  è che  $x^*$  sia un punto stazionario di  $f$  e che la matrice hessiana sia semidefinita positiva in  $x^*$ , ovvero:

1.  $\nabla f(x^*) = 0$
2.  $y^T \nabla^2 f(x^*) y \geq 0, \quad \forall y \in \mathbb{R}^n$

**Teorema 4.1.7** (Condizione sufficiente del secondo ordine).

Sia  $x^* \in \mathbb{R}^n$  e sia  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  una funzione due volte differenziabile con continuità in un intorno aperto di  $x^*$ . Se valgono le condizioni

1.  $\nabla f(x^*) = 0$
2. la matrice hessiana è definita positiva in  $x^*$ , ovvero

$$y^T \nabla^2 f(x^*) y > 0 \quad \forall y \in \mathbb{R}^n, y \neq 0$$

allora  $x^*$  è un punto di minimo locale stretto di  $f$ .

Nel caso particolare in cui  $f$  sia convessa le condizioni necessarie del primo ordine diventano anche sufficienti.

**Teorema 4.1.8.**

*Sia  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  differenziabile. Se  $f$  è una funzione convessa, allora  $x^*$  è un punto di minimo globale di  $f$  su  $\mathbb{R}^n$  se e solo se  $\nabla f(x^*) = 0$ .*

Il lettore interessato alla dimostrazione di questi risultati può fare riferimento a [15].

## 4.2 Problema dei minimi quadrati

Il problema dei minimi quadrati è un problema in cui la funzione da minimizzare  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  assume la forma

$$f(x) = \|r(x)\|_2^2 \quad (4.8)$$

dove  $r : \mathbb{R}^n \rightarrow \mathbb{R}^m$  e  $\|\cdot\|_2$  rappresenta la norma euclidea.

In particolare la (4.8) si può riscrivere come

$$f(x) = \sum_{j=1}^m r_j^2(x) \quad (4.9)$$

dove  $r_j : \mathbb{R}^n \rightarrow \mathbb{R}$  è detto *residuo*  $j$ -esimo.

Problemi di questo tipo si presentano tipicamente nei problemi di stima dei parametri di modelli matematici sulla base di osservazioni sperimentali. In questi casi  $x$  è il vettore dei parametri del modello e il termine  $r_j(x)$  esprime lo scostamento tra la previsione fornita dal modello in corrispondenza di  $x$  e il dato sperimentale.

Vediamo ora come si può risolvere un problema di questo tipo sia nel caso vincolato, sia nel caso non vincolato.

### 4.2.1 Metodo del gradiente proiettato

Il metodo del gradiente proiettato è un metodo iterativo, appartenente ai metodi delle direzioni ammissibili, molto utilizzato nella risoluzione di

problemi vincolati del tipo (4.2), ovvero

$$\begin{aligned} \min \quad & f(x) \\ & x \in S \end{aligned}$$

con le ipotesi illustrate in (4.1.1).

Le iterazioni del metodo del gradiente proiettato cercano di determinare

$$x^* = \arg \min_{x \in S} f(x)$$

e hanno la forma seguente

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k (\bar{\mathbf{x}}_k - \mathbf{x}_k) \quad (4.10)$$

dove

$$\bar{\mathbf{x}}_k = [\mathbf{x}_k - \mathbf{s}_k \nabla f(\mathbf{x}_k)]^+ \quad (4.11)$$

Il simbolo  $[\cdot]^+$  denota la proiezione (descritta nel seguito) sull'insieme  $S$ ,  $\alpha_k \in (0, 1]$  rappresenta il passo e  $\mathbf{s}_k$  è uno scalare positivo.

A partire dall'iterato  $\mathbf{x}_k$ , per calcolare il nuovo iterato  $\mathbf{x}_{k+1}$  si compie quindi prima di tutto un passo lungo la direzione dell'antigradiente  $(-\mathbf{s}_k \nabla f(\mathbf{x}_k))$  e si proietta poi il risultato di  $\mathbf{x}_k - \mathbf{s}_k \nabla f(\mathbf{x}_k)$  su  $S$ , ottenendo così il vettore ammissibile  $\bar{\mathbf{x}}_k$ . Infine, si compie un passo di lunghezza  $\alpha_k$  lungo la direzione ammissibile  $\bar{\mathbf{x}}_k - \mathbf{x}_k$ .

Osserviamo che  $x^* = [x^* - s \nabla f(x^*)]^+$ ,  $\forall s > 0$ , vale se e solo se  $x^*$  è un punto stazionario. Ossia, il metodo si arresterà se e solo se avrà incontrato un punto stazionario.

Infatti, per definizione un punto  $x^*$  è stazionario se

$$\nabla f(x^*)^T (x - x^*) \geq 0 \quad \forall x \in S$$

che è equivalente a

$$((x^* - s \nabla f(x^*)) - x^*)^T (x - x^*) \leq 0 \quad \forall x \in S, s > 0 \quad (4.12)$$

Inoltre, per le ipotesi fatte in (4.1.1) vale la proposizione seguente [15].

**Proposizione 4.2.1.**

Dato  $z \in \mathbb{R}^n$ , un vettore  $x^* \in S$  è uguale alla proiezione  $[z]^+$  se e solo se

$$(z - x^*)^T(x - x^*) \leq 0, \quad \forall x \in S \quad (4.13)$$

Ne viene quindi che  $x^* = [x^* - s\nabla f(x^*)]^+$ ,  $\forall s > 0$  se e solo se  $x^*$  è un punto stazionario.

Affinchè il metodo sia utilizzabile in senso pratico, è necessario che l'operazione di proiezione sia abbastanza semplice. Ad esempio, quando l'insieme di ammissibilità  $S$  può essere espresso nel modo seguente

$$S = \{x \mid \beta_i \leq x_i \leq \gamma_i, \quad i = 1, \dots, n\}$$

la  $i$ -esima componente della proiezione di un vettore  $x$  è data da

$$[x]_i^+ = \begin{cases} \beta_i & \text{se } x_i \leq \beta_i \\ \gamma_i & \text{se } x_i \geq \gamma_i \\ x_i & \text{altrimenti} \end{cases} \quad (4.14)$$

Esistono diverse tecniche per la ricerca del passo  $\alpha_k$  e dello scalare  $s_k$ .

Una di queste è la *Regola di Armijo lungo le direzioni ammissibili*.

Questa tecnica pone  $s_k := s$ ,  $\forall k = 0, 1, \dots$ , con  $s$  costante positiva, e sceglie  $\alpha_k$  attraverso la regola di Armijo sull'intervallo  $[0, 1]$ .

In particolare, fissati due scalari positivi  $\beta$  e  $\sigma$ , con  $\beta \in (0, 1)$  e  $\sigma \in (0, 1)$ , si definisce il passo  $k$ -esimo  $\alpha_k = \beta^{m_k}$  dove  $m_k$  è il primo intero non negativo  $m$  tale che

$$f(\mathbf{x}_k) - f(\mathbf{x}_k + \beta^m(\bar{\mathbf{x}}_k - \mathbf{x}_k)) \geq -\sigma\beta^m\nabla f(\mathbf{x}_k)^T(\bar{\mathbf{x}}_k - \mathbf{x}_k) \quad (4.15)$$

Per il metodo del gradiente proiettato con la ricerca del passo attraverso questa regola di Armijo vale il seguente risultato di convergenza.

**Proposizione 4.2.2.**

Sia  $\{\mathbf{x}_k\}$  una sequenza di iterati generata dal metodo del gradiente proiettato con  $\alpha_k$  scelto attraverso la regola di Armijo lungo le direzioni ammissibili. Allora, il punto limite di  $\{\mathbf{x}_k\}$  è un punto stazionario.

Il lettore interessato alla dimostrazione di questo risultato o ad una analisi più dettagliata della convergenza può fare riferimento a [15].

### 4.2.2 Metodo di Gauss-Newton

Il metodo di Gauss-Newton è uno dei metodi più noti per la risoluzione di un problema di minimi quadrati non vincolato.

La funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  da minimizzare su  $\mathbb{R}^n$  è quindi, come già detto in (4.9),

$$f(x) = \sum_{j=1}^m r_j^2(x) \quad (4.16)$$

dove ogni residuo  $r_j$  è una funzione  $r_j : \mathbb{R}^n \rightarrow \mathbb{R}$ .

In realtà, per comodità e semplificazione di calcoli si preferisce considerare la funzione

$$f(x) = \frac{1}{2} \sum_{j=1}^m r_j^2(x) \quad (4.17)$$

invece della precedente funzione (4.16), in quanto questo cambiamento non è influente al fine di determinare  $x^* = \arg \min_{x \in \mathbb{R}^n} f(x)$ .

Se indichiamo ora il *vettore residuo*  $r : \mathbb{R}^n \rightarrow \mathbb{R}^m$  come

$$r(x) = \begin{pmatrix} r_1(x) \\ r_2(x) \\ \vdots \\ r_m(x) \end{pmatrix} \quad (4.18)$$

possiamo scrivere

$$f(x) = \frac{1}{2} \|r(x)\|_2^2 \quad (4.19)$$

ed il problema dei minimi quadrati che vogliamo risolvere assume la forma

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|r(x)\|_2^2 \quad (4.20)$$

La matrice jacobiana  $J$  di  $r$  è una matrice  $(m \times n)$  definita come

$$J(x) = \begin{pmatrix} \nabla r_1(x)^T \\ \nabla r_2(x)^T \\ \vdots \\ \nabla r_m(x)^T \end{pmatrix} \quad (4.21)$$

ovvero

$$J_{ij}(x) = \frac{\partial r_i}{\partial x_j}, \quad \text{per } i = 1, \dots, m, \quad j = 1, \dots, n \quad (4.22)$$

Possiamo quindi scrivere

$$\nabla f(x) = \sum_{j=1}^m r_j(x) \nabla r_j(x) = J(x)^T r(x) \quad (4.23)$$

e

$$\begin{aligned} \nabla^2 f(x) &= \sum_{j=1}^m \nabla r_j(x) \nabla r_j(x)^T + \sum_{j=1}^m r_j(x) \nabla^2 r_j(x) = \\ &= J(x)^T J(x) + \sum_{j=1}^m r_j(x) \nabla^2 r_j(x) = \\ &= J(x)^T J(x) + Q(x) \end{aligned} \quad (4.24)$$

La matrice hessiana di  $f$ ,  $\nabla^2 f(x)$ , è quindi data dalla somma di due termini:  $J(x)^T J(x)$  e  $\sum_{j=1}^m r_j(x) \nabla^2 r_j(x)$ .

Il metodo di Gauss-Newton è un metodo iterativo che si basa sull'assunto che il primo di tali termini sia dominante rispetto all'altro e approssima quindi l'hessiana di  $f$  come  $\nabla^2 f(x) \approx J(x)^T J(x)$ .

Introducendo ora la seguente notazione

$$f_k = f(\mathbf{x}_k) \quad J_k = J(\mathbf{x}_k) \quad r_k = r(\mathbf{x}_k)$$

possiamo descrivere le iterazioni del metodo di Gauss-Newton come

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_k^{(gn)} \quad (4.25)$$

dove  $\mathbf{d}_k^{(gn)}$  è ottenuto come soluzione del sistema

$$J_k^T J_k \mathbf{d} = -J_k^T r_k \quad (4.26)$$

*Osservazione 1.* L'approssimazione  $\nabla^2 f_k \approx J_k^T J_k$  consente di evitare il calcolo di ogni matrice hessiana  $\nabla^2 r_j$ , per  $j = 1, \dots, m$ .

*Osservazione 2.* Se la matrice  $J_k$  ha rango pieno ed il gradiente  $\nabla f_k$  è diverso da zero, allora la direzione  $\mathbf{d}_k^{(gn)}$  è una direzione di discesa.

Infatti:

$$\begin{aligned} \mathbf{d}_k^{(gn)T} \nabla f_k &= \mathbf{d}_k^{(gn)T} J_k^T r_k = -\mathbf{d}_k^{(gn)T} J_k^T J_k \mathbf{d}_k^{(gn)} = \\ &= -\|J_k \mathbf{d}_k^{(gn)}\|_2^2 < 0 \end{aligned} \quad (4.27)$$

il che dimostra, per la proposizione (4.1.4), che nel caso considerato  $\mathbf{d}_k^{(gn)}$  è una direzione di discesa.

Se per qualche  $k$  la matrice  $J_k$  non ha rango pieno, la matrice dei coefficienti  $J_k^T J_k$  del sistema (4.26) è singolare.

Per ovviare a questo inconveniente il metodo di Gauss-Newton è spesso implementato nella forma modificata

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k (J_k^T J_k + D_k)^{-1} J_k^T r_k \quad (4.28)$$

dove  $\alpha_k$  è il passo scelto con un'opportuna ricerca unidimensionale e  $D_k$  rappresenta una matrice diagonale tale che la matrice  $J_k^T J_k + D_k$  sia definita positiva.

In particolare, nel caso in cui  $D_k$  sia un multiplo positivo della matrice identità il metodo è noto con il nome di **metodo di Levenberg-Marquardt**.

In questo caso l'iterazioni assume la forma

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k (J_k^T J_k + \mu_k I)^{-1} J_k^T r_k \quad (4.29)$$

dove lo scalare  $\mu_k$  è definito con la regola

$$\mu_k = \min\{\mu_1, \|J_k^T r_k\|_2\}$$

essendo  $\mu_1 > 0$ .

Il passo  $\alpha_k$  è determinato invece con il *metodo di Armijo per la ricerca in linea*. Questo metodo effettua successive riduzioni del passo  $\alpha$ , a partire

da un valore assegnato  $\bar{\alpha} > 0$ , fino a determinare un valore  $\alpha_k$  che soddisfi la seguente condizione

$$f(\mathbf{x}_k + \alpha \mathbf{p}_k) \leq f(\mathbf{x}_k) + \gamma \alpha \nabla f(\mathbf{x}_k)^T \mathbf{p}_k \quad (4.30)$$

dove  $\gamma \in (0, 1)$  e  $\mathbf{p}_k$  in questo caso vale  $-(J_k^T J_k + \mu_k I)^{-1} J_k^T r_k$ .

In generale si parte da un passo  $\bar{\alpha}=1$  che viene ridotto di volta in volta moltiplicando per un fattore  $\rho = \frac{1}{2}$ .

Il metodo di Levenberg-Marquardt descritto attraverso le iterazioni in (4.29) è un metodo convergente.

Il lettore che fosse interessato ad un'analisi più dettagliata può fare riferimento a [15].

### 4.3 Problema lineare dei minimi quadrati

Nel caso particolare in cui i residui  $r_j$ , per  $j = 1, \dots, m$ , siano funzioni del tipo  $r_j(x) = a_j^T x - b_j$ , con  $a_j \in \mathbb{R}^n$  e  $b_j \in \mathbb{R}$ , la funzione da minimizzare in (4.9) assume la forma

$$f(x) = \sum_{j=1}^m (a_j^T x - b_j)^2 \quad (4.31)$$

Assumendo poi che  $m > n$  e definendo una matrice  $A$  ( $m \times n$ ) con righe  $a_j^T \in \mathbb{R}^n$  e un vettore  $b \in \mathbb{R}^m$  con componenti  $b_j$ , si può riscrivere (4.31) come

$$f(x) = \|Ax - b\|_2^2 \quad (4.32)$$

Il *problema lineare dei minimi quadrati* è quindi un problema in cui la funzione da minimizzare è del tipo  $\|Ax - b\|_2^2$ .

Nel caso non vincolato, il problema viene formulato come

$$\begin{aligned} \min \quad & \|Ax - b\|_2^2 \\ & x \in \mathbb{R}^n \end{aligned} \quad (4.33)$$

e la sua soluzione  $x^* = \arg \min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2$  è detta soluzione ai minimi quadrati del sistema sovradimensionato  $Ax = b$ .

A partire da (4.32) si verifica immediatamente che

$$\nabla f(x) = 2A^T(Ax - b), \quad \nabla^2 f(x) = 2A^T A \quad (4.34)$$

e quindi la matrice hessiana è semidefinita positiva, in quanto

$$2x^T A^T A x = 2\|Ax\|_2^2 \geq 0 \quad (4.35)$$

La funzione da minimizzare è quindi una funzione convessa e l'annullamento del gradiente si esprime attraverso il cosiddetto *sistema delle equazioni normali*

$$A^T A x = A^T b \quad (4.36)$$

Se  $A$  ha rango massimo uguale ad  $n$  la matrice  $A^T A$  è simmetrica e definita positiva, per cui il sistema (4.36) ammette un'unica soluzione

$$x^* = (A^T A)^{-1} A^T b \quad (4.37)$$

È importante osservare che il sistema delle equazioni normali è sempre più *mal condizionato* rispetto al sistema di partenza sovradeterminato  $Ax = b$ , in quanto vale  $K_2(A^T A) = K_2(A)^2$ , dove  $K_2(\cdot)$  rappresenta il numero di condizionamento. Ricordiamo che “sistema mal condizionato” significa che a piccole perturbazioni sui dati corrispondono grandi perturbazioni nei risultati.

Inoltre, se la matrice  $A$  è già mal condizionata, il calcolo di  $A^T A$  può portare a grossi inconvenienti.

Da un punto di vista numerico è quindi preferibile utilizzare un metodo più stabile per la risoluzione del problema basato sulla fattorizzazione  $QR$  della matrice  $A$ .

**Proposizione 4.3.1.**

*Sia  $A$  una matrice  $(m \times n)$ . Allora esiste una matrice  $Q$  ortogonale di ordine  $m$  tale che  $A=QR$  dove  $R$  è una matrice trapezoidale superiore  $(m \times n)$ . Inoltre  $\text{rango}(A)=\text{rango}(R)$ .*

La dimostrazione dell'esistenza di  $Q$  è costruttiva e si possono utilizzare le riflessioni elementari di Householder o le rotazioni elementari di Givens.

