Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE

Corso di Laurea Magistrale in Matematica

Recupero con "relevance feedback" di immagini dermatologiche mediante funzioni filtranti

Tesi di Laurea in Topologia Algebrica

Relatore:
Prof.
Massimo Ferri

Presentata da:
Eleonora Monti

Correlatore:
Dot.
Ivan Tomba

Seconda Sessione
Anno Accademico 2016/2017

Introduzione

L'incremento dei casi di melanoma cutaneo e la complessità della diagnosi portano spesso gli specialisti a ricorrere al supporto di macchinari per la diagnosi strumentale.
Chiaramente questi macchinari non si sostituiscono al ruolo del medico, ma fungono da supporto nella diagnosi di lesioni particolarmente complesse da classificare.
A tal proposito è stato realizzato un progetto per la costruzione di una macchina ad alta tecnologia per il supporto medico nell'individuazione di lesioni melanocitiche a cura del Dott. Ivan Tomba, dell' Ing. Andrea Visotti e della ditta CA-MI S.r.l., con la supervisione del Prof. Massimo Ferri, leader del gruppo di ricerca di Matematica della Visione del Centro ARCES dell'Università di Bologna, e la collaborazione del Prof. Ignazio Stanganelli dell'Università di Parma, responsabile del Centro di Oncologia Dermatologica dell'istituto IRST di Meldola.
La macchina contiene un database di immagini cliniche di nevi e melanomi e, una volta acquisita l' immagine di una nuova lesione, la confronta con tutte quelle del database e recupera le immagini più vicine a quella da esaminare.
Il problema affrontato in questa tesi è il seguente: spesso le immagini recuperate dalla macchina come le più "vicine"a quella in esame, vengono giudicate dai medici con valutazioni piuttosto basse di somiglianza; pertanto il lavoro svolto mira a migliorare ulteriormente la ricerca delle immagini dal database tenendo conto, in fase di recupero, dei pareri forniti dai medici (relevance feedback).
In particolare l'elaborato fornisce inizialmente alcune nozioni di base sulle lesioni della pelle, sulle loro caratteristiche e sulla diagnosi strumentale. Il secondo e il terzo capitolo contengono nozioni teoriche di omologia persistente e di relevance feedback utili a comprendere il funzionamento della macchina e il lavoro svolto su di essa. Infine l'ultimo capitolo spiega tutto il procedimento svolto e gli algoritmi di relevance feedback elaborati e utilizzati per cercare di migliorare il recupero delle immagini, illustrando i risultati ottenuti.

Indice

Capitolo 1

Melanoma cutaneo e diagnosi precoce

Una *neoplasia* o un *tumore* è una massa anomala di tessuto che cresce in eccesso e in modo incontrollato rispetto ai tessuti normali. Ogni tumore viene generalmente classificato, in base alla propria aggressività, in tumore benigno o maligno:

- Il *tumore benigno* è caratterizzato da cellule che mantengono, almeno in parte, le caratteristiche morfologiche del tessuto originario e restano confinate all'interno di un tessuto connettivo; così facendo comprimono i tessuti circostanti senza distruggerli direttamente
- Il *tumore maligno*, anche detto *cancro*, presenta cellule che tendono a diffondersi all'interno dei tessuti e degli organi vicini, propagandosi attraverso il sistema linfatico o i vasi sanguigni e generando spesso delle *metastasi*.
  Le metastasi tumorali derivano dalla crescita di cellule maligne distaccatesi dal tumore originario, ma sono situate in siti diversi.

Nonostante questa classificazione ci sono comunque situazioni intermedie che presentano caratteristiche dell'una e dell'altra classe.

1.1 Il melanoma cutaneo

Il *melanoma cutaneo* è un tumore maligno che ha origine nei melanociti, le cellule epiteliali responsabili della produzione di melanina, perciò si origina nella cute o, più raramente, negli occhi o nelle mucose[2].
I melanociti sono anche i responsabili, in condizioni normali, della presenza di macchioline scure sulla superficie della pelle conosciute comunemente come nei e designate in gergo clinico come *nevi*.
Il melanoma può manifestarsi su una pelle integra, oppure da nevi situati sulla cute che possono essere presenti sin dalla nascita o dalla prima infanzia (congeniti) oppure comparsi nel corso della vita (acquisiti).
Clinicamente si possono distinguere quattro tipologie di melanoma cutaneo[5]:

1. *melanoma lentigginoso acrale*: è il tipo di melanoma più raro nelle persone con carnagione chiara e si manifesta alle estremità degli arti, specialmente nelle sedi palmo-plantari e subungueali. Questo particolare melanoma ha una fase intraepidermica rapida e la comparsa di una regione nodulare rispecchia l'inizio di una crescita verticale del tumore.

2. *melanoma di tipo lentigo maligna*: è poco frequente (circa il 5-10% dei casi) e insorge soprattutto nelle persone anziane; tende a comparire in aree fotoesposte e danneggiate da un'esposizione solare cronica e pertanto si manifesta principalmente sul volto, dove assume l'aspetto di una macchia asimmetrica e piana che va dal bruno pallido al bruno nerasto. La sua evoluzione è piuttosto lenta e solo in fase avanzata possono comparire noduli sulla superficie.

3. *melanoma nodulare*: è una tipologia di melanoma caratterizzata da una crescita radiale scarsa o addirittura assente e da una crescita verticale sin dall'inizio; per questo motivo è molto aggressivo e spesso presenta metastasi alla diagnosi. Si riscontra nel 10-15% dei casi di melanoma e specialmente nei maschi intorno ai 50-60 anni; la sua diagnosi è difficile in quanto non dà sintomi e a volte non presenta la caratteristica colorazione tumorale.

4. *melanoma a diffusione superficiale*: è il melanoma più comune, circa il 60-70% dei casi, e spesso ha un decorso bifasico: dapprima ha una crescita orizzontale stabile e lenta che si manifesta come una lesione maculare; successivamente interviene una crescita verticale che corrisponde ad un'invasione in profondità e a un'evoluzione della lesione in placca con aree policromatiche o talvolta chiare.

1.1.1 I fattori di rischio

Il rischio di sviluppare un melanoma è legato sia a fattori ambientali che a fattori genetici propri dell'individuo.

Il principale *fattore esogeno* è la luce ultravioletta UV, e in particolare i raggi solari sotto forma di raggi UVA e UVB; perciò la troppa esposizione al sole e l'utilizzo eccessivo di lampade e lettini solari rappresenta un potenziale pericolo perché può danneggiare il DNA delle cellule della pelle e innescare la trasformazione tumorale[1].

- I *raggi UVA* sono circa il 95% delle radiazioni utlraviolette che arrivano sulla superficie terrestre, sono caratterizzati da un'energia bassa e penetrano nella cute in profondità; in tal modo provocano invecchiamento cutaneo, possono causare intolleranze solari e generano un'ampia serie di reazioni che danneggiano il DNA[5].
- I *raggi UVB* sono il restante 5% di raggi UV che raggiungono la superficie terrestre, hanno molto energia e, sebbene siano bloccati dalle nuvole, possono penetrare l'epidermide; sono i responsabili dell'abbronzatura, ma inducono anche infiammazione, apoptosi cellulare e immunosoppressione[5].

I *fattori endogeni*, ovvero fattori genetici, sono molteplici[5]:

- l'appartenenza al *fototipo 1-2*, cioè carnagione, capelli e occhi chiari, raddoppia il rischio di melanoma rispetto all'appartenenza al fototipo 4, ovvero pelle, occhi e capelli scuri.
- la *familiarità*
- la presenza di un *numero significativo di nevi* acquisiti o di nevi atipici
- la *recidività*.

1.1.2 I sintomi

Come detto precedentemente, i melanomi possono formarsi sia da nevi congeniti che da nevi acquisiti. Il primo passo per distinguere un nevo da un melanoma è sintetizzato nella regola dell'ABCDE[2]:

- ***A****simmetria nella forma*: un neo è generalmente circolare o tondeggiante, mentre un melanoma ha una forma più irregolare
- ***B****ordi irregolari*: un melanoma, a differenza di un neo benigno, presenta bordi frastagliati
- ***C****olore disomogeneo*: i nei sono generalmente caratterizzati da una colorazione uniforme contrariamente ai melanomi che invece presentano colorazioni diverse al proprio interno
- ***D****iametro superiore a 6 mm*: generalmente le dimensioni massime di un neo si aggirano intorno ai 6 mm di diametro, mentre un melanoma è caratterizzato da uno sviluppo, sia in larghezza che in spessore
- ***E****voluzione*: a differenza dei nei, i melanomi cambiano aspetto in un tempo piuttosto breve.

Altri campanelli d'allarme che devono essere valutati da un medico sono un neo che sanguina, che prude o che è circondato da un nodulo o da un'area arrossata[1].
Chiaramente la presenza di tutte queste caratteristiche agevola il riconoscimento e la diagnosi di un melanoma, tuttavia quando il melanoma è nelle fasi iniziali o quando solo alcuni caratteri clinici dell'ABCDE vengono riscontrati in un nevo la diagnosi può risultare complessa.
La difficoltà della valutazione clinica può provocare sia un alto tasso di inutili rimozioni di nevi, 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.