Ora, utilizzando la fattorizzazione  $QR$  della matrice  $A$  si ottiene

$$\|Ax - b\|_2^2 = \|QRx - QQ^Tb\|_2^2 = \|Q(Rx - Q^Tb)\|_2^2 = \|Rx - Q^Tb\|_2^2 \quad (4.38)$$

Introduciamo la seguente notazione

$$R = \begin{pmatrix} R_1 \\ 0 \end{pmatrix} \quad Q^Tb = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \quad (4.39)$$

dove  $R_1$  è una matrice triangolare superiore ( $n \times n$ ) e  $y_1 \in \mathbb{R}^n$ ,  $y_2 \in \mathbb{R}^{m-n}$ .

Ne viene che

$$\|Ax - b\|_2^2 = \|Rx - Q^Tb\|_2^2 = \|R_1x - y_1\|_2^2 + \|y_2\|^2 \quad (4.40)$$

Il problema (4.33) diventa quindi

$$\min_{x \in \mathbb{R}^n} \|R_1x - y_1\|_2^2 + \|y_2\|^2 \quad (4.41)$$

In questo modo, il sistema che si deve cercare di risolvere è diventato il seguente sistema triangolare

$$R_1x = y_1 \quad (4.42)$$

la cui soluzione nel caso in cui  $\text{rango}(A) = n$  è unica ed è data da

$$x^* = R_1^{-1}y_1 \quad (4.43)$$

Osserviamo inoltre che in questo caso  $K_2(R_1) = K_2(A)$ .

Per questo motivo è sempre preferibile passare attraverso la fattorizzazione  $QR$  e risolvere il sistema triangolare (4.42) piuttosto che risolvere direttamente il sistema delle equazioni normali.

# Capitolo 5

## Stima dei parametri

### 5.1 Obiettivo di questo lavoro

Il lavoro svolto all'interno di questa tesi si inserisce, come detto alla fine del capitolo 3, all'interno del momento *off-line* del retrieval presente nel progetto a cura del Dott. Ivan Tomba e dell'Ing. Andrea Visotti.

Per poter spiegare in modo esauriente l'obiettivo del nostro lavoro, dobbiamo prima di tutto descrivere in termini matematici la situazione su cui lavoreremo.

#### Situazione

Fissato un database di  $N$  immagini  $I_1, I_2, \dots, I_N$ , con  $N \in \mathbb{N}$ ,  $N \gg 1$ , sono date  $K$  misurazioni del database ( $K \in \mathbb{N}$ ) che permettono di associare ad ogni immagine  $I_i$   $K$  oggetti (o descrittori)  $O_i^{(k)}$ , con  $k = 1, \dots, K$ .

Ad esempio, le misurazioni e gli oggetti associati ad un'immagine possono essere rispettivamente le funzioni misuranti e le corrispondenti funzioni di taglia. Tuttavia si possono prendere in considerazione anche altri descrittori, diversi dalle funzioni di taglia, come vedremo più avanti nel sesto capitolo.

Sullo spazio degli oggetti generati dalla misurazione  $k$ -esima è definita una distanza  $d_k$ ,  $k = 1, \dots, K$ , alla quale è associata la matrice  $D^{(k)}$  delle

distanze reciproche fra gli oggetti, definita da

$$D_{ij}^{(k)} := d_k(O_i^{(k)}, O_j^{(k)}) \quad \text{per } i, j = 1, \dots, N. \quad (5.1)$$

Ad esempio, data la  $k$ -esima funzione misurante, la distanza  $d_k$  è la distanza bottleneck e la matrice  $D^{(k)}$  è ottenuta a partire da una matrice distanza come (3.1) sopprimendo la prima e la seconda riga e colonna, ovvero la parte della matrice distanza relativa alla diagnosi.

Lavoreremo sempre con matrici  $D^{(k)}$ ,  $k = 1, \dots, K$ , normalizzate tra 0 e 1, ovvero ottenute dividendo ogni termine della matrice per il termine di valore massimo della matrice stessa.

### Obiettivo

Ci prefiggiamo di trovare  $K$  parametri  $\alpha_1, \dots, \alpha_K$ ,  $\alpha_k \geq 0 \forall k = 1, \dots, K$ , raccolti in un vettore colonna  $\alpha \in \mathbb{R}^K$ , in modo tale che definite

$$d = d(\alpha) := \sum_{k=1}^K \alpha_k d_k, \quad D = D(\alpha) := \sum_{k=1}^K \alpha_k D^{(k)} \quad (5.2)$$

la distanza globale e la corrispondente matrice globale, il *retrieval rispetto a d* sia ottimizzato.

Per *retrieval rispetto a d* ( $d(\alpha)$ ) si intende, presa un'immagine come query, il recupero effettuato recuperando le immagini “più vicine” rispetto a  $d$ , ovvero le immagini che si trovano alle distanze globali più basse dalla query. Osserviamo che riordinando le colonne della matrice  $D$  in ordine crescente, si effettua il *retrieval rispetto a d* per tutte le immagini del database, considerate una ad una come query.

Per poter essere in grado nel seguito di stabilire cosa significhi *retrieval ottimizzato*, dobbiamo spiegare come viene definita la rilevanza nel nostro sistema di recupero.

Purtroppo, per le immagini del database da noi utilizzato, non abbiamo a disposizione un giudizio di rilevanza o non rilevanza, rispetto ad un'immagine del database stesso scelta come query, proveniente da esperti del settore (ovvero dermatologi) e siamo quindi costretti ad utilizzare la diagnosi istologica

dicotomica *nevo-melanoma*. Ossia, supporremo che nel database vi siano  $n$  nevi e  $m := N - n$  melanomi, con  $n \geq m$ .

Vogliamo ora scrivere un modello semplice per la stima dei parametri basato sulla risoluzione di un problema dei minimi quadrati, in cui la funzione da minimizzare sia del tipo

$$g(\alpha) := \|f(\alpha) - \mu\|_2^2 \quad (5.3)$$

## 5.2 Modello vettoriale

Per poter definire un modello che ci permetta di stimare i parametri  $\alpha_k$  attraverso la risoluzione del problema dei minimi quadrati in (5.3), dobbiamo definire chi siano  $f(\alpha)$  e  $\mu$ .

**Definizione 5.1.** Sia  $T \in \mathbb{R}^N$  il vettore colonna le cui componenti  $T(i)$ , per  $i = 1, \dots, N$ , sono definite come segue:

$$T(i) = \begin{cases} \frac{1}{n} & \text{se l'immagine } I_i \text{ è nevo} \\ -\frac{1}{m} & \text{se l'immagine } I_i \text{ è melanoma} \end{cases} \quad (5.4)$$

ricordando che  $N$  è il numero di immagini presenti nel database,  $n$  il numero di nevi ed  $m := N - n$  il numero di melanomi.

Data una generica matrice di distanze reciproche  $\mathcal{D}$ , denotiamo con  $\mathcal{D}_i$ ,  $i = 1, \dots, N$ , la  $i$ -esima colonna di  $\mathcal{D}$  e con  $\tilde{\mathcal{D}}_i$  il riordinamento, in senso crescente, di  $\mathcal{D}_i$ . Osserviamo che il prodotto scalare tra  $\mathcal{D}_i$  e  $T$  fornisce, in un certo senso, una possibile “diagnosi” della  $i$ -esima lesione  $I_i$ . Infatti, quanto più le immagini più vicine a  $I_i$  saranno nevi, tanto più il risultato di questo prodotto si avvicinerà a -1; viceversa, quanto più le immagini più vicine ad  $I_i$  saranno melanomi, tanto più il risultato si avvicinerà a 1.

A partire da questa considerazione, abbiamo deciso di definire  $f(\alpha)$  come

$$f(\alpha) := D(\alpha)T = \sum_{k=1}^K \alpha_k D^{(k)}T \quad (5.5)$$

Per definire il vettore  $\mu$  procediamo come segue.

Data come query l'immagine  $I_i$ , assumiamo come *retrieval* più auspicabile per la query la seguente situazione:

- se la query  $I_i$  è un nevo:
  1. le  $n$  immagini più vicine sono nevi (la query stessa è compresa e sarà chiaramente la prima immagine recuperata, in quanto la sua distanza da se stessa è uguale a 0);
  2. il vettore  $U$  delle distanze dalla query è distribuito uniformemente sull'intervallo  $[0, 1]$ ;
- se la query  $I_i$  è un melanoma:
  1. le  $m$  immagini più vicine sono melanomi (la query stessa è compresa);
  2. il vettore  $U$  delle distanze dalla query è distribuito uniformemente sull'intervallo  $[0, 1]$ .

*Osservazione 3.* Ciò che cambia nei due casi, in sostanza, è solo l'ipotesi 1 a seconda che la query  $I_i$  sia un nevo o un melanoma.

Calcolando ora la diagnosi come prima, ovvero calcolando il prodotto scalare tra il vettore  $U$  ed il vettore  $T$ , si ottengono i seguenti valori:

- se  $I_i$  è nevo:

$$\begin{aligned}
 \mu^- &= \sum_{i=1}^{n-1} \frac{i}{n(N-1)} - \sum_{i=n}^{N-1} \frac{i}{m(N-1)} = \\
 &= \frac{n-1}{2(N-1)} - \left( \frac{N}{2m} - \frac{n(n-1)}{2m(N-1)} \right) = \quad (5.6) \\
 &= -\frac{m(n+m)}{2m(N-1)} = \\
 &= -\frac{N}{2(N-1)}
 \end{aligned}$$

- se  $I_i$  è melanoma:

$$\begin{aligned}
\mu^+ &= -\sum_{i=1}^{m-1} \frac{i}{m(N-1)} + \sum_{i=m}^{N-1} \frac{i}{n(N-1)} = \\
&= -\frac{m-1}{2(N-1)} + \left( \frac{N}{2n} - \frac{m(m-1)}{2n(N-1)} \right) = \\
&= \frac{n(n+m)}{2n(N-1)} = \\
&= \frac{N}{2(N-1)}
\end{aligned} \tag{5.7}$$

da cui si ottiene la definizione del vettore  $\mu$ :

$$\mu(i) = \begin{cases} \mu^- & \text{se l'immagine } I_i \text{ è nevo} \\ \mu^+ & \text{se l'immagine } I_i \text{ è melanoma} \end{cases} \tag{5.8}$$

per  $i = 1, \dots, N$ .

*Osservazione 4.* I valori ottenuti per  $\mu^+$  e  $\mu^-$  non dipendono dal numero di nevi ( $n$ ) e melanomi ( $m$ ) presenti nel database, ma dipendono soltanto dalla dimensione del database stesso ( $N$ ).

*Osservazione 5.* Le componenti  $f_i$  della funzione  $f : \mathbb{R}^K \rightarrow \mathbb{R}^N$  si possono interpretare come la “diagnosi” basata sul *retrieval di  $I_i$  rispetto a  $D$* . Infatti, quanto più i melanomi saranno vicini alla query rispetto ai nevi, tanto più questo valore tenderà ad avvicinarsi  $\mu^+$ ; quanto più i nevi saranno vicini alla query tanto più tenderà ad avvicinarsi a  $\mu^-$ .

*Osservazione 6.* La matrice jacobiana della  $f$  è data da

$$J_{ik}(\alpha) = \frac{\partial f_i}{\partial \alpha_k}(\alpha) = \langle D_i^{(k)}, T \rangle \quad \text{per } i = 1, \dots, N, \quad k = 1, \dots, K \tag{5.9}$$

*Osservazione 7.* Le componenti del gradiente di  $g(\alpha) = \|f(\alpha) - \mu\|_2^2$  sono date da

$$\frac{\partial g}{\partial \alpha_k}(\alpha) = 2 \langle (f(\alpha) - \mu), D^{(k)} T \rangle \quad \forall k = 1, \dots, K \tag{5.10}$$

Infatti, poiché

$$\begin{aligned} g(\alpha) &= \|f(\alpha) - \mu\|_2^2 \\ &= \sum_{j=1}^N (f_j(\alpha) - \mu_j)^2 \end{aligned}$$

si ottiene:

$$\begin{aligned} \frac{\partial g}{\partial \alpha_k}(\alpha) &= \frac{\partial}{\partial \alpha_k} \sum_{j=1}^N (f_j(\alpha) - \mu_j)^2 = \\ &= \sum_{j=1}^N 2(f_j(\alpha) - \mu_j) ((D_j^{(k)})^T T) = \\ &= 2 \langle (f(\alpha) - \mu), D^{(k)} T \rangle \end{aligned}$$

*Osservazione 8.* Le ipotesi assunte per definire il vettore  $\mu$  possono essere sostituite parzialmente o interamente. In particolare, se in futuro si avrà a disposizione il giudizio di rilevanza del dermatologo, si potrà valutare di costruire il vettore  $\mu$  sulla base di quanto da lui indicato.

In questo modello il vettore  $\mu$  è stato costruito in base ad un *retrieval* ideale per ogni singola immagine  $I_i$ , ma globalmente non è detto che esista una matrice di distanze reciproche  $\mathcal{D}$  (ovvero simmetrica con diagonale nulla) di dimensione  $N$ , con  $N \geq 3$ , tale che  $\tilde{\mathcal{D}}_i$  (ovvero il riordinamento in ordine crescente della  $i$ -esima colonna) soddisfi le ipotesi 1 e 2 per ogni  $i = 1, \dots, N$ . Infatti, per  $N$  dispari non è possibile costruire una tale matrice, come possibile dedurre da un risultato noto di teoria dei grafi (colorazione dei lati di grafi completi).

Vale però il seguente risultato.

**Proposizione 5.2.1.**

Sia  $\bar{\mathcal{D}}$  una matrice di distanze reciproche costante a blocchi del tipo:

$$\bar{\mathcal{D}}_{ij} = \begin{cases} 0 & \text{per } i = j \\ s & \text{per } i, j \leq n \text{ e } i \neq j \\ r & \text{per } i \leq n, j > n \text{ o } i > n, j \leq n \\ t & \text{per } i, j > n \text{ e } i \neq j \end{cases} \quad (5.11)$$

con  $0 < s < 1$ ,  $0 < r < 1$  e  $0 < t < 1$ .

Siano  $2 \leq m \leq n$ . Allora, per una tale matrice a blocchi  $\overline{\mathcal{D}}$  vale che

$$\overline{\mathcal{D}}T = \mu \iff r > \max\{s, t\} \quad (5.12)$$

dove  $r$  rappresenta la distanza reciproca nevo-melanoma,  $s$  la distanza nevo-nevo e  $t$  la distanza melanoma-melanoma.

Inoltre si possono scegliere  $r$ ,  $s$  e  $t$  in modo tale che  $\overline{\mathcal{D}}T = \mu$ .

*Dimostrazione.*

$$\begin{aligned} \overline{\mathcal{D}}T = \mu &\iff \begin{cases} \left(\frac{n-1}{n}\right)s - r = \mu^- \\ r - \left(\frac{m-1}{m}\right)t = \mu^+ \end{cases} \\ &\iff \begin{cases} s = (r + \mu^-)\frac{n}{n-1} \\ t = (r - \mu^+)\frac{m}{m-1} \end{cases} \end{aligned} \quad (5.13)$$

Le condizioni di positività su  $r$ ,  $s$  e  $t$  impongono che

$$r > \max\{-\mu^-, \mu^+\} \quad (5.14)$$

mentre da (5.13) segue immediatamente che

$$\begin{cases} s > r \iff r > -n\mu^- \\ t > r \iff r > m\mu^+ \end{cases} \quad (5.15)$$

Dato che abbiamo assunto distanze comprese tra 0 e 1, le due condizioni in (5.15) non possono mai verificarsi essendo  $2 \leq m \leq n$ .

Quindi questo dimostra che  $\overline{\mathcal{D}}T = \mu \iff r > \max\{s, t\}$ .

Inoltre, poiché  $\mu^+ < 1$  una qualunque scelta di  $r$  nell'intervallo  $]\mu^+, 1[$  porta, attraverso la formula (5.13), ad una scelta della matrice  $\overline{\mathcal{D}}$  che verifichi  $\overline{\mathcal{D}}T = \mu$ .

□

Questo risultato è per noi particolarmente importante in quanto, se fosse stato possibile ottenere il vettore  $\mu$  a partire da una matrice di distanze reciproche costante a blocchi con  $r < s$  o  $r < t$ , la soluzione del problema dei

minimi quadrati da noi descritto sarebbe potuta essere relativa ad una configurazione matriciale molto diversa o addirittura opposta rispetto a quella da noi auspicata.

Con le scelte di  $f(\alpha)$  e  $\mu$  illustrate in questo paragrafo, un *retrieval ottimizzato* sarà dunque ottenuto mediante una scelta dei parametri  $\alpha_1, \dots, \alpha_K$  ( $\alpha_k \geq 0 \forall k = 1, \dots, K$ ) per cui la funzione dei minimi quadrati

$$g(\alpha) = \left\| \sum_{k=1}^K \alpha_k D^{(k)} T - \mu \right\|_2^2 \quad (5.16)$$

sia minima.

Osserviamo infine che possiamo riscrivere il problema che vogliamo risolvere come un problema lineare dei minimi quadrati. Vale infatti

$$g(\alpha) = \|A\alpha - b\|_2^2 \quad (5.17)$$

dove  $b := \mu \in \mathbb{R}^N$  e la matrice  $A$  è la seguente matrice ( $N \times K$ )

$$A_{ik} = \langle D_i^{(k)}, T \rangle \quad \text{per } i = 1, \dots, N, \quad k = 1, \dots, K \quad (5.18)$$

### 5.3 Modello matriciale

La proposta di un secondo modello per la risoluzione del nostro problema dei minimi quadrati nasce a partire dalla proposizione (5.2.1) che, come descritto precedentemente, dimostra la possibilità di trovare una matrice  $\bar{D}$  di distanze reciproche  $r, s, t$  costante a blocchi, con  $r > \max\{s, t\}$ , tale che  $\bar{D}T = \mu$ . Inoltre è importante sottolineare che una tale matrice di distanze reciproche  $r, s, t$  è realizzabile in uno spazio euclideo di dimensione sufficiente.

Questi risultati ci hanno spinto a costruire un modello in cui  $\mu$  fosse una matrice e non un vettore (da qui il nome modello matriciale), ed in particolare una matrice ( $N \times N$ ) definita come segue:

$$\mu_{ij} = \begin{cases} 0 & \text{se } i = j \\ r & \text{se } I_i \text{ nevo e } I_j \text{ melanoma o viceversa} \\ s & \text{se } I_i \text{ nevo e } I_j \text{ nevo} \\ t & \text{se } I_i \text{ melanoma e } I_j \text{ melanoma} \end{cases} \quad (5.19)$$

con  $r > \max\{s, t\}$ ,  $0 < s < 1$ ,  $0 < t < 1$ ,  $0 < r < 1$ .

Come sequenza di questa definizione di  $\mu$  abbiamo definito in modo molto naturale  $f(\alpha)$  come

$$f(\alpha) = \sum_{k=1}^K \alpha_k D^{(k)} \quad (5.20)$$

dove  $D^{(k)}$  rappresenta come sempre la  $k$ -esima matrice delle distanze reciproche.

*Osservazione 9.* Il modello proposto presenta una chiara dipendenza dai valori  $r$ ,  $s$  e  $t$  che determinano la matrice  $\mu$ .

*Osservazione 10.* Le componenti del gradiente di  $g(\alpha) = \|f(\alpha) - \mu\|_2^2$  sono date da

$$\frac{\partial g}{\partial \alpha_s}(\alpha) = 2 \sum_{i,j=1}^N \left( \sum_{k=1}^K \alpha_k D_{ij}^{(k)} - \mu_{ij} \right) D_{ij}^{(s)} \quad \forall s = 1, \dots, K \quad (5.21)$$

in quanto:

$$\begin{aligned} g(\alpha) &= \|f(\alpha) - \mu\|_2^2 \\ &= \sum_{i,j=1}^N \left( \sum_{k=1}^K \alpha_k D_{ij}^{(k)} - \mu_{ij} \right)^2 \end{aligned}$$

*Osservazione 11.* Se consideriamo le matrici  $D^{(k)}$  di dimensione  $(N \times N)$  come vettori in  $\mathbb{R}^{N^2}$ , che indichiamo con  $\bar{d}^{(k)}$ , attraverso una concatenazione verticale delle colonne, possiamo rappresentare le componenti  $f_i$  della funzione  $f : \mathbb{R}^K \rightarrow \mathbb{R}^{N^2}$  come

$$f_i(\alpha) = \sum_{k=1}^K \alpha_k \bar{d}_i^{(k)} \quad \forall i = 1, \dots, N^2 \quad (5.22)$$

La matrice jacobiana della  $f$  sarà quindi data da

$$J_{ik} = \frac{\partial f_i}{\partial \alpha_k}(\alpha) = \bar{d}_i^{(k)} \quad \text{per } i = 1, \dots, N^2, k = 1, \dots, K \quad (5.23)$$

Con le scelte di  $f(\alpha)$  e  $\mu$  illustrate in questo modello, un *retrieval ottimizzato* sarà dunque ottenuto mediante una scelta dei parametri  $\alpha_1, \dots, \alpha_K$  ( $\alpha_k \geq 0 \forall k = 1, \dots, K$ ) per cui la funzione dei minimi quadrati

$$g(\alpha) = \left\| \sum_{k=1}^K \alpha_k D^{(k)} - \mu \right\|_2^2 \quad (5.24)$$

sia minima.

Osserviamo infine che, come nel caso precedente, possiamo riscrivere il problema che vogliamo risolvere come un problema lineare dei minimi quadrati. Vale infatti

$$g(\alpha) = \|A\alpha - b\|_2^2 \quad (5.25)$$

dove  $b \in \mathbb{R}^{N^2}$  è il vettore ottenuto concatenando verticalmente le colonne della matrice  $\mu$  e la matrice  $A$  è la seguente matrice ( $N^2 \times K$ )

$$A_{ik} = \bar{d}_i^{(k)} \quad \text{per } i = 1, \dots, N^2, \quad k = 1, \dots, K \quad (5.26)$$

# Capitolo 6

## Sperimentazione e valutazione risultati

### 6.1 Metodi di risoluzione

Per stimare i parametri da utilizzare per costruire la matrice globale delle distanze reciproche  $D$ , abbiamo implementato alcuni metodi in MATLAB volti alla risoluzione dei problemi dei minimi quadrati (5.16) e (5.24) relativi ai due modelli proposti.

Dal momento che abbiamo richiesto che i parametri  $\alpha_k$ , raccolti nel vettore colonna  $\alpha \in \mathbb{R}^K$ , siano tutti non negativi, ovvero  $\alpha_k \geq 0, \forall k = 1, \dots, K$ , i problemi che vogliamo risolvere diventano problemi vincolati, in cui il vincolo si può esprimere come  $\alpha \in [0, +\infty[^K$ .

Può però risultare interessante valutarne la risoluzione anche dal punto di vista non vincolato, ammettendo cioè che  $\alpha \in \mathbb{R}^K$ , per osservare “quanto negative” siano le componenti della soluzione calcolata in questo modo.

Lo scopo è quello di poter poi analizzare “quanto cambierebbe” la valutazione del *retrieval rispetto a*  $d(\alpha)$  con  $\alpha$  vettore calcolato nel caso vincolato, rispetto alla valutazione del *retrieval rispetto a*  $d(\alpha)$  con  $\alpha$  vettore calcolato nel caso non vincolato, dove vengono poste uguali a zero le componenti negative.

Abbiamo deciso di implementare sia metodi di risoluzione propri di un problema lineare dei minimi quadrati, sia metodi più generali che possano essere utilizzati per la ricerca del minimo qualunque sia la funzione  $g$  dei minimi quadrati. Questa scelta deriva principalmente dal fatto che abbiamo la necessità di avere pronti algoritmi in grado di risolvere il nostro problema di stima dei parametri, anche nel caso in cui in futuro si debba cambiare la funzione  $g$  dei minimi quadrati da minimizzare. Infatti, appena si avrà a disposizione il giudizio di rilevanza da parte dei dermatologi, le definizioni di  $\mu$  e  $f$  potrebbero cambiare sensibilmente e potrebbe anche succedere che il problema non sia più lineare.

Inoltre, i metodi “generali” che abbiamo scelto sono metodi iterativi che, per loro natura, permettono anche di analizzare la successione degli iterati calcolati.

In particolare, abbiamo implementato il metodo del gradiente proiettato (4.2.1) per la risoluzione nel caso vincolato e il metodo di Levenberg-Marquardt (4.2.2) per la risoluzione nel caso non vincolato. In entrambi i metodi abbiamo utilizzato un criterio d’arresto sulla norma della differenza tra due iterati successivi; in più, nel metodo di Levenberg-Marquardt abbiamo utilizzato anche un criterio d’arresto relativo sulla norma del gradiente della funzione obiettivo. I file contenenti le funzioni relative a questi due metodi, da noi scritte in MATLAB, sono riportate nell’Appendice C : `gradiente_proiettato_01.m` e `levenberg_marquardt.m`.

Dopo alcuni test preliminari abbiamo deciso di considerare come insieme di ammissibilità per il metodo del gradiente proiettato l’insieme  $S = \{\alpha \mid 0 \leq \alpha_k \leq 1, k = 1, \dots, K\}$  e di fissare la tolleranza per il criterio d’arresto `tol1=1.e-6`. Per il metodo di Levenberg-Marquardt abbiamo invece fissato `tol1=1.e-3` e `tol2=1.e-6`.

Per quanto riguarda invece i metodi diretti propri di un problema lineare dei minimi quadrati, abbiamo deciso di risolvere il caso non vincolato utilizzando l’operatore (`\`) di MATLAB, che chiameremo da qui in avanti funzione `backslash`. Questa funzione utilizza al suo interno la fattorizzazione  $QR$  e,

per quanto detto nel paragrafo (4.3), fornisce quindi un metodo più stabile rispetto alla risoluzione diretta del sistema delle equazioni normali. Riportiamo, sempre nell'Appendice C, un esempio di `script` che utilizza questa funzione.

Per il problema lineare nel caso vincolato abbiamo invece scritto una funzione `TrovaAlfa` (Appendice C: file `TrovaAlfa.m`) che, a partire dal sistema che si ottiene imponendo uguali a zero le componenti del gradiente della funzione  $g$ , calcola la soluzione  $\alpha$  corrispondente e nel caso in cui ci siano componenti  $\alpha_k$  negative ricalcola la soluzione, imponendo via via che queste componenti siano uguali a zero, fino ad ottenere la soluzione minima che verifica la condizione  $\alpha_k \geq 0, \forall k = 1, \dots, K$ .

Prima di proseguire riportiamo i sistemi da dare in input alla funzione `TrovaAlfa` relativi ai nostri due modelli.

Modello vettoriale:

per semplificare la notazione indichiamo con  $D_i^{(k)}$  la  $i$ -esima riga della matrice  $D^{(k)}$ . Si ottiene:

$$\frac{\partial g}{\partial \alpha_s} = 2 \sum_{i=1}^N \left( \sum_{k=1}^K \alpha_k D_i^{(k)} T - \mu_i \right) D_i^{(s)} T = 0 \quad \forall s = 1, \dots, K \quad (6.1)$$

$$\tilde{A}\alpha = \tilde{b}, \quad \text{con } \tilde{A} \text{ matrice } (K \times K), \quad \tilde{b} \in \mathbb{R}^K$$

$$\tilde{A}_{sk} = \sum_{i=1}^N (D_i^{(k)} T)(D_i^{(s)} T) \quad \text{per } s, k = 1, \dots, K \quad (6.2)$$

$$\tilde{b}_s = \sum_{i=1}^N \mu_i (D_i^{(s)} T) \quad \text{per } s = 1, \dots, K$$

Modello matriciale:

$$\frac{\partial g}{\partial \alpha_s} = 2 \sum_{i,j=1}^N \left( \sum_{k=1}^K \alpha_k D_{ij}^{(k)} - \mu_{ij} \right) D_{ij}^{(s)} = 0 \quad \forall s = 1, \dots, K \quad (6.3)$$

$$\begin{aligned}
\tilde{A}\alpha &= \tilde{b}, \quad \text{con } \tilde{A} \text{ matrice } (K \times K), \quad \tilde{b} \in \mathbb{R}^K \\
\tilde{A}_{sk} &= \sum_{i,j=1}^N D_{ij}^{(k)} D_{ij}^{(s)} \quad \text{per } s, k = 1, \dots, K \\
\tilde{b}_s &= \sum_{i,j=1}^N \mu_{ij} D_{ij}^{(s)} \quad \text{per } s = 1, \dots, K
\end{aligned} \tag{6.4}$$

## 6.2 Metodi di valutazione

Per valutare il sistema di recupero in esame sulla base del *retrieval rispetto a*  $d(\alpha)$ , con  $\alpha$  vettore contenente i parametri ottimali stimati attraverso i metodi precedenti, utilizzeremo *precision* e *recall* illustrati nel paragrafo (2.1) e altri indici propri delle classificazioni.

Infatti, come già sottolineato più volte, non avendo a disposizione un giudizio di rilevanza proveniente da dermatologi, siamo obbligati ad utilizzare la diagnosi istologica, ovvero la classificazione nevo-melanoma.

In particolare, utilizzeremo gli indici percentuali di *specificità*, *sensibilità* e *accuratezza diagnostica* che nel nostro caso rappresentano:

- *specificità* : percentuale di nevi riconosciuti dal sistema;
- *sensibilità* : percentuale di melanomi riconosciuti dal sistema;
- *accuratezza diagnostica* : percentuale di diagnosi riconosciute dal sistema.

Da un punto di vista implementativo, gli indici appena descritti vengono calcolati come segue.

Indichiamo con  $\bar{T}$  il vettore le cui componenti sono così definite

$$\bar{T}_i = \bar{T}(i) = \begin{cases} 0 & \text{se } I_i \text{ nevo} \\ 1 & \text{se } I_i \text{ melanoma} \end{cases}$$

per  $i = 1, \dots, N$ .

Indichiamo come al solito con  $D = D(\alpha)$  la matrice globale, con  $N$  il numero di elementi del database, con  $n$  il numero di nevi presenti nel database e con  $m := N - n$  il numero di melanomi.

Possimo allora schematizzare il calcolo degli indici di specificità, sensibilità e accuratezza diagnostica attraverso alcuni passi:

1. fissare un valore  $s$  di *vicini* da considerare per valutare il *retrieval* ed inizializzare le variabili **spec**=0, **sens**=0 e **acc**=0.
2. per ogni  $i = 1, \dots, N$  ripetere:

(a) determinare il riordinamento  $\tilde{D}_i$ , in senso crescente, della  $i$ -esima colonna di  $D$ ;

(b) determinare il corrispondente riordinamento  $\tilde{T}^{(i)}$  del vettore  $\bar{T}$ ;

(c) calcolare:

$$v_i := \sum_{j=2}^{s+1} \sum_{h=2}^j \tilde{T}_h^{(i)} - \frac{(m - \bar{T}_i)s(s+1)}{2(N-1)} \quad (6.5)$$

(d) se  $v_i < 0$  e  $\bar{T}_i = 0$  incrementare **spec** e **acc**;

se  $v_i > 0$  e  $\bar{T}_i = 1$  incrementare **sens** e **acc**.

3. calcolare gli indici percentuali:

**spec** \*  $\frac{100}{n}$   $\longrightarrow$  specificità

**sens** \*  $\frac{100}{m}$   $\longrightarrow$  sensibilità

**acc** \*  $\frac{100}{N}$   $\longrightarrow$  accuratezza diagnostica

*Osservazione 12.* La somma in (6.5) viene fatta a partire dalla posizione  $j = 2$  in quanto nella prima posizione del *retrieval* figurerà sempre l'immagine stessa a distanza 0.

*Osservazione 13.* Il valore  $v_i$  è una sorta di diagnosi della lesione. Equivale alla somma per  $j = 2, \dots, s+1$  del numero dei melanomi trovati dal *retrieval* nelle posizioni dalla 2 alla  $j$  meno il valore atteso (in relazione al loro numero totale nel database) del numero dei melanomi nelle posizioni dalla 2 alla  $j$ .

Infatti, detto  $X(j)$  il numero di melanomi trovati da un generico *retrieval* nelle posizioni dalla 2 alla  $j$  e detto  $X(j)^{(i)}$  il numero di melanomi trovati dal *retrieval* di  $I_i$  nelle posizioni dalla 2 alla  $j$ , si ha:

$$\begin{aligned}
\sum_{j=2}^{s+1} (X(j)^{(i)} - \mathbb{E}(X(j))) &= \sum_{j=2}^{s+1} \left( \sum_{h=2}^j \tilde{T}_h^{(i)} - \frac{(j-1)(m - \bar{T}_i)}{N-1} \right) = \\
&= \sum_{j=2}^{s+1} \sum_{h=2}^j \tilde{T}_h^{(i)} - \sum_{l=1}^s \frac{l(m - \bar{T}_i)}{N-1} = \\
&= \sum_{j=2}^{s+1} \sum_{h=2}^j \tilde{T}_h^{(i)} - \frac{(m - \bar{T}_i)s(s+1)}{2(N-1)} \quad (6.6)
\end{aligned}$$

Come visto nel capitolo 2, *precision* e *recall* sono misure di valutazione legate ad una singola immagine presa come query. Dal momento che noi vogliamo valutare il *retrieval* rispetto a  $d$  in cui tutte le immagini del database sono considerate una ad una come query, non analizzeremo i valori di *precision* e *recall* per ogni singola immagine, ma analizzeremo la media tra tutte le immagini dei valori di *precision* e *recall* relativi ad un  $k$  fissato, dove ricordiamo che  $k$  in questo caso rappresenta il numero di immagini recuperate. Per non appesantire la notazione, continueremo comunque a scrivere semplicemente *precision* e *recall*, ma si tenga presente che da qui in avanti si farà sempre riferimento alla media di questi valori tra tutte le immagini.

Per quanto riguarda invece l'analisi grafica utilizzeremo il *grafico PR*, di cui abbiamo parlato nel paragrafo (2.1), e la *curva ROC*.

La curva ROC è una curva ottenuta considerando *sensibilità* e *specificità* non come indici percentuali, ma come valori compresi tra 0 e 1 (in sostanza, rispetto al calcolo che abbiamo illustrato precedentemente, non si moltiplica per 100) e riportando sull'asse delle ascisse i valori del complemento a 1 della specificità (*1-specificità*) e sull'asse delle ordinate i valori della sensibilità ottenuti al variare del valore soglia che determina la classificazione dicotomica nevo-melanoma.

Nel caso di un sistema perfetto, ossia un sistema in cui la percentuale di accuratezza diagnostica è uguale al 100%, la curva ROC sale perfettamente verticale sull'asse delle ordinate e piega poi ad angolo retto parallela all'asse delle ascisse.

Nella situazione opposta in cui si verifica che i valori *sensibilità* sono sempre uguali ai valori *1-specificità* la curva ROC corrisponde alla rappresentazione della retta  $x = y$  per valori di  $x \in [0, 1]$ . Questa situazione è chiaramente poco auspicabile. Infatti, prendendo come esempio il nostro caso nevo-melanoma, una situazione di questo tipo comporterebbe che, comunque si ponga il valore soglia, il numero di melanomi riconosciuti sarebbe sempre uguale al numero di nevi non riconosciuti, ovvero possiamo dire “scambiati” per melanomi.

Quindi in conclusione, più la curva si discosterà da questa retta e si avvicinerà all'angolo superiore sinistro, più il sistema sarà valido.

## 6.3 Database 107

Il database a nostra disposizione è formato da  $N = 107$  immagini di lesioni,  $n = 72$  delle quali sono nevi e  $m = 35$  sono melanomi. A partire da questo database abbiamo creato attraverso il momento *off-line* del retrieval tre set di matrici distanza differenti sui cui applicare i vari modelli e metodi illustrati in precedenza.