1.1.3 Diagnosi strumentale

In Italia i dati AIRTUM (Associazione italiana registri tumori) parlano di circa 13 casi di melanoma cutaneo ogni 100.000 persone con una stima che si aggira attorno a 3.150 nuovi casi ogni anno tra gli uomini e 2.850 tra le donne. Inoltre, l'incidenza è in continua crescita ed è addirittura raddoppiata negli ultimi 10 anni: il melanoma cutaneo è piuttosto raro nei bambini e colpisce soprattutto attorno ai 45-50 anni, anche se l'età media alla diagnosi si è abbassata negli ultimi decenni[1].
Considerati questi dati, la prevenzione e la diagnosi precoce diventano chiavi fondamentali per combattere il melanoma e a tale scopo è necessario controllare periodicamente l'aspetto dei propri nei, sia consultando il dermatologo, sia autonomamente.
Come abbiamo visto in precedenza, spesso l'osservazione a occhio nudo non permette di avere un' accurata sensibilità diagnostica e per questo motivo lo specialista può ricorrere all'utilizzo di strumentazioni apposite[5]:

dermatoscopio manuale: utilizzato in prima analisi dai dermatologi, è uno strumento portatile munito di lente d'ingrandimento fisso a 10x da appoggiare sulla cute da esaminare in modo da poter osservare in maniera più accurata le strutture anatomiche

tecnica dell' epiluminescenza: rende più accurato l'esame e consiste nell'applicazione di un mezzo di contrasto che permette di osservare la struttura microscopica dei primi strati di pelle sottostanti quelli visibili a occhio nudo

videodermatoscopio: strumento che utilizza la luce polarizzata, costituito da una fotocamera digitale con fibre ottiche e lenti che permettono di ottenere ingrandimenti fino a 1000x; 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.

Capitolo 2

Omologia persistente

Il confronto di immagini, problema centrale di questa tesi, rientra nel vasto campo della *visione artificiale*.
Ci sono diversi approcci a questo tipo di problema, ma quello che prendiamo in considerazione in questo caso è di tipo geometrico-topologico.
La geometria spesso è troppo rigida nell'affrontare i problemi relativi alla forma e al suo riconoscimento e la topologia lo è troppo poco, infatti spazi topologici omeomorfi possono essere molto diversi dal punto di vista intuitivo[7]. Per questo motivo sfruttiamo la **topologia persistente**, che cerca di aggirare queste difficoltà studiando non solo gli spazi topologici, ma funzioni continue definite su di essi che rappresentino l'idea di forma dette **funzioni filtranti**.
Gli strumenti principali utilizzati per descrivere e confrontare forme sono:

- *Pseudo-distanza naturale*
- *Diagrammi di persistenza*

ed entrambi ereditano dalla funzione filtrante eventuali sue invarianze per determinati tipi di trasformazioni.

2.1 Pseudo-distanza naturale

La topologia persistente si occupa di uno studio di tipo geometrico-topologico, quindi rappresenta le forme che sono oggetto del problema come spazi topologici.
Perciò consideriamo M spazio topologico compatto, che generalmente si può immergere in uno spazio euclideo $\mathbb{R}^k$, e introduciamo alcune definizioni:

Definizione 1 *Siano M uno spazio topologico compatto, non vuoto e localmente connesso e $\phi : M \longrightarrow \mathbb{R}$ una funzione continua. La coppia (M, ϕ) si dice* ***coppia di taglia*** *e ϕ è la* ***funzione filtrante*** *o funzione misurante.*

Osservazione 1 *Date due coppie di taglia (M, ϕ) e (N, ψ), diciamo che $(M, \phi) = (N, \psi)$ se $M = N$ e le due funzioni misuranti coincidono.*

Definizione 2 *Siano (M, ϕ) e (N, ψ) due coppie di taglia con M ed N omeomorfi e indichiamo con H(M,N) l' insieme degli omeomorfismi $\varphi : M \longrightarrow N$.*
Per ogni omeomorfismo $\varphi \in H(M, N)$ definiamo la funzione $\Theta : H(M, N) \longrightarrow \mathbb{R}$ come segue:

$$\Theta(\varphi) := \max_{P \in M} |\phi(P) - \psi(\varphi(P))|. \tag{2.1}$$

La funzione Θ si dice ***misura naturale di taglia*** *su H(M,N) relativa alle funzioni misuranti ϕ e ψ.*