Il primo set di matrici è composto da  $K = 5$  matrici di distanze reciproche ottenute a partire dalle cinque funzioni misuranti utilizzate dal sistema, ovvero il blu, il rosso, il verde, il bordo e la variazione totale, calcolate su grafi di taglia ottenuti attraverso una “risoluzione 4x”, considerando cioè blocchi di 4 pixel ciascuno.

Il secondo set di matrici è composto da  $K = 5$  matrici di distanze reciproche ottenute sempre a partire dalle cinque funzioni misuranti, calcolate però questa volta su grafi di taglia ottenuti attraverso una “risoluzione 8x”, considerando cioè blocchi di 8 pixel ciascuno.

Il terzo set di matrici è composto invece da  $K = 9$  matrici di distanze reciproche relative ai seguenti nove descrittori (o oggetti  $O^{(k)}$ ): diametro, eccentricità, ellitticità, fattore di forma, circolarità di Haralick, simmetria rispetto all'asse maggiore, simmetria rispetto all'asse minore, simmetria rispetto al baricentro e istogramma dei colori.

Abbiamo fissato il valore  $s = 10$  di *vicini* da considerare per valutare il *retrieval rispetto a d* con gli indici percentuali di specificità, sensibilità e accuratezza diagnostica, e abbiamo deciso di analizzare i valori di precision e recall ottenuti per  $h = 5, 10, 20, 50$ , dove ricordiamo che  $h$  rappresenta il numero di elementi recuperati.

Abbiamo poi considerato per ogni set di matrici un griglia piuttosto fitta di possibili vettori  $\alpha$  e abbiamo calcolato specificità, sensibilità e accuratezza diagnostica corrispondenti al *retrieval rispetto a d*( $\alpha$ ) per tutti i vettori  $\alpha$  della griglia, per valutare le percentuali massime effettivamente raggiungibili.

Infine abbiamo calcolato il vettore  $\alpha$  ottimale attraverso i vari modelli e metodi di risoluzione illustrati e abbiamo valutato, attraverso i metodi descritti nel paragrafo (6.2), il *retrieval rispetto a d*( $\alpha$ ) corrispondente.

Il vettore iniziale da fornire in input ai metodi iterativi è stato fissato uguale a  $x_0 := \bar{\alpha}$ , con  $\bar{\alpha}_k = \frac{1}{K} \forall k = 1, \dots, K$ .

Nelle pagine che seguono riportiamo per ogni set di matrici i seguenti risultati.

1. Massimi indici percentuali di specificità, sensibilità e accuratezza diagnostica ottenuti con la griglia di vettori  $\alpha$ .
2. Tabella di valutazione del *retrieval rispetto a d*( $\alpha$ ) con  $\alpha_k = \frac{1}{K}$ ,  $k = 1, \dots, K$ , poiché al momento, nel sistema di recupero su cui stiamo lavorando, questi sono i valori attribuiti ai pesi considerati.
3. Modello vettoriale:  
 tabelle di valutazione del *retrieval rispetto a d*( $\alpha$ ) con i vettori  $\alpha$  calcolati attraverso le funzioni `gradiente_proiettato_01`, `TrovaAlfa`, `backslash` e `levenberg_marquardt`.

Ricordiamo che nel caso non vincolato il vettore  $\alpha$  utilizzato per effettuare il *retrieval rispetto a*  $d(\alpha)$  si ottiene ponendo uguali a zero le componenti negative del vettore  $\alpha$  calcolato.

4. Modello matriciale:

- calcolo degli indici percentuali di specificità, sensibilità e accuratezza diagnostica relativi al *retrieval rispetto a*  $d(\alpha)$  con i vettori  $\alpha$  calcolati attraverso la funzione `TrovaAlfa`, a partire da quattro terne fissate di valori  $r, s, t$ , per stabilire la terna corrispondente al massimo valore di accuratezza diagnostica;
- tabelle di valutazione del *retrieval rispetto a*  $d(\alpha)$  con i vettori  $\alpha$  calcolati attraverso le funzioni `gradiente_proiettato_01`, `TrovaAlfa`, `backslash` e `levenberg_marquardt`, utilizzando la terna di valori di  $r, s, t$  individuata al punto precedente.

Ricordiamo come sopra la distinzione tra  $\alpha$  calcolato e  $\alpha$  utilizzato.

Riportiamo invece nell'Appendice B tutte le *curve ROC* e tutti i *grafici PR* relativi alle valutazioni effettuate.

### 6.3.1 Matrici 107 4x - 5 funzioni misuranti

1. Percentuali massime ottenute con la griglia di valori di  $\alpha_k$  generati da  $\text{linspace}(0, 1, 25)$ .

- Massima specificità = 94.4%

Dati corrispondenti:

- $\alpha = [0, 0, 0.0417, 0.1617, 0.0417]$

- sensibilità = 65.7%

- accuratezza diagnostica = 85%

- Massima sensibilità = 94.3%

Dati corrispondenti:

- $\alpha = [0.8750, 0.3750, 0.2083, 0.1667, 0.9583]$

- specificità = 88.9%

- accuratezza diagnostica = 90.7%

- **Massima accuratezza diagnostica = 90.7%**

Dati corrispondenti:

- $\alpha = [0, 0.0833, 0.2500, 0.0417, 0.2083]$

- specificità = 91.7%

- sensibilità = 88.6%

2. Matrice globale ottenuta come somma delle 5 matrici con  $\alpha_k = \frac{1}{5}$

specificità	sensibilità	accuratezza diagnostica
86.1%	85.7%	<b>86%</b>

	precision	recall
$h = 5$	0.7570	0.0669
$h = 10$	0.7514	0.1325
$h = 20$	0.7360	0.2569
$h = 50$	0.6972	0.5946

Tabella 6.1: Matrici 107 4x -  $\alpha_k = \frac{1}{5}$

## 3. MODELLO VETTORIALE

precision, recall: valori per  $h$  uguale a 5, 10, 20, 50

	gradiente_proiettato_01	TrovaAlfa	backslash	levenberg_marquardt
$\alpha$ calcolato	0	0	-2.2587	-2.2587
	0.5141	0.3811	0.0527	0.0527
	1.0000	1.0010	2.3829	2.3829
	1.0000	2.8272	2.4689	2.4689
	1.0000	1.5262	2.1222	2.1222
$\alpha$ utilizzato	0	0	0	0
	0.5141	0.3811	0.0527	0.0527
	1.0000	1.0010	2.3829	2.3829
	1.0000	2.8272	2.4689	2.4689
	1.0000	1.5262	2.1222	2.1222
tempo (s)	0.0468	0.0462	7.4560e-05	0.0453
iterazioni	39	/	/	14

Tabella 6.2: Matrici 107 4x - modello vettoriale

	gradiente_proiettato_01	TrovaAlfa	backslash	levenberg_marquardt
specificità	86.1%	90.3%	90.3%	90.3%
sensibilità	85.7%	82.9%	85.7%	85.7%
acc. diagnostica	<b>86%</b>	<b>87.9%</b>	<b>88.8%</b>	<b>88.8%</b>

Tabella 6.3: Matrici 107 4x - modello vettoriale - percentuali

	gradiente_proiettato_01	TrovaAlfa	backslash	levenberg_marquardt
precision	0.7682	0.7514	0.7720	0.7720
	0.7486	0.7234	0.7374	0.7374
	0.7341	0.6991	0.7126	0.7126
	0.6976	0.6495	0.6810	0.6810
recall	0.0674	0.0648	0.0674	0.0674
	0.1308	0.1230	0.1256	0.1256
	0.2541	0.2380	0.2426	0.2426
	0.5942	0.5507	0.5744	0.5744

Tabella 6.4: Matrici 107 4x - modello vettoriale - precision e recall

## 4. MODELLO MATRICIALE

percentuali = specificità, sensibilità, accuratezza diagnostica

(a) **r=0.9 s=0.1 t=0.4**

Soluzione calcolata dalla funzione TrovaAlfa:

- $\alpha = [0, 0.1081, 0.7127, 0.0958, 0.5702]$
- **percentuali = 90.3%, 85.7%, 88.8%**

(b) r=0.99 s=0.01 t=0.01

Soluzione calcolata dalla funzione TrovaAlfa:

- $\alpha = [0, 0.3405, 0.5987, 0, 0.4249]$
- percentuali = 90.3%, 85.7%, 88.8%

(c) r=0.6 s=t=0.4

Soluzione calcolata dalla funzione TrovaAlfa:

- $\alpha = [0.1233, 0.1285, 0.2713, 0.4434, 0.3711]$
- percentuali = 86.1%, 85.7%, 86%

(d) r=0.9 s=0.4 t=0.1

Soluzione calcolata dalla funzione TrovaAlfa:

- $\alpha = [0, 0.3611, 0.4655, 0.4232, 0.4080]$
- percentuali = 86.1%, 85.7%, 86%

→ matrice  $\mu$  costruita con  $r = 0.9, s = 0.1, t = 0.4$

precision, recall: valori per  $h$  uguale a 5, 10, 20, 50

	gradiente_proiettato_01	TrovaAlfa	backslash	levenberg_marquardt
$\alpha$ calcolato	0	0	-0.2757	-0.2757
	0.1081	0.1081	0.0014	0.0014
	0.7127	0.7127	1.0282	1.0282
	0.0958	0.0958	0.1603	0.1603
	0.5702	0.5702	0.6394	0.6394
$\alpha$ utilizzato	0	0	0	0
	0.1081	0.1081	0.0014	0.0014
	0.7127	0.7127	1.0282	1.0282
	0.0958	0.0958	0.1603	0.1603
	0.5702	0.5702	0.6394	0.6394
tempo (s)	35.2132	0.0672	0.0014	0.2133
iterazioni	14971	/	/	7

Tabella 6.5: Matrici 107 4x - modello matriciale

	gradiente_proiettato_01	TrovaAlfa	backslash	levenberg_marquardt
specificità	90.3%	90.3%	90.3%	90.3%
sensibilità	85.7%	85.7%	82.9%	82.9%
acc. diagnostica	<b>88.8%</b>	<b>88.8%</b>	<b>87.9%</b>	<b>87.9%</b>

Tabella 6.6: Matrici 107 4x - modello matriciale - percentuali

	gradiente_proiettato_01	TrovaAlfa	backslash	levenberg_marquardt
precision	0.7850	0.7850	0.7701	0.7701
	0.7682	0.7682	0.7701	0.7701
	0.7514	0.7514	0.7481	0.7481
	0.7164	0.7164	0.7118	0.7118
recall	0.0690	0.0690	0.0673	0.0673
	0.1343	0.1343	0.1344	0.1344
	0.2608	0.2608	0.2586	0.2586
	0.6080	0.6080	0.6021	0.6021

Tabella 6.7: Matrici 107 4x - modello matriciale - precision e recall

### 6.3.2 Matrici 107 8x - 5 funzioni misuranti

1. Percentuali massime ottenute con la griglia di valori di  $\alpha_k$  generati da  $\text{linspace}(0, 1, 25)$ .

- Massima specificità = 94.4%  
Dati corrispondenti:
  - $\alpha = [0.0833, 0.0417, 0.1250, 0.8333, 0.0833]$
  - sensibilità = 60%
  - accuratezza diagnostica = 83.2%
- Massima sensibilità = 85.7%  
Dati corrispondenti:
  - $\alpha = [0, 0, 0.1250, 0.0417, 0]$
  - specificità = 88.9%
  - accuratezza diagnostica = 87.9%
- **Massima accuratezza diagnostica = 90.7%**  
Dati corrispondenti:
  - $\alpha = [0, 0.0833, 0.7083, 0.2083, 0.5833]$
  - specificità = 93.1%
  - sensibilità = 85.7%

2. Matrice globale ottenuta come somma delle 5 matrici con  $\alpha_k = \frac{1}{5}$

specificità	sensibilità	accuratezza diagnostica
87.5%	82.9%	<b>86%</b>

	precision	recall
$h = 5$	0.7645	0.0672
$h = 10$	0.7542	0.1327
$h = 20$	0.7360	0.2569
$h = 50$	0.7009	0.5989

Tabella 6.8: Matrici 107 8x -  $\alpha_k = \frac{1}{5}$

## 3. MODELLO VETTORIALE

precision, recall: valori per  $h$  uguale a 5, 10, 20, 50

	gradiente_proiettato_01	TrovaAlfa	backslash	levenberg_marquardt
$\alpha$ calcolato	0	0	-1.6774	-1.6774
	0.5287	0.3761	0.2050	0.2050
	1.0000	1.0971	2.4217	2.4217
	1.0000	2.3625	1.8556	1.8556
	1.0000	1.2908	1.3425	1.3425
$\alpha$ utilizzato	0	0	0	0
	0.5287	0.3761	0.2050	0.2050
	1.0000	1.0971	2.4217	2.4217
	1.0000	2.3625	1.8556	1.8556
	1.0000	1.2908	1.3425	1.3425
tempo (s)	0.0405	0.2268	1.4750e-04	0.0495
iterazioni	43	/	/	12

Tabella 6.9: Matrici 107 8x - modello vettoriale

	gradiente_proiettato_01	TrovaAlfa	backslash	levenberg_marquardt
specificità	91.7%	91.7%	91.7%	91.7%
sensibilità	77.1%	68.6%	77.1%	77.1%
acc. diagnostica	<b>86.9%</b>	<b>84.1%</b>	<b>86.9%</b>	<b>86.9%</b>

Tabella 6.10: Matrici 107 8x - modello vettoriale - percentuali

	gradiente_proiettato_01	TrovaAlfa	backslash	levenberg_marquardt
precision	0.7701	0.7364	0.7626	0.7626
	0.7477	0.7336	0.7561	0.7561
	0.7350	0.7103	0.7294	0.7294
	0.6927	0.6542	0.6895	0.6895
recall	0.0670	0.0632	0.0666	0.0666
	0.1295	0.1250	0.1311	0.1311
	0.2530	0.2426	0.2500	0.2500
	0.5920	0.5550	0.5876	0.5876

Tabella 6.11: Matrici 107 8x - modello vettoriale - precision e recall

## 4. MODELLO MATRICIALE

percentuali = specificità, sensibilità, accuratezza diagnostica

(a) **r=0.9 s=0.1 t=0.4**

Soluzione calcolata dalla funzione TrovaAlfa:

- $\alpha = [0, 0.0432, 0.8157, 0.1667, 0.4551]$
- **percentuali = 93.1%, 77.1%, 87.9%**

(b) r=0.99 s=0.01 t=0.01

Soluzione calcolata dalla funzione TrovaAlfa:

- $\alpha = [0, 0.2560, 0.7601, 0.0200, 0.3412]$
- percentuali = 88.9%, 80%, 86%

(c) r=0.6 s=t=0.4

Soluzione calcolata dalla funzione TrovaAlfa:

- $\alpha = [0.0804, 0.0462, 0.3775, 0.5078, 0.3406]$
- percentuali = 91.7%, 71.4%, 85%

(d) r=0.9 s=0.4 t=0.1

Soluzione calcolata dalla funzione TrovaAlfa:

- $\alpha = [0, 0.2601, 0.5739, 0.4896, 0.3452]$
- percentuali = 88.9%, 80%, 86%

→ matrice  $\mu$  costruita con  $r = 0.9, s = 0.1, t = 0.4$

precision, recall: valori per  $h$  uguale a 5, 10, 20, 50

	gradiente_proiettato_01	TrovaAlfa	backslash	levenberg_marquardt
$\alpha$ calcolato	0	0	-0.4148	-0.4148
	0.0432	0.0432	-0.0791	-0.0791
	0.8157	0.8157	1.2772	1.2772
	0.1667	0.1667	0.2244	0.2244
	0.4551	0.4551	0.5239	0.5239
$\alpha$ utilizzato	0	0	0	0
	0.0432	0.0432	0	0
	0.8157	0.8157	1.2772	1.2772
	0.1667	0.1667	0.2244	0.2244
	0.4551	0.4551	0.5239	0.5239
tempo (s)	24.4770	0.0411	0.0014	0.2180
iterazioni	10879	/	/	8

Tabella 6.12: Matrici 107 8x - modello matriciale

	gradiente_proiettato_01	TrovaAlfa	backslash	levenberg_marquardt
specificità	93.1%	93.1%	91.7%	91.7%
sensibilità	77.1%	77.1%	80%	80%
acc. diagnostica	<b>87.9%</b>	<b>87.9%</b>	<b>87.9%</b>	<b>87.9%</b>

Tabella 6.13: Matrici 107 8x - modello matriciale - percentuali

	gradiente_proiettato_01	TrovaAlfa	backslash	levenberg_marquardt
precision	0.7850	0.7850	0.7869	0.7869
	0.7598	0.7598	0.7645	0.7645
	0.7388	0.7388	0.7449	0.7449
	0.7056	0.7056	0.7114	0.7114
recall	0.0692	0.0692	0.0697	0.0697
	0.1321	0.1321	0.1342	0.1342
	0.2555	0.2555	0.2587	0.2587
	0.5982	0.5982	0.6034	0.6034

Tabella 6.14: Matrici 107 8x - modello matriciale - precision e recall

### 6.3.3 Matrici 107 - 9 descrittori

1. Percentuali massime ottenute con la griglia di valori di  $\alpha_k$  generati da  $\text{linspace}(0, 1, 6)$ .

- Massima specificità = 90.3%

Dati corrispondenti:

- $\alpha = [0, 0.4000, 0.6000, 0, 0.2000, 0.2000, 0.2000, 1.0000, 0]$
- sensibilità = 62.9%
- accuratezza diagnostica = 81.3%

- Massima sensibilità = 88.6%

Dati corrispondenti:

- $\alpha = [0, 0, 0.2000, 0, 0.8000, 0.2000, 0, 0, 0.4000]$
- specificità = 65.3%
- accuratezza diagnostica = 72.9%

- **Massima accuratezza diagnostica = 84.1%**

Dati corrispondenti:

- $\alpha = [0, 0.2000, 0.2000, 0.4000, 1.0000, 0.6000, 0, 0, 0.2000]$
- specificità = 83.4%
- sensibilità = 85.7%

2. Matrice globale ottenuta come somma delle 9 matrici con  $\alpha_k = \frac{1}{9}$

specificità	sensibilità	accuratezza diagnostica
76.4%	57.1%	<b>70.1%</b>

	precision	recall
$h = 5$	0.6673	0.0576
$h = 10$	0.6570	0.1126
$h = 20$	0.6364	0.2162
$h = 50$	0.6034	0.5103

Tabella 6.15: Matrici 107 9desc -  $\alpha_k = \frac{1}{9}$

## 3. MODELLO VETTORIALE

precision, recall: valori per  $h$  uguale a 5, 10, 20, 50

	gradiente_proiettato_01	TrovaAlfa	backslash	levenberg_marquardt
$\alpha$ calcolato	1.0000	12.2190	12.2190	12.2190
	1.0000	1.2454	1.2454	1.2454
	1.0000	1.8260	1.8260	1.8260
	1.0000	2.3399	2.3399	2.3399
	1.0000	4.3635	4.3635	4.3635
	1.0000	2.5425	2.5425	2.5425
	1.0000	1.9390	1.9390	1.9390
	1.0000	2.8215	2.8215	2.8215
	1.0000	0.4328	0.4328	0.4328
$\alpha$ utilizzato	1.0000	12.2190	12.2190	12.2190
	1.0000	1.2454	1.2454	1.2454
	1.0000	1.8260	1.8260	1.8260
	1.0000	2.3399	2.3399	2.3399
	1.0000	4.3635	4.3635	4.3635
	1.0000	2.5425	2.5425	2.5425
	1.0000	1.9390	1.9390	1.9390
	1.0000	2.8215	2.8215	2.8215
	1.0000	0.4328	0.4328	0.4328
tempo (s)	0.0248	0.0149	1.1143e-04	0.1176
iterazioni	27	/	/	27

Tabella 6.16: Matrici 107 9desc - modello vettoriale

	gradiente_proiettato_01	TrovaAlfa	backslash	levenberg_marquardt
specificità	76.4%	62.5%	62.5%	62.5%
sensibilità	57.1%	51.4%	51.4%	51.4%
acc. diagnostica	<b>70.1%</b>	<b>58.9%</b>	<b>58.9%</b>	<b>58.9%</b>

Tabella 6.17: Matrici 107 9desc - modello vettoriale - percentuali

	gradiente_proiettato_01	TrovaAlfa	backslash	levenberg_marquardt
precision	0.6673	0.5963	0.5963	0.5963
	0.6570	0.5981	0.5981	0.5981
	0.6364	0.5972	0.5972	0.5972
	0.6034	0.5742	0.5742	0.5742
recall	0.0576	0.0497	0.0497	0.0497
	0.1126	0.1013	0.1013	0.1013
	0.2162	0.2033	0.2033	0.2033
	0.5103	0.4862	0.4862	0.4862

Tabella 6.18: Matrici 107 9desc - modello vettoriale - precision e recall

## 4. MODELLO MATRICIALE

percentuali = specificità, sensibilità, accuratezza diagnostica

(a)  $r=0.9$   $s=0.1$   $t=0.4$

Soluzione calcolata dalla funzione TrovaAlfa:

- $\alpha = [0.0419, 0.5923, 0, 0.2228, 0, 0.4795, 0.1322, 0, 0]$
- percentuali = 80.6%, 62.9%, 74.8%

(b)  $r=0.99$   $s=0.01$   $t=0.01$

Soluzione calcolata dalla funzione TrovaAlfa:

- $\alpha = [0, 0.6202, 0, 0.1680, 0, 0.4569, 0.0978, 0, 0]$
- percentuali = 80.6%, 62.9%, 74.8%

(c)  **$r=0.6$   $s=t=0.4$**

Soluzione calcolata dalla funzione TrovaAlfa:

- $\alpha = [0.1524, 0.3184, 0.1623, 0.1586, 0, 0.5098, 0.1256, 0, 0]$
- **percentuali = 81.9%, 65.7%, 76.6%**

(d)  $r=0.9$   $s=0.4$   $t=0.1$

Soluzione calcolata dalla funzione TrovaAlfa:

- $\alpha = [0.1466, 0.4699, 0.1596, 0.1570, 0, 0.6440, 0.1286, 0, 0]$
- percentuali = 79.2%, 68.6%, 75.7%

→ matrice  $\mu$  costruita con  $r = 0.6, s = 0.4, t = 0.4$

precision, recall: valori per  $h$  uguale a 5, 10, 20, 50

	gradiente_proiettato_01	TrovaAlfa	backslash	levenberg_marquardt
$\alpha$ calcolato	0.1524	0.1524	0.1569	0.1569
	0.3184	0.3184	0.3130	0.3130
	0.1623	0.1623	0.1606	0.1606
	0.1586	0.1586	0.1875	0.1875
	0	0	0.0007	0.0007
	0.5098	0.5098	0.5114	0.5114
	0.1256	0.1256	0.1448	0.1448
	0	0	-0.0968	-0.0968
	0	0	0.0435	0.0435
$\alpha$ utilizzato	0.1524	0.1524	0.1569	0.1569
	0.3184	0.3184	0.3130	0.3130
	0.1623	0.1623	0.1606	0.1606
	0.1586	0.1586	0.1875	0.1875
	0	0	0.0007	0.0007
	0.5098	0.5098	0.5114	0.5114
	0.1256	0.1256	0.1448	0.1448
	0	0	0	0
	0	0	0.0435	0.0435
tempo (s)	51.5043	36.8416	0.0031	0.3224
iterazioni	15484	/	/	6

Tabella 6.19: Matrici 107 9desc - modello matriciale

	gradiente_proiettato_01	TrovaAlfa	backslash	levenberg_marquardt
specificità	81.9%	81.9%	84.7%	84.7%
sensibilità	65.7%	65.7%	60%	60%
acc. diagnostica	<b>76.6%</b>	<b>76.6%</b>	<b>76.6%</b>	<b>76.6%</b>

Tabella 6.20: Matrici 107 9desc - modello matriciale - percentuali

	gradiente_proiettato_01	TrovaAlfa	backslash	levenberg_marquardt
precision	0.7290	0.7290	0.7271	0.7271
	0.7103	0.7103	0.7047	0.7047
	0.6944	0.6944	0.6864	0.6864
	0.6370	0.6370	0.6353	0.6353
recall	0.0634	0.0634	0.0631	0.0631
	0.1230	0.1230	0.1219	0.1219
	0.2410	0.2410	0.2372	0.2372
	0.5414	0.5414	0.5391	0.5391

Tabella 6.21: Matrici 107 9desc - modello matriciale - precision e recall

## 6.4 Osservazioni

Nei tre set di matrici considerati la massima accuratezza diagnostica, calcolata attraverso la griglia di possibili vettori  $\alpha$ , non è mai pari al 100%. In particolare nei primi due set di matrici è pari al 90.7%, mentre nel terzo set non supera l'84.1%.

Per valutare questo risultato, non bisogna dimenticare che l'accuratezza diagnostica fa riferimento alla diagnosi dicotomica nevo-melanoma che, come già più volte sottolineato, siamo obbligati ad utilizzare, ma che in realtà non rappresenta il vero obiettivo di questo sistema di recupero.

Infatti il dott. Ivan Tomba, dopo aver visionato le immagini, ci ha confermato che nella maggior parte di quei casi in cui la classificazione fornita dal sistema risulta errata, le dieci immagini recuperate dal retrieval sembrano però essere "rilevanti" per l'immagine considerata come query.

In pratica, se un nevo ha caratteristiche simili a un melanoma e il sistema di recupero restituisce tra i dieci *vicini* molte immagini di melanomi, è per noi un risultato accettabile, in quanto si aspira ad un sistema di recupero che restituisca le immagini più rilevanti.

Osserviamo che le percentuali ed i valori di precision e recall sono migliori nei set di matrici provenienti dalle funzioni misuranti.

Anche l'analisi grafica mette in luce questa differenza. Infatti, le curve ROC relative al terzo set di matrici sono molto più vicine alla diagonale  $x = y$  rispetto alle curve ROC relative agli altri due set, che hanno invece un andamento molto più simile a quello sperato.

Nel modello matriciale il metodo del gradiente proiettato ha impiegato molte iterazioni per calcolare la soluzione. In questo caso, abbiamo valutato di considerare anche una tolleranza più bassa,  $\text{tol1}=1.e-5$ . L'effetto immediato è stato un dimezzamento del numero di iterazioni effettuate ed una piccola variazione sul valore della soluzione calcolata.

Abbiamo analizzato se questa variazione potesse produrre una diversa valutazione del *retrieval* corrispondente e abbiamo osservato che nel caso matriciale del set di matrici relativo ai nove descrittori, la percentuale di accuratezza diagnostica e i valori di precision e recall diminuivano leggermente.

Dato che questi metodi verranno utilizzati dal momento *off-line* del retrieval che non necessita dell'istantaneità della risposta, propria invece del momento *real time*, abbiamo deciso di preferire l'accuratezza alla velocità e abbiamo quindi mantenuto  $\text{tol1}=1.e-6$  come valore per la tolleranza.

Il modello matriciale restituisce parametri di valutazione a volte migliori rispetto al modello vettoriale. Bisogna però tenere presente che nel modello matriciale abbiamo avuto la possibilità di scegliere i valori  $r, s, t$  sulla base della terna che è risultata migliore delle altre. Inoltre, la configurazione individuata non è stata la stessa in tutti e tre i set di matrici.

Per questi motivi, riteniamo il modello vettoriale più stabile e affidabile.

Successivamente, abbiamo utilizzato anche un secondo database composto da  $N = 202$  immagini,  $m = 60$  delle quali melanomi e  $n = 142$  nevi. I risultati ottenuti utilizzando questo database sono risultati meno soddisfacenti rispetto ai risultati ottenuti con il database di 107 immagini.

Il dott. Ivan Tomba ci ha consigliato di diffidare del database di 202 immagini sia perché le immagini contenute non sono state catturate tutte allo stesso modo, sia perché circa l'80% delle immagini contenute sono relative

a lesioni molto dubbie. Per questi motivi abbiamo deciso di sospendere l'analisi.

# Conclusioni

In questa tesi abbiamo proposto due diversi modelli per la stima dei parametri  $\alpha_k$  da utilizzare nel momento *off-line* del retrieval affinché il sistema di recupero sia ottimizzato.

Questi modelli sono risultati adatti allo scopo di migliorare la situazione attuale. Infatti, nei vari set di matrici considerati, siamo sempre riusciti a calcolare almeno un vettore  $\alpha$  per cui il *retrieval rispetto a  $d(\alpha)$*  abbia una percentuale di accuratezza diagnostica e valori di precision e recall superiori rispetto a quelli ottenuti considerando il vettore  $\alpha$  con componenti  $\alpha_k = \frac{1}{K}$ ,  $k = 1, \dots, K$ .

I responsabili del sistema di recupero potranno valutare di inserire nel momento *off-line* solo alcuni dei modelli e metodi proposti o eventualmente di considerarli tutti, al fine di poter utilizzare poi solo il vettore  $\alpha$  corrispondente ai risultati migliori.

Nel momento in cui si avrà a disposizione un giudizio di rilevanza fornito da dermatologi si potranno avere sviluppi sia per quanto riguarda i modelli proposti, sia per quanto riguarda i metodi utilizzabili per la loro risoluzione.



# Appendice A

## Omologia persistente

L'omologia persistente è uno strumento matematico che affronta i problemi relativi all'analisi e al confronto della forma da un punto di vista geometrico-topologico.

Questa teoria è stata sviluppata indipendentemente da tre diversi gruppi di ricerca. I suoi concetti si ritrovano infatti all'interno del progetto di biogeometria di Herbert Edelsbrunner a Duke, Carolina del Nord, nel lavoro di dottorato di Vanessa Robins alla Boulder University, Colorado, e nei lavori di Patrizio Frosini e Massimo Ferri sotto il nome di Teoria della Taglia, all'Università di Bologna.

Dato lo sviluppo indipendente che si è verificato, ogni gruppo di ricerca ha introdotto le proprie notazioni e definizioni e, per quanto i vari risultati si possano tradurre nei linguaggi scelti dall'uno o dall'altro, questo fatto può creare qualche problema a chi si interessa al contenuto di questo utilissimo strumento per la prima volta.

Per questo motivo, al fine di rimanere coerenti con le terminologie già utilizzate all'interno di questo lavoro, abbiamo deciso di utilizzando il linguaggio della *Teoria della Taglia* per introdurre brevemente i concetti fondamentali di questa teoria.

Il lettore che fosse interessato ad una trattazione più dettagliata può fare riferimento ai lavori di P. Frosini e dei suoi collaboratori [12, 13, 14].

Il lettore che fosse invece interessato ad un linguaggio più vicino ai lavori degli altri gruppi di ricerca sull'omologia persistente, può fare riferimento a [11].

## A.1 Distanze di taglia

Come già detto in precedenza, la teoria della taglia studia i problemi relativi all'analisi e al confronto della forma da un punto di vista geometrico-topologico.

Il modo più naturale di pensare al concetto di forma è quello di considerare una  $n$ -varietà  $\mathcal{M}$  compatta e connessa con una funzione continua  $\varphi$  a valori reali definita su di essa. La coppia  $(\mathcal{M}, \varphi)$  è detta *coppia di taglia* e la funzione  $\varphi : \mathcal{M} \rightarrow \mathbb{R}$  è detta *funzione misurante*.

La varietà  $\mathcal{M}$  rappresenta quindi l'oggetto di cui ci interessa la forma, mentre la funzione misurante  $\varphi$  è scelta in modo arbitrario a seconda delle caratteristiche oggetto di studio.

**Definizione A.1.** Siano  $(\mathcal{M}, \varphi)$  e  $(\mathcal{N}, \psi)$  due coppie di taglia e sia  $\mathcal{H}(\mathcal{M}, \mathcal{N})$  l'insieme degli omeomorfismi da  $\mathcal{M}$  a  $\mathcal{N}$ . Se  $\mathcal{H}(\mathcal{M}, \mathcal{N}) \neq \emptyset$ , definiamo la funzione  $\Theta : \mathcal{H}(\mathcal{M}, \mathcal{N}) \rightarrow \mathbb{R}$  come la funzione che ad ogni omeomorfismo  $f \in \mathcal{H}(\mathcal{M}, \mathcal{N})$  associa il numero reale:

$$\Theta(f) = \max_{P \in \mathcal{M}} |\varphi(P) - \psi(f(P))|$$

La funzione  $\Theta$  è detta la *misura naturale* su  $\mathcal{H}(\mathcal{M}, \mathcal{N})$  relativa alle funzioni misuranti  $\varphi$  e  $\psi$ .

**Definizione A.2.** Siano  $(\mathcal{M}, \varphi)$  e  $(\mathcal{N}, \psi)$  due coppie di taglia. Allora, si definisce la *pseudo-distanza naturale* tra  $(\mathcal{M}, \varphi)$  e  $(\mathcal{N}, \psi)$  come segue:

$$d((\mathcal{M}, \varphi), (\mathcal{N}, \psi)) = \begin{cases} \inf_{f \in \mathcal{H}(\mathcal{M}, \mathcal{N})} \Theta(f) & \text{se } \mathcal{H}(\mathcal{M}, \mathcal{N}) \neq \emptyset \\ +\infty & \text{altrimenti} \end{cases}$$

*Osservazione 14.*

Si può verificare che la definizione precedente da effettivamente una pseudometrica sull'insieme delle coppie di taglia, nel senso che date due coppie  $(\mathcal{M}, \varphi)$ ,  $(\mathcal{N}, \psi)$  tali che  $d((\mathcal{M}, \varphi), (\mathcal{N}, \psi)) = 0$  non necessariamente  $(\mathcal{M}, \varphi) = (\mathcal{N}, \psi)$  (dove per coppie di taglia uguali si intende che  $\mathcal{M} = \mathcal{N}$  e le funzioni misuranti  $\varphi, \psi$  coincidono). Soddisfa invece la simmetria, la disuguaglianza triangolare e si annulla per coppie uguali.

Da un punto di vista applicativo, è facile comprendere la difficoltà connessa al calcolo della pseudo-distanza naturale tra due coppie di taglia, in quanto richiede lo studio di tutti gli omeomorfismi tra le varietà.

Per questo motivo, passiamo ora ad introdurre uno strumento più facilmente calcolabile: le *funzioni di taglia*.

## A.2 Funzioni di taglia

In questo paragrafo concentreremo la nostra attenzione sulle funzioni di taglia ed in particolare sulle funzioni di taglia ridotte, con lo scopo di essere poi in grado nel paragrafo successivo di definire la distanza di corrispondenza, più conosciuta in letteratura come distanza bottleneck.

Iniziamo però introducendo alcune definizioni per arrivare alla definizione classica di funzione di taglia.

D'ora in poi assumiamo sempre data una coppia di taglia  $(\mathcal{M}, \varphi)$ .

**Definizione A.3.** Per ogni  $y \in \mathbb{R}$  definiamo la relazione  $\cong_{\varphi \leq y}$  su  $\mathcal{M}$  in modo tale che, dati  $P, Q \in \mathcal{M}$ ,  $P \cong_{\varphi \leq y} Q$  se e solo se  $P = Q$  oppure se e solo se esiste un cammino continuo  $\gamma : [0, 1] \rightarrow \mathcal{M}$  tale che  $\gamma(0) = P$  e  $\gamma(1) = Q$  e  $\varphi(\gamma(\tau)) \leq y$  per ogni  $\tau \in [0, 1]$ .

*Osservazione 15.* La relazione  $\cong_{\varphi \leq y}$  è una relazione di equivalenza su  $\mathcal{M}$  per ogni  $y \in \mathbb{R}$ .

**Definizione A.4.** Per ogni  $x \in \mathbb{R}$  indichiamo con  $\mathcal{M}\langle \varphi \leq x \rangle$  l'insieme  $\{P \in \mathcal{M} : \varphi(P) \leq x\}$ .