Osserviamo che la misura naturale di taglia è uno strumento che quantifica il modo in cui φ modifica i valori delle due funzioni misuranti.
Indichiamo ora con $Size$ l'insieme delle coppie di taglia $\{(M, \phi)/(M, \phi)$ è coppia di taglia$\}$ e definiamo una pseudo-metrica su $Size$:

Definizione 3 *Sia $d : Size \times Size \longrightarrow \mathbb{R} \cup \{+\infty\}$ la seguente funzione:*

$$d((M, \phi), (N, \psi)) := \begin{cases} \inf_{\varphi \in H(M,N)} \Theta(\varphi) \ , \ seH(M, N) \neq \emptyset \\ +\infty \ , \ altrimenti \end{cases} \tag{2.2}$$

d si dice ***pseudo-distanza naturale*** *tra (M, ϕ) e (N, ψ).*

Osservazione 2 *d è effettivamente una pseudo-metrica: questo significa che due coppie di taglia (M, ϕ) e (N, ψ) tali che $d((M, \phi), (N, \psi)) = 0$ potrebbero essere diverse tra loro. Sono soddisfatte invece le altre due proprietà della metrica:*

- *simmetria: $d((M, \phi),(N, \psi))=d((N, \psi),(M, \phi))$*
- *disuguaglianza triangolare*

Osserviamo che, sebbene la pseudo-distanza naturale sia uno strumento molto valido per stabilire la somiglianza tra due spazi topologici, ha tuttavia lo svantaggio di essere piuttosto complessa da calcolare.
Infatti per calcolarla occorre tener conto di tutti gli omeomorfismi tra i due spazi topologici delle due coppie di taglia in questione.

2.2 Grafi di taglia e diagrammi di persistenza

Poichè, come abbiamo detto precedentemente, la pseudo-distanza naturale è piuttosto complessa da calcolare, si può approssimare l' insieme M con un sottoinsieme finito P di punti che in qualche modo conservi la struttura dello spazio M: la struttura matematica che meglio si presta a descrivere l'insieme P è quella di **grafo**.
I grafi sono strutture che possono avere un numero finito o infinito di vertici, ma quelli che sfrutteremo in questa tesi presentano un numero finito di vertici; pertanto ci occuperemo di definire questo tipo di grafi:

Definizione 4 *Un **grafo** G è dato dai seguenti elementi:*

- *un insieme finito i cui elementi si dicono vertici o nodi, V(G)=$\{v_1, ..., v_n\}$*
- *un particolare sottoinsieme del prodotto cartesiano V(G)× V(G) i cui elementi si dicono spigoli o archi, E(G)=$\{e_1, ..., e_m\} \subset$ V(G)× V(G).*
  Ogni spigolo e ∈ E(G) si indica con e=(v,w), dove v e w sono vertici del grafo G.

Definizione 5 *I grafi possono essere classificati in due classi:*

- *Un grafo G si dice **orientato** se E(G) è formato da coppie ordinate; quindi se (v,w) ∈ E(G) allora (w,v) ∉ E(G).*
  In questo caso di predilige la notazione "nodi" per gli elementi di V(G) e "archi" per quelli di E(G).
- *Un grafo G si dice **non orientato** se data la coppia (v,w) ∈ E(G) allora anche (w,v) ∈ E(G).*
  In tal caso si parla di "vertici" per gli elementi di V(G) e di "spigoli" per gli elementi di E(G).

In generale quando si parla di grafo si intende un grafo non orientato.

La connessione tra due vertici del grafo assume il ruolo che precedentemente aveva la connessione topologica: questo è il motivo per cui il grafo è la struttura matematica che meglio descrive l'insieme P con cui si approssima lo spazio tologico M.

Definizione 6 *Si dice **grafo di taglia** una coppia G=(G,ϕ) data da un grafo G e da una funzione definita sui vertici di G che assume valori reali, $\phi : V(G) \longrightarrow \mathbb{R}$.*

Partendo da un grafo di taglia (G, ϕ), si può ottenere una struttura molto utile per il confronto di immagini, il **diagramma di persistenza**[10], un descrittore geometrico che permette di immagazzinare una notevole quantità di informazioni.
Due immagini possono infatti essere comparate mettendo a confronto i rispettivi diagrammi di persistenza tramite la **bottleneck distance** o matching distance, che è la distanza che meglio approssima la pseudo-distanza naturale.
Per poter definire un diagramma di persistenza, occorre prima introdurre alcuni concetti.