**Definizione A.5.** Consideriamo la funzione  $l_{(\mathcal{M},\varphi)} : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{N} \cup \{\infty\}$  definita in modo tale che  $l_{(\mathcal{M},\varphi)}(x, y)$  sia uguale al numero di classi di equivalenza di  $\mathcal{M}\langle\varphi \leq x\rangle$  rispetto alla relazione di equivalenza  $\cong_{\varphi \leq y}$ .

Questa funzione si dice *funzione di taglia* associata alla coppia di taglia  $(\mathcal{M}, \varphi)$ .

*Osservazione 16.*

Quando  $x \leq y$  il valore della funzione di taglia  $l_{(\mathcal{M},\varphi)}(x, y)$  può essere interpretato geometricamente come il numero di componenti connesse per archi di  $\mathcal{M}\langle\varphi \leq y\rangle$  contenenti almeno un punto di  $\mathcal{M}\langle\varphi \leq x\rangle$ .

Passiamo ora alla definizione di funzione di taglia ridotta.

**Definizione A.6.** Dati due punti  $P$  e  $Q$  appartenenti ad  $\mathcal{M}$  si dice che sono  $\langle\varphi \leq y\rangle$ -connessi se e solo se  $P = Q$  o se e solo se esiste una componente connessa di  $\mathcal{M}\langle\varphi \leq y\rangle$  che li contiene entrambi.

*Osservazione 17.* La relazione  $\langle\varphi \leq y\rangle$ -connessione è una relazione di equivalenza su  $\mathcal{M}$  per ogni  $y \in \mathbb{R}$ .

Introduciamo la seguente notazione:

$$\Delta = \{(x, y) \in \mathbb{R}^2 : x = y\}, \quad \Delta^+ = \{(x, y) \in \mathbb{R}^2 : x < y\}$$

**Definizione A.7.** Consideriamo la funzione  $l_{(\mathcal{M},\varphi)}^* : \Delta^+ \rightarrow \mathbb{N}$  definita in modo tale che  $l_{(\mathcal{M},\varphi)}^*(x, y)$  sia uguale al numero di classi di equivalenza di  $\mathcal{M}\langle\varphi \leq x\rangle$  rispetto alla relazione di equivalenza  $\langle\varphi \leq y\rangle$ -connessione.

Questa funzione si dice *funzione di taglia ridotta* associata alla coppia di taglia  $(\mathcal{M}, \varphi)$ .

In generale, è preferibile lavorare con le funzioni di taglia ridotte in quanto conservano tutte le proprietà fondamentali delle funzioni di taglia e permettono di evitare alcuni aspetti tecnici, semplificando così un gran numero di dimostrazioni.

Da un punto di vista matematico, la differenza risiede nell'uso della connessione al posto della connessione per archi e nella restrizione del dominio da  $\mathbb{R}^2$  a  $\Delta^+$ .

La grande forza delle funzioni di taglia e delle funzioni di taglia ridotte risiede nel fatto che permettono di trasformare il problema del confronto tra forme in un problema di confronto tra funzioni.

Per poter arrivare a definire un buon metodo di confronto tra queste funzioni dobbiamo prima di tutto mostrare come si possono trasformare in oggetti ancora più semplici, precisamente in sequenze di punti.

### A.3 Punti angolari e distanza bottleneck

Da qui in avanti faremo sempre riferimento a funzioni di taglia ridotte.

**Definizione A.8.** Per ogni punto  $p = (x, y) \in \Delta^+$  definiamo  $\mu(p)$ , molteplicità di  $p$ , come il minimo su tutti i numeri reali positivi  $\epsilon$ , con  $x + \epsilon < y - \epsilon$ , di

$$l_{(\mathcal{M}, \varphi)}^*(x + \epsilon, y - \epsilon) - l_{(\mathcal{M}, \varphi)}^*(x - \epsilon, y - \epsilon) - l_{(\mathcal{M}, \varphi)}^*(x + \epsilon, y + \epsilon) + l_{(\mathcal{M}, \varphi)}^*(x - \epsilon, y + \epsilon)$$

Se  $\mu(p) > 0$  il punto  $p \in \Delta^+$  è detto *punto angolare proprio* di  $l_{(\mathcal{M}, \varphi)}^*$ .

**Definizione A.9.** Per ogni retta verticale  $r$  di equazione  $x = k$ , definiamo  $\mu(r)$ , molteplicità di  $r$ , come il minimo su tutti i numeri reali positivi  $\epsilon$ , con  $k + \epsilon < \frac{1}{\epsilon}$ , di

$$l_{(\mathcal{M}, \varphi)}^*(k + \epsilon, \frac{1}{\epsilon}) - l_{(\mathcal{M}, \varphi)}^*(k - \epsilon, \frac{1}{\epsilon})$$

Se  $\mu(r) > 0$  la retta  $r$  è detta *punto angolare all'infinito* o *retta angolare* di  $l_{(\mathcal{M}, \varphi)}^*$ .

Introduciamo ora la seguente notazione:

$$\Delta^* = \Delta^+ \cup \{(k, \infty) : k \in \mathbb{R}\}$$

A questo punto possiamo enunciare il seguente teorema di rappresentazione.

#### Teorema A.3.1.

Per ogni punto  $(\bar{x}, \bar{y}) \in \Delta^+$  si ha:

$$l_{(\mathcal{M}, \varphi)}^*(\bar{x}, \bar{y}) = \sum_{(x, y) \in \Delta^*, x \leq \bar{x}, y > \bar{y}} \mu((x, y))$$

Il risultato precedente mette in relazione le funzioni di taglia ridotte con le serie formali.

Indichiamo ora semplicemente con  $l^*$  una funzione di taglia ridotta e consideriamo  $\bar{\Delta}^*$ , definito a partire da  $\bar{\Delta}^+$  in modo analogo a  $\Delta^*$ .

**Definizione A.10.** Si definisce *sequenza rappresentativa* per  $l^*$  ogni sequenza di punti  $a : \mathbb{N} \rightarrow \bar{\Delta}^*$  (indicati brevemente con  $a_i$ ) con le seguenti proprietà:

1.  $a_0$  è una retta angolare per  $l^*$ ;
2. per ogni  $i > 0$ ,  $a_i$  è un punto angolare proprio per  $l^*$  oppure  $a_i \in \Delta$ ;
3. se  $p$  è un punto angolare proprio per  $l^*$  con molteplicità  $\mu(p)$ , allora la cardinalità dell'insieme  $\{i \in \mathbb{N} : a_i = p\}$  è uguale a  $\mu(p)$ ;
4. l'insieme degli indici per cui  $a_i$  appartiene a  $\Delta$  è numerabile.

Definiamo ora una pseudometrica in  $\bar{\Delta}^*$  che darà luogo a una distanza tra funzioni di taglia ridotte, la distanza bottleneck, obiettivo e punto di arrivo di questa piccola appendice.

**Definizione A.11.** Per ogni  $(x, y), (x', y')$  in  $\bar{\Delta}^*$  definiamo  $d^*$  *pseudo-distanza* in  $\bar{\Delta}^*$  come segue:

$$d^*((x, y), (x', y')) = \min \left\{ \max\{|x - x'|, |y - y'|\}, \max \left\{ \frac{y - x}{2}, \frac{y' - x'}{2} \right\} \right\}$$

con le seguenti convenzioni su  $\infty$ :

- $\infty - y = y - \infty = \infty$ , per  $y \neq \infty$
- $\infty - \infty = 0$ ;  $\frac{\infty}{2} = \infty$ ;  $|\infty| = \infty$
- $\min\{\infty, c\} = c$ ;  $\max\{\infty, c\} = \infty$ .

**Definizione A.12.** Se  $(a_i)$  e  $(b_i)$  sono due sequenze rappresentative per le funzioni di taglia  $l_1^*$  e  $l_2^*$  rispettivamente, allora la *distanza di corrispondenza* o *distanza bottleneck* tra  $l_1^*$  e  $l_2^*$  è il numero

$$d_{match}(l_1^*, l_2^*) := \min_{\sigma} \max_i d^*(a_i, b_{\sigma(i)})$$

dove  $i$  varia in  $\mathbb{N}$  e  $\sigma$  varia tra tutte le possibili biiezioni da  $\mathbb{N}$  a  $\mathbb{N}$ .

Questa definizione è indipendente dalla scelta della sequenza rappresentativa di punti per le funzioni di taglia ridotte  $l_1^*$  e  $l_2^*$ .

In più è importante sottolineare che la distanza di corrispondenza  $d_{match}$  è effettivamente una distanza, non solo una pseudo-distanza, tra le funzioni di taglia ridotte.

Per concludere riportiamo il seguente risultato che mette in relazione la pseudo-distanza naturale tra coppie di taglia e la distanza di corrispondenza.

**Teorema A.3.2.**

*Siano  $(\mathcal{M}, \varphi)$  e  $(\mathcal{N}, \psi)$  due coppie di taglia, con  $\mathcal{M}$  e  $\mathcal{N}$  omeomorfe.*

*Allora:*

$$d((\mathcal{M}, \varphi), (\mathcal{N}, \psi)) \geq d_{match}(l_{(\mathcal{M}, \varphi)}^*, l_{(\mathcal{N}, \psi)}^*)$$



# Appendice B

## Grafici

Riportiamo nelle pagine seguenti i *grafici PR* e le *curve ROC* relativi ai modelli e ai metodi analizzati nel paragrafo (6.3).

Nel caso non vincolato trattiamo insieme le funzioni `levenberg_marquardt` e `backslash` visto che hanno calcolato sempre soluzioni uguali.

## B.1 Matrici 107 4x - 5 funzioni misuranti

### Modello vettoriale

- *Gradiente proiettato*

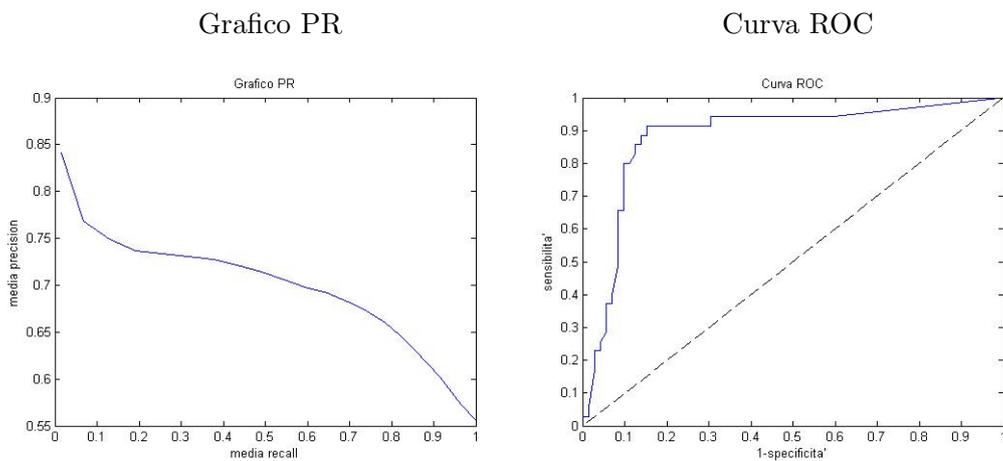


Figura B.1: Matrici 107 4x - mod.vettoriale - gradiente\_proiettato\_01

- *TrovaAlfa*

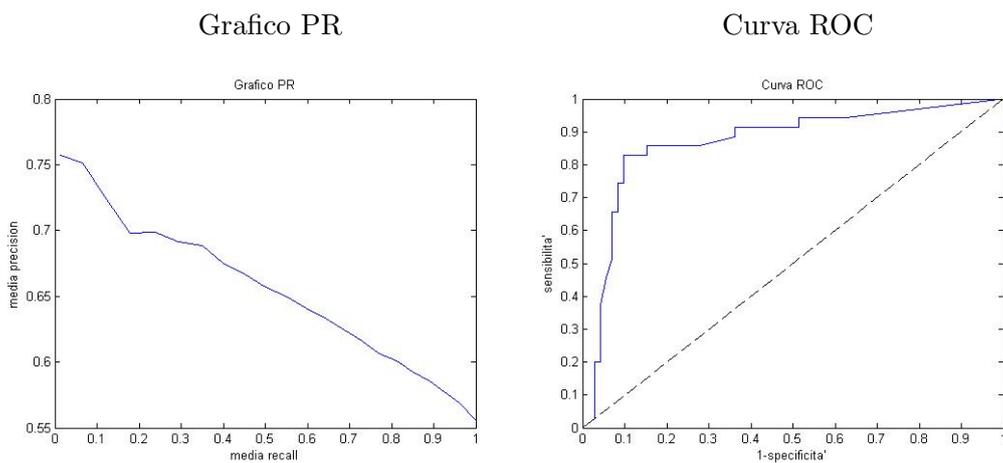
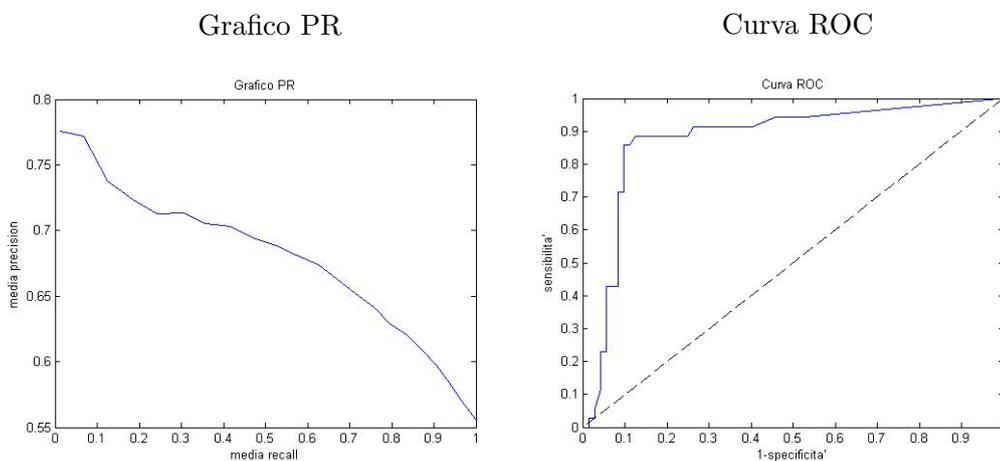


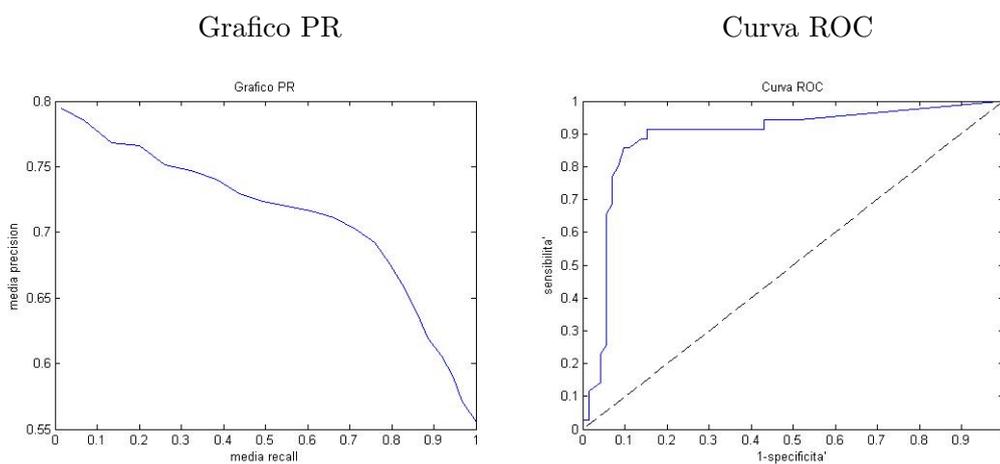
Figura B.2: Matrici 107 4x - mod.vettoriale - TrovaAlfa