Definizione 7 *Il* ***k-esimo numero di Betti*** $\beta_k(M)$ *è la dimensione del k-esimo spazio vettoriale di omologia* $H_k(M)$*.*
Intuitivamente $\beta_0(M)$ *conta le componenti connesse che compongono M,* $\beta_1(M)$ *misura il numero di buchi 1-dimensionali e* $\beta_2(M)$ *i fori 2-dimensionali.*

Notazione 1 *Siano* (M, ϕ) *una coppia di taglia e* $u \in \mathbb{R}$ *un numero reale.*
Si indica con $\boldsymbol{M_u}$ *l'insieme* $\{P \in M : \phi(P) \leq u\}$ *che si definisce* ***sottolivello di M rispetto a u****.*

Definizione 8 *Per ogni* $u, v \in \mathbb{R}$*,* $u < v$*, l' inclusione tra sottolivelli* $i : M_u \longrightarrow M_v$ *è una mappa continua e induce, per ogni grado k, una trasformazione lineare tra gli spazi di omologia* $i_* : H_k(M_u) \longrightarrow H_k(M_v)$*.*
La ***funzione dei numeri di Betti k-persistenti*** *(funzione k-PBN) associa alla coppia* (u, v) *il numero dim(Im* i_**), ovvero il numero di classi di k-cicli di* $H_k(M_u)$ *che sopravvivono in* $H_k(M_v)$*[8].*

Le funzioni dei numeri di Betti k-persistenti sono interamente determinate dalla posizione di alcuni punti e linee di discontinuità, chiamati *cornerpoints e cornerlines.*
Le coordinate (u, v) di un cornerpoint rappresentano i livelli di "nascita" e "morte" rispettivamente di un generatore e la *persistenza di un cornerpoint* è la differenza delle sue coordinate $v - u$. L' ascissa di una cornerline è il livello di nascita di un generatore che non muore mai.
Cornerpoints e cornerlines formano il **k-esimo diagramma di persistenza**.

Come anticipato precedentemente, due diagrammi di persistenza possono essere comparati mediante la distanza di Bottleneck, che mette in relazione i cornerpoints e le cornerlines dei due diagrammi.

Definizione 9 *Dati $D_{M,\phi}$ e $D_{N,\psi}$ diagrammi di k-persistenza delle due coppie di taglia (M,ϕ) e (N,ψ), si confrontino i cornerpoints di $D_{M,\phi}$ con quelli di $D_{N,\psi}$ o con la loro proiezione sulla diagonale $u=v$. La misura di questa corrispondenza è il sup delle L_∞-distanze di punti corrispondenti.*

*La **bottleneck distance** o **distanza di matching** dei diagrammi $D_{M,\phi}$ e $D_{N,\psi}$ è l'inf di queste misure calcolate tra tutti i possibili abbinamenti[8].*

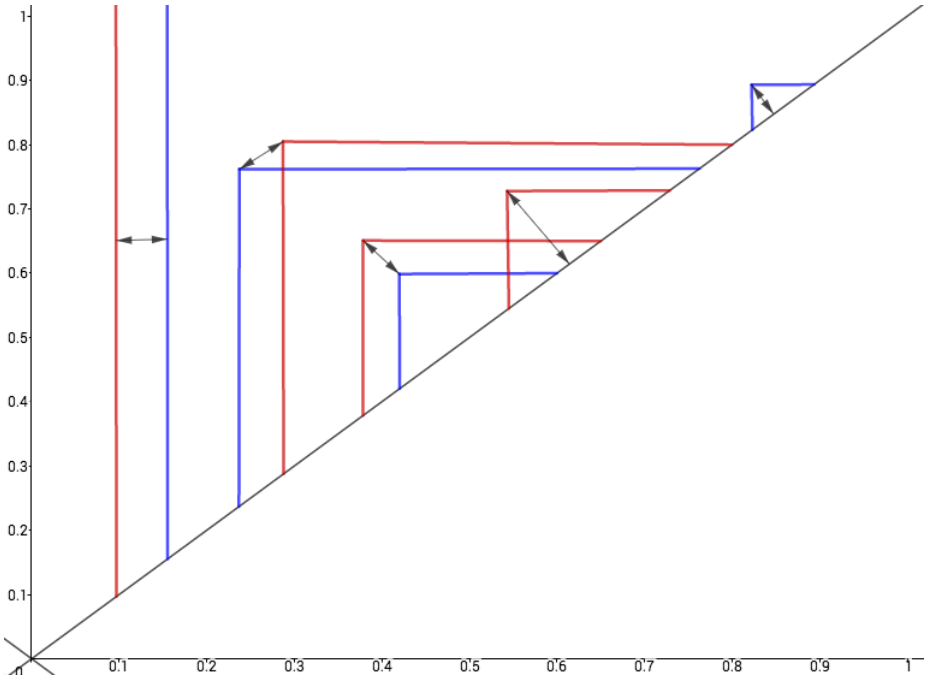

Figura 2.1: Bottleneck distance

Capitolo 3

Sistemi di recupero e Relevance Feedback

Le tecniche di relevance feedback mirano a ricoprire un ruolo importante nei motori di ricerca 3D in quanto aiutano a creare un collegamento tra l'utente (user) e il sistema.
Come vedremo successivamente, il lavoro di un motore di ricerca è strettamente correlato con il concetto di **rilevanza** (similarity), un processo cognitivo che dipende dall'osservatore, la cui soggettività è il principale problema dei sistemi di recupero.
Le tecniche di relevance feedback attualmente in uso sono diverse, ma quella che prendiamo in considerazione in questo elaborato si basa sull'idea che la rilevanza sia la mancanza di diversità tra due oggetti a confronto.

3.1 Information retrieval

L'information retrieval, ovvero il recupero di informazioni, è la disciplina informatica che si occupa della memorizzazione e del reperimento di documenti e dati; lo scopo di questa disciplina è la realizzazione di sistemi software che permettano la memorizzazione di grandi quantità di dati in un archivio dal quale possano poi essere reperiti facilmente i documenti più inerenti alle necessità dell'utente.
I due concetti base dell'information retrieval sono:

1. la **query**, ovvero l'insieme di parole chiave che rappresentano l'informazione richiesta e vengono digitate direttamente dall'utente all' interno del sistema di recupero

2. il **meccanismo del confronto**, cioè il meccanismo con cui il sistema confronta i documenti archiviati con la query.

Tuttavia il concetto cardine per il funzionamento di un sistema di recupero di informazioni è la definizione di **rilevanza**, che lega i due concetti precedenti stabilendo quali documenti siano rilevanti e quali non rilevanti per una data query.
Infatti un buon sistema di recupero ambisce a fare in modo che i documenti recuperati siano i più rilevanti per la richiesta ricevuta.
Come accennato precedentemente, il concetto di rilevanza è soggettivo e totalmente dipendente dall'osservatore, perciò il suo ruolo di centralità crea i principali problemi di un sistema di recupero: infatti la soggettività della rilevanza comporta l'incertezza sulla correttezza e sulla valutazione del risultato del recupero.

3.1.1 Motore di ricerca

Un motore di ricerca è un sistema automatico che si colloca nell'ambito dell'information retrieval: in particolare è un sistema che, data una determinata chiave di ricerca, analizza un insieme di dati restituendo un indice dei contenuti disponibili, classificati secondo il grado di rilevanza sulla base di formule matematiche.
Uno dei campi in cui i motori di ricerca trovano maggiore utilizzo è il web, dove il principale problema di un utente è trovare, in maniera efficace ed efficiente, le informazioni di cui ha bisogno all'interno di una quantità vastissima 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 presentano un'apposita maschera in cui l'utente può inserire la **query**, ovvero parole o frasi per definire l'oggetto cercato. Dopodichè viene effettuata la ricerca all'interno del database del motore, che è stato creato e catalogato seguendo particolari algoritmi tipici del motore stesso.
Infine, per rispondere alla richiesta dell'utente, elenca gli elementi recuperati dal database in ordine di **rilevanza** rispetto alla query. Quindi anche in questo caso il concetto di rilevanza è fondamentale per il meccanismo di ricerca.

3.2 Image retrieval

Un sistema di recupero di immagini, conosciuto anche come sistema di image retrieval, è un sistema informatico per la ricerca e il recupero di immagini digitali da un ampio database. Quindi si tratta di un sistema di information retrieval con la particolarità che i dati da recuperare sono immagini digitali.
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 studia il recupero di immagini a partire dal testo (text-based) e l'altra basata invece sullo studio dell'immagine vera e propria (content-based).

Il *recupero di immagini basato sul testo* è il primo approccio che si è tentato sin dal 1970 per l'image retrieval e consiste nel contrassegnare le immagini con delle parti testuali (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 le conoscenze già acquisite grazie all'information retrieval.