- *Backslash e Levenberg-Marquardt*



## Modello matriciale

- *Gradiente proiettato*



- *TrovaAlfa*

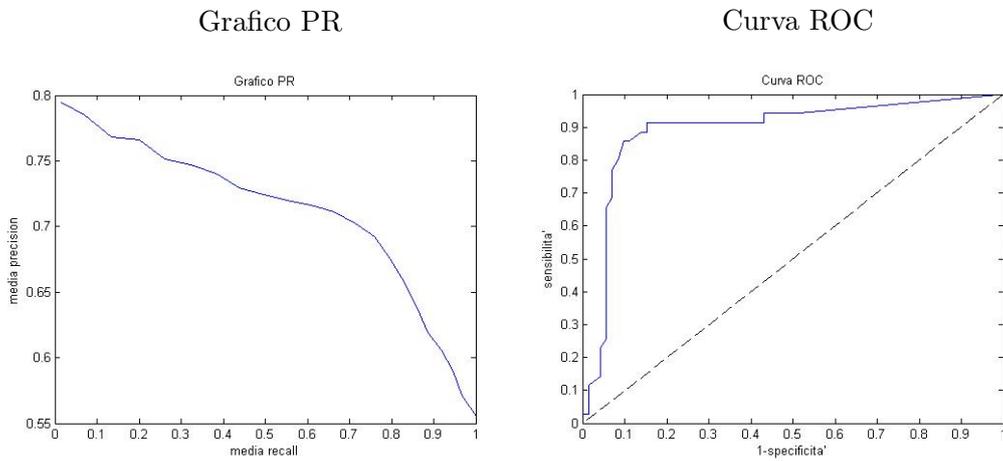


Figura B.5: Matrici 107 4x - mod.matriciale - TrovaAlfa

- *Backslash e Levenberg-Marquardt*

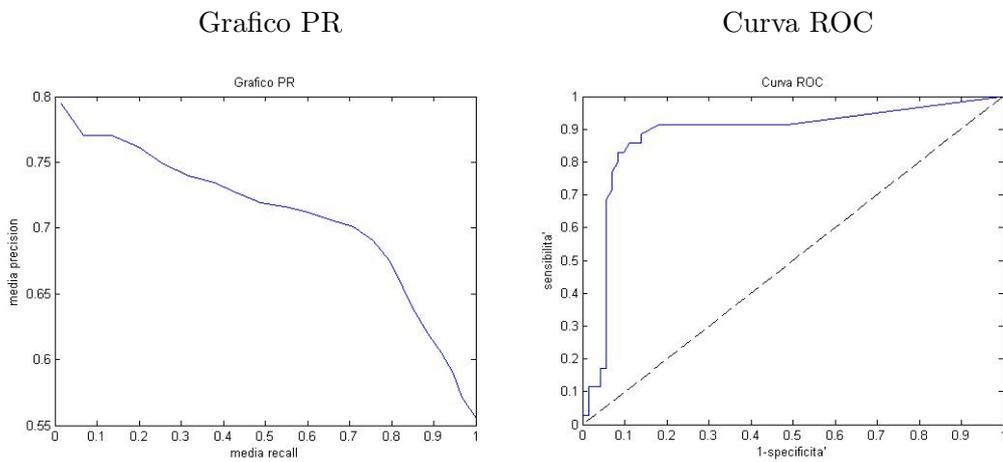


Figura B.6: Matrici 107 4x - mod.matriciale - non vincolato

## B.2 Matrici 107 8x - 5 funzioni misuranti

### Modello vettoriale

- *Gradiente proiettato*

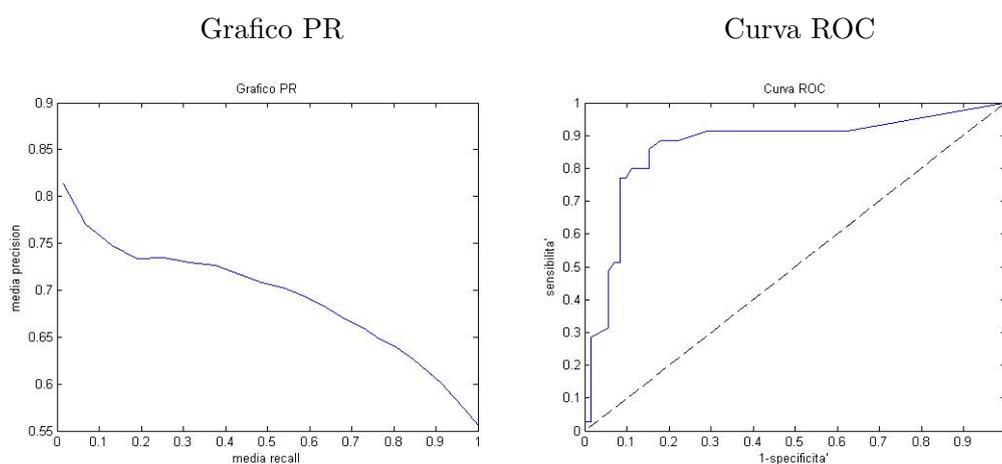


Figura B.7: Matrici 107 8x - mod.vettoriale - gradiente\_proiettato\_01

- *TrovaAlfa*

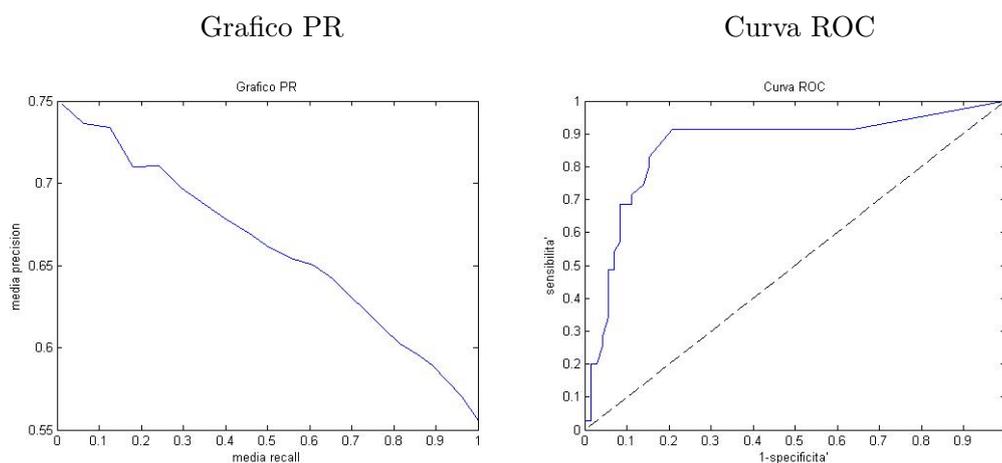


Figura B.8: Matrici 107 8x - mod.vettoriale - TrovaAlfa

- *Backslash e Levenberg-Marquardt*

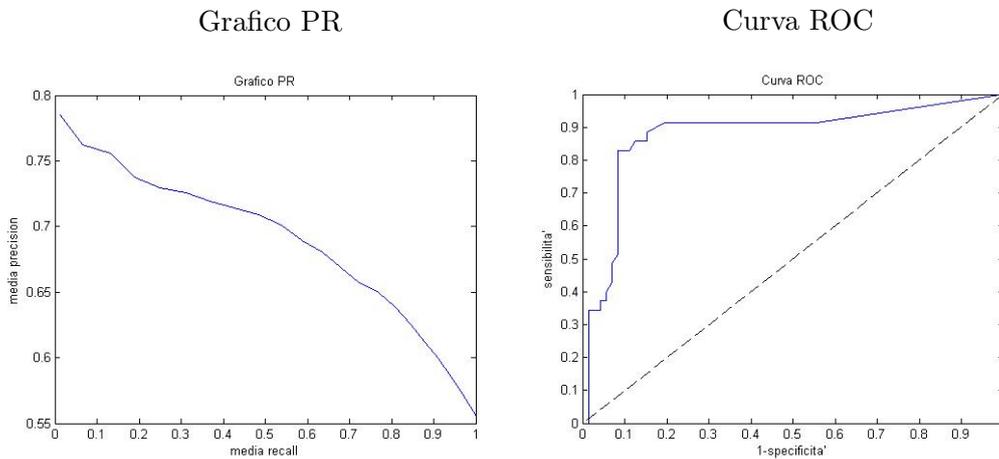


Figura B.9: Matrici 107 8x - mod.vettoriale - non vincolato

## Modello matriciale

- *Gradiente proiettato*

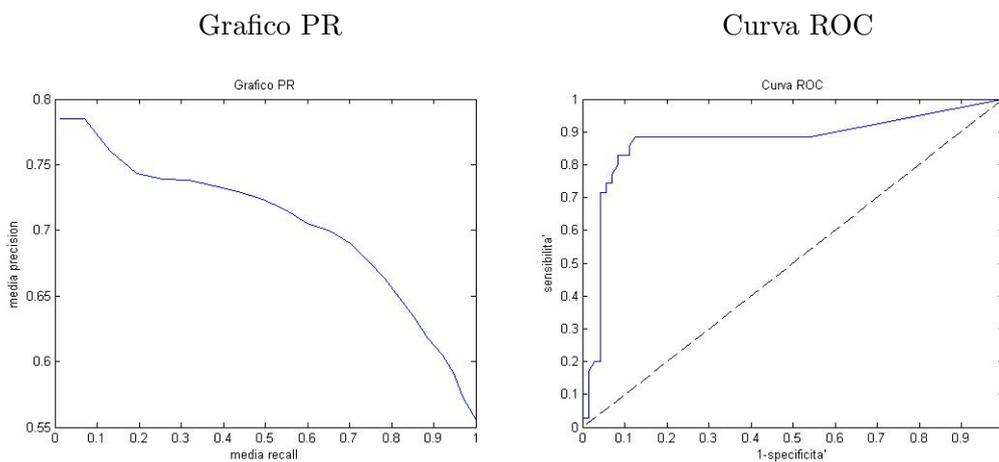


Figura B.10: Matrici 107 8x - mod.matriciale - gradiente\_proiettato\_01

- *TrovaAlfa*

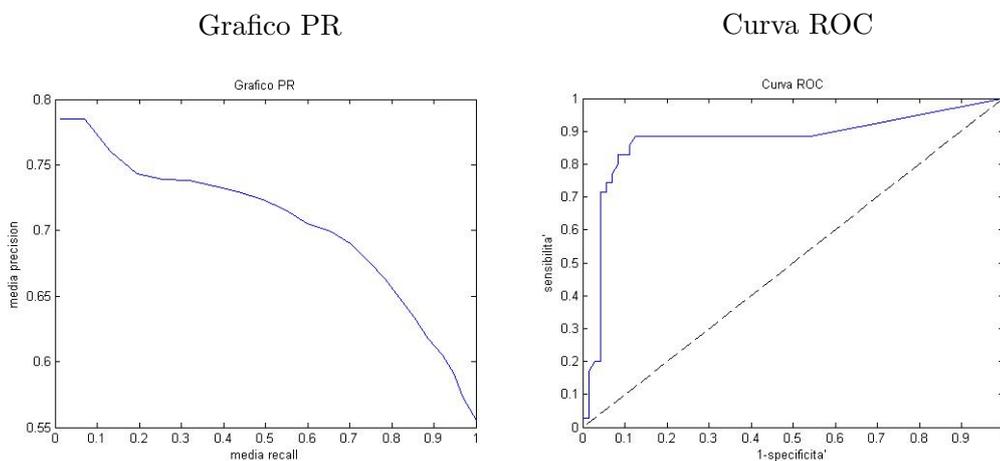


Figura B.11: Matrici 107 8x - mod.matriciale - TrovaAlfa

- *Backslash e Levenberg-Marquardt*

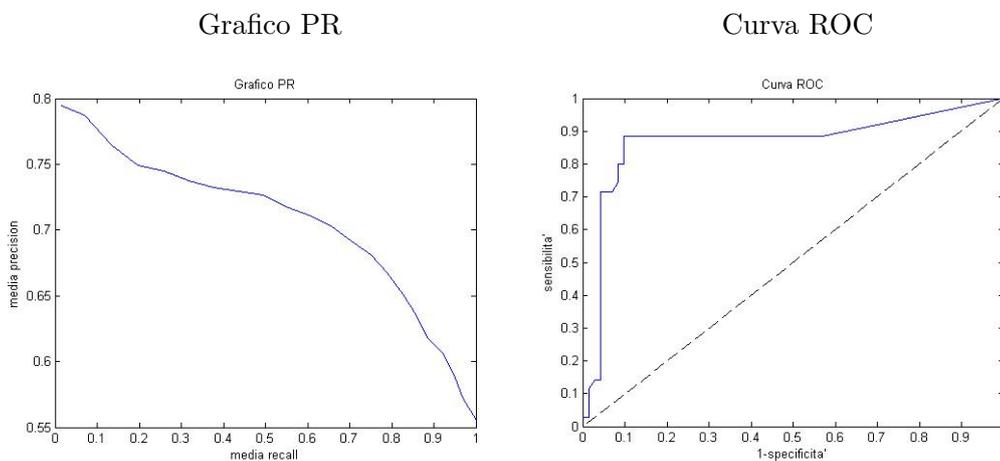


Figura B.12: Matrici 107 8x - mod.matriciale - non vincolato

## B.3 Matrici 107 - 9 descrittori

### Modello vettoriale

- *Gradiente proiettato*

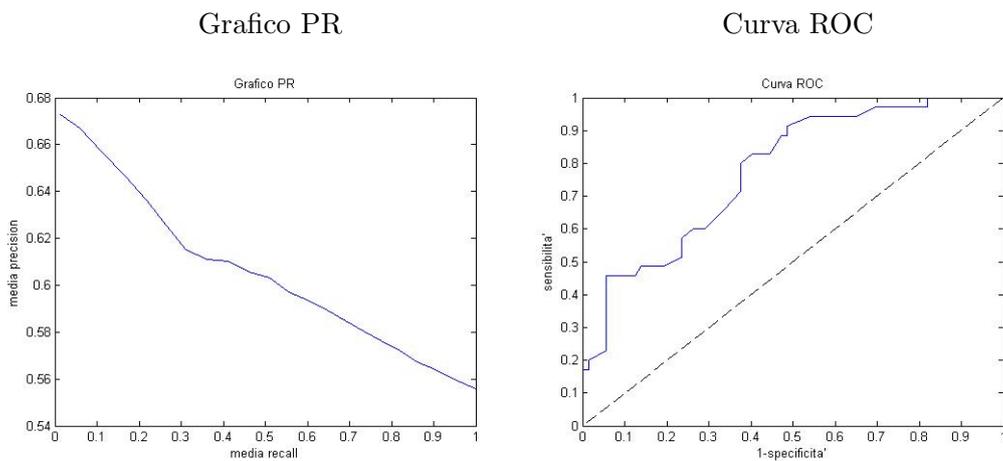


Figura B.13: Matrici 107 9desc - mod.vettoriale - gradiente\_proiettato\_01

- *TrovaAlfa*

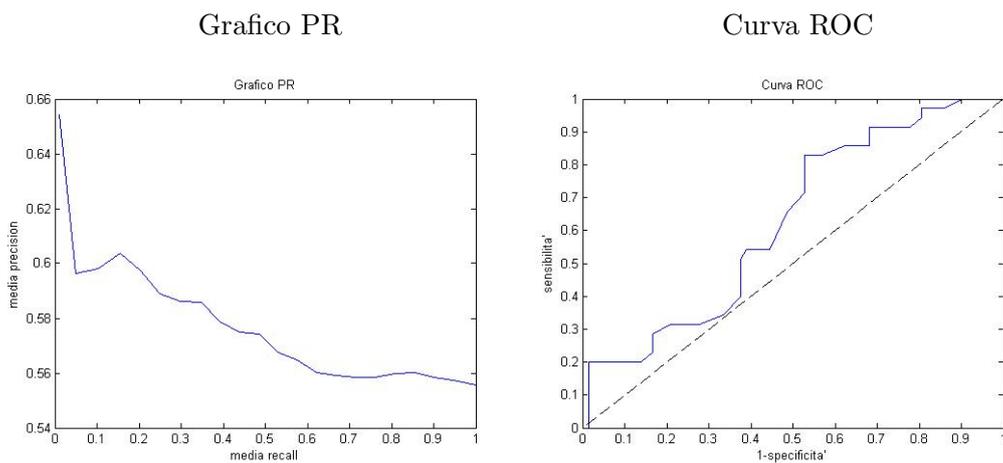


Figura B.14: Matrici 107 9desc - mod.vettoriale - TrovaAlfa

- *Backslash e Levenberg-Marquardt*

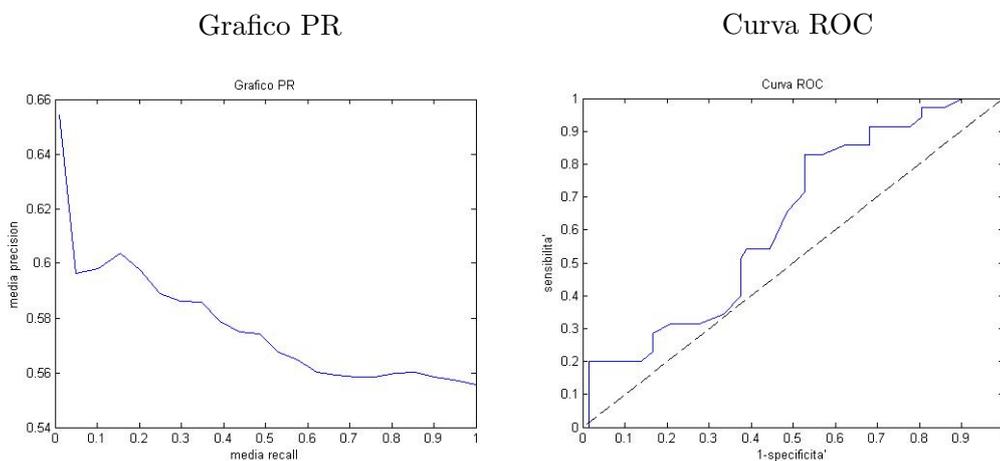


Figura B.15: Matrici 107 9desc - mod.vettoriale - non vincolato

## Modello matriciale

- *Gradiente proiettato*

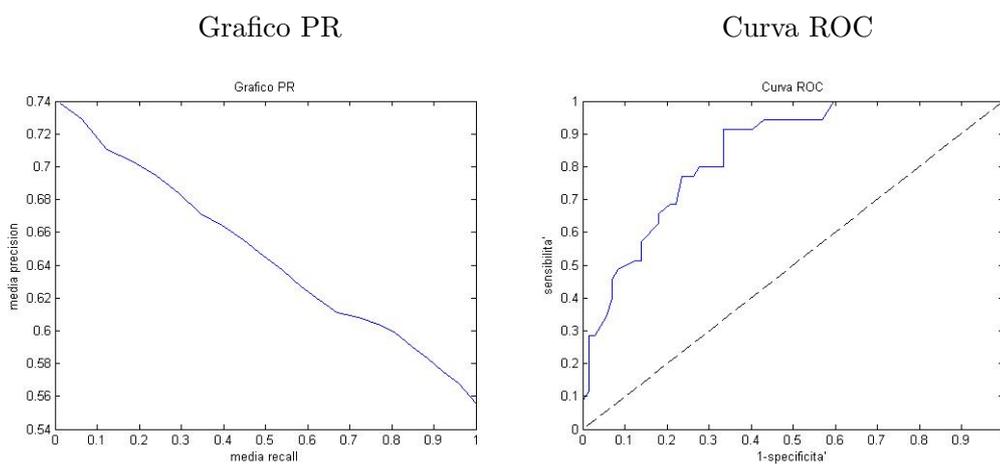


Figura B.16: Matrici 107 9desc - mod.matriciale - gradiente\_proiettato\_01

- *TrovaAlfa*

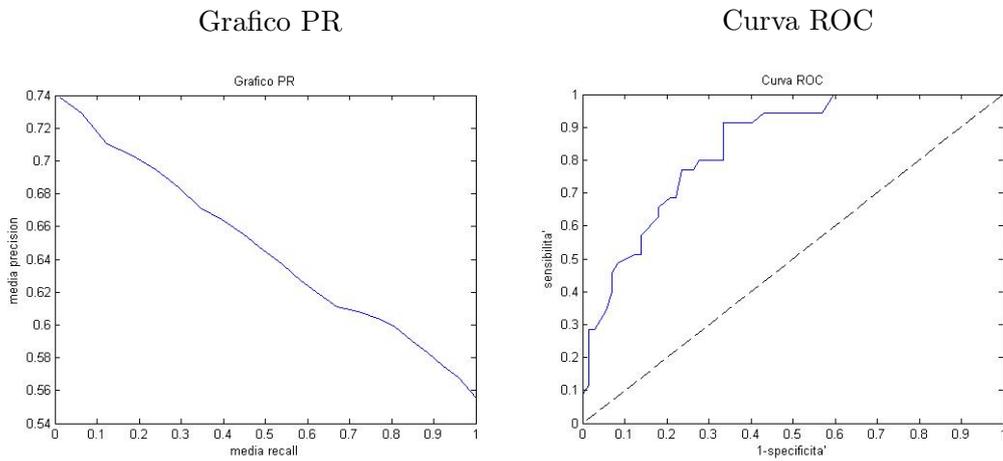


Figura B.17: Matrici 107 9desc - mod.matriciale - TrovaAlfa

- *Backslash e Levenberg-Marquardt*

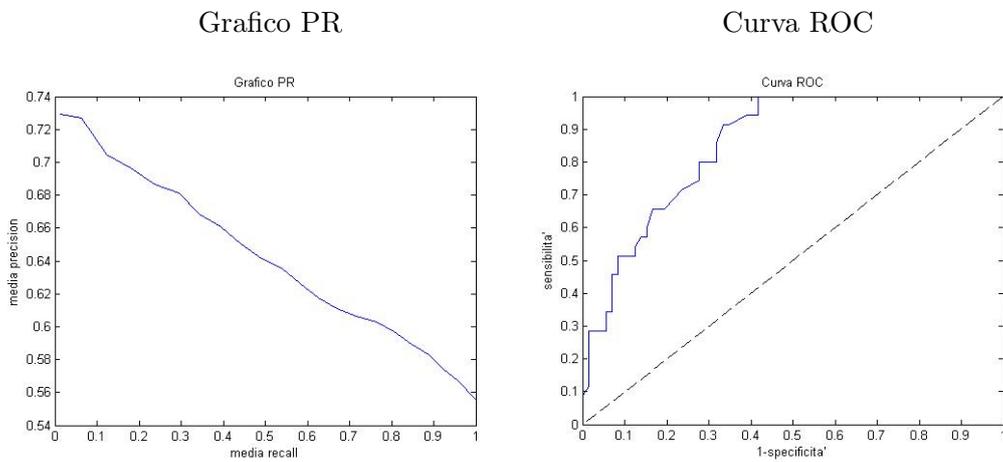


Figura B.18: Matrici 107 9desc - mod.matriciale - non vincolato

# Appendice C

## Codici MATLAB

A seguire riportiamo i codici da noi scritti in MATLAB delle funzione utilizzate e un esempio di `script` che utilizza l'operatore `\` di MATLAB (funzione `backslash`).

### `gradiente_proiettato_01.m`

```
function [x,iter,funzvec,itervec,gradvec]=  
gradiente_proiettato_01(funz,x0,itermax,tol1,mu,matrici,K,T)  
  
% Funzione che calcola il minimo della funzione funz con il  
% metodo del gradiente proiettato imponendo che gli  $\alpha_k$   
% abbiano valori compresi tra 0 e 1.  
% INPUT:  
% funz = funzione di cui si vuole calcolare il minimo  
% x0 = iterato iniziale  
% itermax = numero massimo di iterazioni  
% tol1 = tolleranza per il criterio d'arresto sulla distanza  
%       tra due iterati  
% mu = vettore o matrice mu  
% matrici = struttura contenente le K matrici distanza(NxN)  
% K = numero di matrici distanza
```

```
% T = vettore diagnosi (N nel caso matriciale)
% OUTPUT:
% x = soluzione calcolata
% iter = numero iterazioni effettuate
% funzvec = vettore valori funzione negli iterati calcolati
% gradvec = matrice valori gradiente negli iterati calcolati
% itervec = matrice contenente gli iterati calcolati

if nargin>=3
    funzvec = zeros(itermax,1);
    gradvec = zeros(K,itermax);
    itervec = zeros(K,itermax);
end
x = x0;
[z,w]=funz(x,mu,matrici,K,T);
iter=0;
continua = 1;
s=0.5;
while continua
    iter = iter+1;
    x_1=x-s*w;
    for i=1:K
        if x_1(i)> 0 && x_1(i)<1
            x_proj(i,1)=x_1(i);
        end
        if x_1(i)<=0
            x_proj(i,1)=0;
        elseif x_1(i)>=1
            x_proj(i,1)=1;
        end
    end
end
end
```

```
% Ricerca del passo con Armijo lungo le direzioni ammissibili
m=0;
condizione=1;
beta=0.3;
sigma=0.5;
while condizione
    beta_m=beta^m;
    arg=x+beta_m*(x_proj-x);
    [z1]=funz(arg,mu,matrici,K,T);
    sinistra=z-z1;
    destra=-sigma*beta_m*w'*(x_proj-x);
    if sinistra < destra
        condizione=1;
        m=m+1;
    else
        condizione=0;
    end
end
alpha=beta^m;
incremento=alpha*(x_proj-x);
x=x+incremento;
[z,w]=funz(x,mu,matrici,K,T);
diff= norm(incremento);
if nargout>=3
    funzvec(iter,1)=z;
    itervec(1:K,iter)=x;
    gradvec(1:K,iter)=w;
end
if diff<=tol1
    continua = 0;
    fprintf('Condizione di arresto sulla distanza tra due...
```

```

        ...iterati successivi \n');
elseif iter >= itermax
    continua = 0;
    fprintf('Massimo numero di iterazioni\n');
end
end
if nargout>=3
    funzvec= funzvec(1:iter,1);
    itervec= itervec(1:K,1:iter);
    gradvec= gradvec(1:K,1:iter);
end

```

## levenberg\_marquardt.m

```

function [x,iter,itervec,gradvec,funzvec]=
levenberg_marquardt(funz,x0,itermax,tol1,tol2,mu,matrici,K,T)

% Funzione che implementa il metodo di Levenberg-Marquardt
% (metodo di Gauss-Newton modificato) per la ricerca del minimo
% di una funzione funz del tipo  $F(x) = 1/2 ||r(x)||^2_2$ 
% INPUT:
% funz = funzione di cui si vuole trovare il minimo
% x0 = iterato iniziale
% itermax = numero massimo di iterazioni
% tol1 = tolleranza per il criterio di arresto sulla distanza
%        tra due iterati successivi
% tol2 = tolleranza per il criterio d'arresto sul residuo
%        relativo alla norma del gradiente della funzione
%        obiettivo
% matrici = struttura contenente le K matrici distanza (NxN)
% mu = vettore o matrice mu
% K = numero di matrici distanza presenti nella struttura

```

```
% T = vettore modello (caso vettoriale)
% T = N dimimensione matrici distanza (caso matriciale)

% OUTPUT:
% x = soluzione calcolata
% iter = numero iterazioni effettuate
% itervec = matrice contenente gli iterati calcolati
% gradvec = vettore valori norma gradiente negli iterati
%          calcolati
% funzvec = vettore valori funzione negli iterati calcolati

x = x0;
iter=0;
if nargout>=3
    itervec=zeros(K,itermax);
    gradvec=zeros(itermax,1);
    funzvec = zeros(itermax,1);
end
continua = 1;
%J matrice jacobiana di r
[F,r,J]=funz(x,matrici,mu,K,T);
JJ=J'*J;
dim=size(JJ);
I=eye(dim(1));
b=J'*r;
grad0=norm(b);
lambda1=grad0;
grad0=grad0*tol2;
gamma=1.e-4;
ro=0.5;
```

```
iter2max=15;
while continua
    iter = iter+1;
    lambda=min([lambda1; norm(b)]);
    d=(JJ+lambda*I)\b;
    % Armijo per la ricerca del passo
    alpha=1;
    x1=x-alpha*d;
    [F1]=funz(x1,matrici,mu,K,T);
    cond=F-gamma*alpha*b'*d;
    j=0;
    while (F1>cond & j<iter2max)
        j=j+1;
        alpha=ro*alpha;
        x1=x-alpha*d;
        [F1]=funz(x1,matrici,mu,K,T);
        cond=F-gamma*alpha*b'*d;
    end
    x = x-alpha*d;
    [F,r,J]=funz(x,matrici,mu,K,T);
    rr=norm(alpha*d);
    JJ=J'*J;
    b=J'*r;
    grad=norm(b);
    if nargout>=3
        itervec(:,iter)=x;
        gradvec(iter)=grad;
        funzvec(iter)=F;
    end
    if rr <= tol1 && grad <=grad0
        continua = 0;
```

```
        fprintf('Condizioni di arresto per tolleranze \n');
elseif iter> itermax
        continua = 0;
        fprintf('Massimo numero di iterazioni \n');
end
end
if nargout>=3
    itervec=itervec(:,1:iter);
    gradvec=gradvec(1:iter,1);
    funzvec=funzvec(1:iter,1);
end
```

## TrovaAlfa.m

```
function [x,Minimo] =
TrovaAlfa(alfa,Mat,Minimo,b,mu,struttura,K,N,T,proj)

% Funzione che risolve il nostro problema lineare nel caso
% vincolato
% INPUT:
%   alfa = vettore colonna qualunque di dimensione K
%   Mat = matrice del sistema KxK
%   Minimo = -1
%   b = vettore del sistema KxK
%   mu = vettore o matrice mu
%   struttura = struttura che contiene le K matrici distanza
%   K = dimensione del sistema
%   N = dimensione delle matrici struttura
%   T = vettore del modello (input non presenta nella funzione
%       utilizzata nel caso matriciale)
%   proj= vettore da dare in input come zeros(K,1)
% OUTPUT:
```

```
% x = soluzione calcolata
% Minimo = valore funzione nella soluzione calcolata

[K1,K2]=size(Mat);
K3=length(b);
K4=length(alfa);
if (K~=K1 || K ~=K2 || K ~=K3 || K~=K4)
    fprintf('Error: size dimension must agree\n');
end
%Fissa alfa_k a -1 se siamo alla prima chiamata
if (Minimo==-1)
    for k=1:K
        alfa(k)=-1;
        proj(k)=0;
    end
end
alfa0=Mat\b;
if (sum(alfa0<0)==0)
    if (Minimo==-1)
        x=alfa0;
    else
        if (compteg(alfa0,mu,struttura,K,N,T)<Minimo);
            x=alfa0;
        else
            x=alfa;
        end
    end
end
Minimo=compteg(x,mu,struttura,K,N,T);
%compteg: funzione che calcola il valore del minimo
% da dare in output
else
```

```
for k=1:K
    if (proj(k)==0)
        alfatilde=alfa0;
        alfatilde(k)=0;
        if (Minimo== -1)
            Minimo=computeeg(alfatilde,mu,struttura,K,N,T);
        else
            Minimo=min([Minimo,computeeg(alfatilde,mu,struttura,
                ...,K,N,T)]);
        end
        [alfa,Minimo]=TrovaAlfa(alfa,...
            [[Mat(1:k-1,1:k-1);zeros(1,k-1);Mat(k+1:K,1:k-1)],...
            [zeros(k-1,1);1;zeros(K-k,1)],...
            [Mat(1:k-1,k+1:K);zeros(1,K-k);Mat(k+1:K,k+1:K)],...
            Minimo,[b(1:k-1,1);0;b(k+1:K,1)],mu,struttura,K,N,T,...
            proj+[zeros(k-1,1);1;zeros(K-k,1)]);
    end
end
x=alfa;
Minimo=computeeg(x,mu,struttura,K,N,T);
end
end
```

## main\_backslash\_vettore.m

```
% Esempio di script in cui si utilizza l'operatore \ di MATLAB
% (da noi chiamato anche funzione backslash) per risolvere
% il problema lineare dei minimi quadrati nel caso non vincolato

% Dati relativi al database
N=107;
K=5;
```

```
nevi=72;
mel=35;
% Lettura matrici in input
struttura(1).matrice=xlsread('matrice_blu_107_4x.xls');
struttura(2).matrice=xlsread('matrice_rosso_107_4x.xls');
struttura(3).matrice=xlsread('matrice_verde_107_4x.xls');
struttura(4).matrice=xlsread('matrice_bordo_107_4x.xls');
struttura(5).matrice=xlsread('matrice_TV_107_4x.xls');
% Mette gli zeri sulla diagonale delle matrici distanza
for k=1:K
    for i=1:N+2
        struttura(k).matrice(i,i)=0;
    end
end
% Normalizza le matrici distanza tra 0 e 1
[matrici_norm] = normalizza_matrici(struttura,K,N);
% Crea la struttura in cui ci saranno le matrici pronte
% per l'analisi
for k=1:K
    matrici(k).analisi=matrici_norm(k).matrice(3:N+2,3:N+2);
end
% Memorizza la diagnosi delle immagini
diagnosi=struttura(1).matrice(3:N+2,2);
% Crea il vettote T e il vettore mu
for i=1:N
    if diagnosi(i)==1
        T(i,1)=-1/mel;
        mu(i,1)=N/(2*(N-1));
    else
        T(i,1)=1/nevi;
        mu(i,1)=-N/(2*(N-1));
    end
end
```

```
    end
end
% Costruisce la matrice del sistema
C=zeros(N,K);
for i=1:K
    C(:,i)=matrici(i).analisi*T;
end
% backslash
tic
x_calcolato=C\ mu;
tempo=toc;
disp('Soluzione calcolata')
disp(x_calcolato)

for k=1:K
    if x_calcolato(k)<0
        x_calcolato(k)=0;
    end
end
disp('Vettore utilizzato')
disp(x_calcolato)
```



# Bibliografia

- [1] Y. Rui, T.S. Huang, S. Chang: *Image Retrieval: Current Techniques, Promising Directions, and Open Issues*. Journal of Visual Communication and Image Representation **10** (1999) 39-62
- [2] J.P. Eakins, M.E. Graham: *Content-based Image Retrieval: A report to the JISC Technology Applications Programme*. Institute for Image Data Research, University of Northumbria at Newcastle, (1999). Disponibile su [http://www.jisc.ac.uk/uploaded\\_documents/jtap-039.doc](http://www.jisc.ac.uk/uploaded_documents/jtap-039.doc)
- [3] F. Long, H. Zhang, D.D. Feng: *Fundamental of Content-Based Image Retrieval*. Multimedia Information Retrieval and Management, Signals and Communication Technology, Springer (2003) 1-26
- [4] Dispense del Corso di *Information Retrieval* di G. Pasi, Università degli Studi di Milano Bicocca (2013)
- [5] H. Muller, W. Muller, D.McG. Squire, S. Marchand-Maillet, T. Pun: *Performance Evaluation in Content-Based Image Retrieval: Overview and Proposals*. Pattern Rec. Letters **22** (2001) 593-601
- [6] A. Cerri, M. Ferri, D. Giorgi: *A Complete Keypics Experiment with Size Functions*. Lecture Notes in Computer Science **3568** (2005) 357-366
- [7] <http://www.lamiapelle.salute.gov.it>
- [8] <http://www.airc.it/cancro/tumori/melanoma-cutaneo>
- [9] <http://www.lilt.it/getDoc.php?id=353>

- 
- [10] [www.videocap.it/public/%5Cpdf%5Cadam.pdf](http://www.videocap.it/public/%5Cpdf%5Cadam.pdf)
- [11] H. Edelsbrunner, J. Harer: *Persistent Homology - a Survey*. Contemporary Mathematics **453**, Amer. Math. Soc., Providence, RI (2008) 257-282
- [12] P. Frosini, C. Landi: *Size theory as a topological tool for computer vision*. Pattern Rec. and Image Analysis **9** (1999) 596-603
- [13] M. d'Amico, P. Frosini, C. Landi: *Using matching distance in size theory: a survey*. International Journal of Imaging Systems and Technology **16** (2006) 154-161
- [14] M. d'Amico, P. Frosini, C. Landi: *Natural pseudo-distance and optimal matching between reduced size functions*. Acta applicandae mathematicae **109**, Springer Netherlands (2010) 527-554
- [15] D.P. Bertsekas: *Nonlinear Programming* (2nd edition). Athena Scientific, Belmont, MA (1999)
- [16] Dispense del Corso di *Calcolo numerico* di E. Loli Piccolomini, Università di Bologna (2011)
- [17] Dispense del Corso di *Metodi numerici* di G. Landi, Università di Bologna (2013)
- [18] C.B. Moler: *Numerical Computing with MATLAB*. SIAM (2004)

# Ringraziamenti

Desidero ringraziare infinitamente il Prof. Massimo Ferri per la serietà con cui mi ha presa in considerazione, per la disponibilità, per le correzioni e gli spunti forniti e per l'entusiasmo che ha sempre dimostrato lungo tutto il percorso di questa tesi.

Ringrazio la Prof.ssa Elena Loli Piccolomini per la gentilezza e la disponibilità che mi ha sempre mostrato in questi anni, oltre che per l'aiuto fornitomi in questo progetto.

Ringrazio il dott. Ivan Tomba per il tempo pazientemente dedicatomi e per l'aiuto fondamentale nella realizzazione di questa tesi.

Vorrei poi ringraziare la mia famiglia per l'amore e il sostegno che mi ha dimostrato in tutti questi anni.

Ringrazio Marco per il passato, il presente e il futuro.

Un grazie a tutte le mie amiche, perché da sempre condividono con me gioie e dolori.

Infine, ringrazio tutti i miei compagni d'avventura che hanno reso indimenticabile questo percorso universitario.