Il *recupero di immagini basato sul contenuto* è un approccio successivo, che risale ai primi anni '90, reso necessario della comparsa di grandi raccolte di immagini; in questo caso le immagini vengono indicizzate a partire dal loro contenuto visivo, ovvero da qualunque informazione che possa essere estrapolata dall'immagine stessa: colori, forma o struttura.

3.3 Relevance Feedback

In questi anni stiamo assistendo ad un aumento esponenziale del numero di modelli 3D, che oggi sono facilmente accessibili sia attraverso raccolte di dati in ambito generale, sia attraverso archivi di ricerca specifici.
Gli utenti sfruttano i contenuti 3D in ambiti consolidati, come la medicina e l'intrattenimento, così come in discipline emergenti quali ad esempio la bioinformatica.
Perciò si manifesta la necessità di sviluppare nuovi motori di ricerca e recupero in grado di fornire agli utenti dati 3D in modo rapido e preciso.
Considerate le grandi quantità di dati, il modello emergente è quello del recupero di dati basato sul contenuto, che permette di superare i problemi di ambiguità tipici dei sistemi di recupero basati sul testo.
Ogni sistema di recupero 3D è composto da tre fasi:

1. la formulazione della query da parte dell'utente
2. il confronto 3D tra la query e gli elementi del database
3. l'eventuale perfezionamento dei risultati.

Il concetto di forma può essere analizzato da ogni osservatore da diversi punti di vista, a seconda delle proprietà su cui desidera concentrarsi; è un concetto legato alla somiglianza, che a sua volta dipende dai punti di vista degli osservatori.
Per questo motivo occorre sviluppare intelligenti tecniche per adattare, selezionare e combinare i descrittori e la somiglianza che essi inducono, in modo che siano in accordo con le idee soggettive dell'osservatore.
Per ottenere questo risultato, la tecnica del **relevance feedback** richiede che l'utente abbia un ruolo attivo nel processo di ricerca: in particolare può migliorare il sistema di recupero attraverso la ripetizione dei tre passi precedentemente elencati, dando un feedback circa la pertinenza di alcuni elementi recuperati secondo le sue esigenze. Così il sistema raffina il suo insieme di risposte, in modo da adattarlo meglio al concetto di somiglianza dell'utente.

Per comprendere meglio il meccanismo, consideriamo un esempio tratto dall'articolo "3D relevance feedback via multilevel relevance judgements"di D. Giorgi, P. Frosini, M. Spagnuolo e B. Falcidieno, in cui si utilizza un dataset di 400 immagini divise in 20 categorie tra cui eseguire la ricerca:

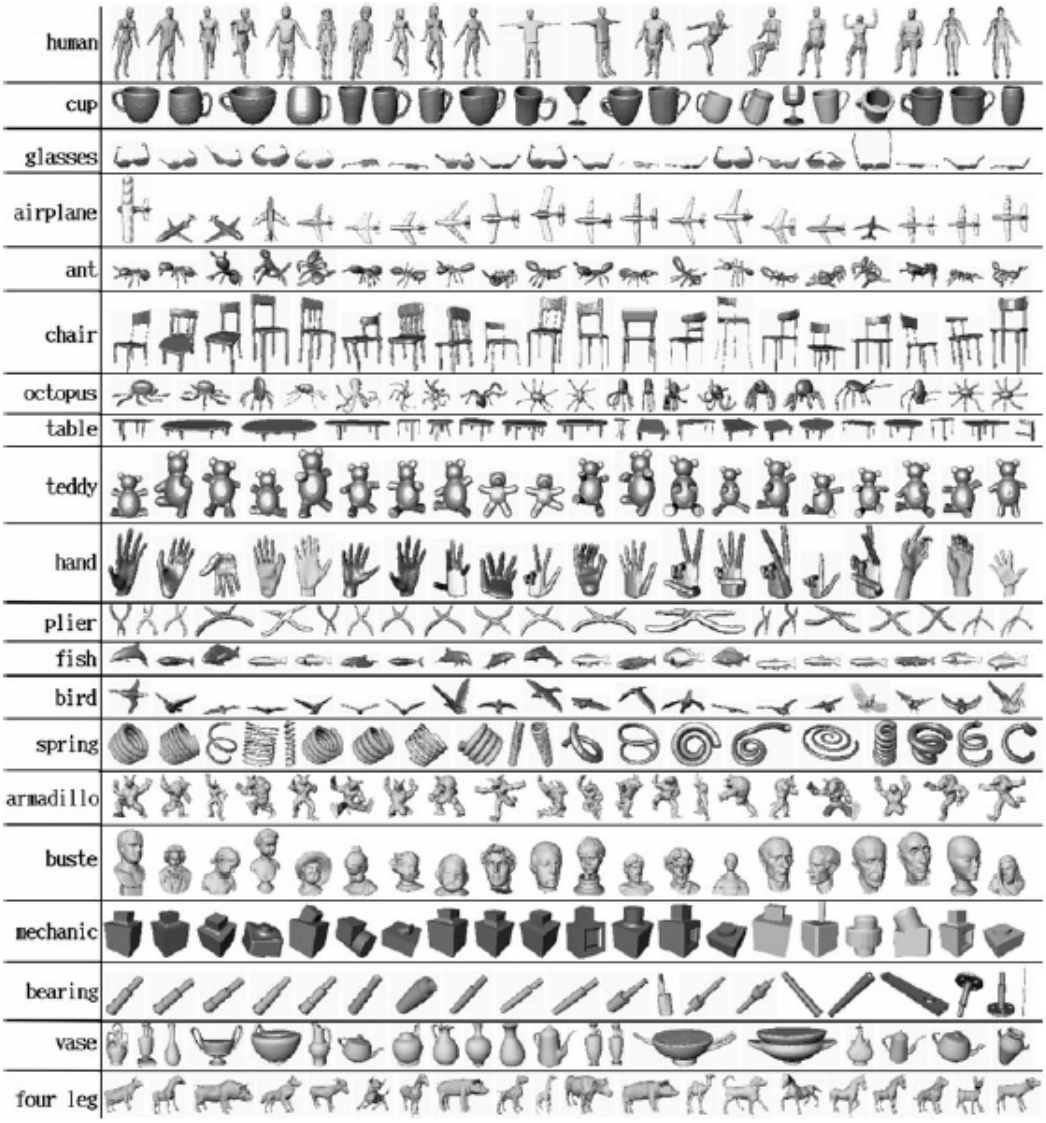

Figura 3.1: Dataset

Una volta fissato il dataset, occorre stabilire quali descrittori di forma 3D si vogliano utilizzare per effettuare poi la procedura del confronto: l'insieme dei descrittori di forma 3D include proprietà geometriche, topologiche, strutturali e di visualizzazione, in modo da catturare forme 3D (preferibilmente indipendenti) diverse.

A questo punto si sceglie la query e si effettua un primo **retrieval**, ovvero l'immagine scelta viene confrontata con tutte quelle del database e il sistema di recupero fornisce le 15 immagini più vicine alla query:

Figura 3.2: Retrieval senza feedback

In questo caso la query è un cavallo e si può osservare che la lista dei risultati contiene alcuni cavalli, ma anche alcuni falsi positivi. Per questo motivo si procede con il perfezionamento dei risultati, al quale partecipa attivamente l'utente fornendo il proprio feedback. In questo esempio l'utente giudica il cavallo nel cerchio blu "simile"(distanza 0.10) alla query, pertanto si effettua un ulteriore retrieval che tiene conto di questo feedback e si sbarazza di gran parte dei falsi positivi, inserendo nelle prime posizioni della nuova lista dei risultati il cavallo selezionato e altri animali quadrupedi.

Figura 3.3: Retrieval dopo il feedback

3.3.1 Idee alla base del relevance feedback

Una volta chiarito cosa si intenda con il termine "relevance feedback"e quale sia l'ambizione di questa tecnica, occorre definire quali siano le idee alla base.
Si tratta di un metodo per l'approssimazione interattiva di un pseudo-distanza δ su un set di dati

$$\Sigma = \{x_1, ..., x_N\} \tag{3.1}$$

basata sul feedback dell'utente: la pseudo-distanza δ, infatti, quantifica le diversità tra gli oggetti di Σ rispetto alla valutazione soggettiva dell'utilizzatore.

Il **primo passo** consiste nello stabilire dei descrittori di forma 3D, i quali generano una famiglia di pseudo-distanze tra gli oggetti del dataset Σ :

$$G = \{d_1, ..., d_n\} \tag{3.2}$$

In un **secondo momento** si definisce la pseudo-distanza iniziale tra gli oggetti nel database come il massimo tra le pseudo-distanze generate dai descrittori delle proprietà di forma:

$$D = \max_{d \in G} d \tag{3.3}$$

Una volta che viene individuata una query, la pseudo-distanza D viene utilizzata per effettuare un primo *retrieval*: gli oggetti del database vengono ordinati in ordine decrescente di distanza dalla query. Quindi il sistema restituisce un primo elenco di risposte.
Osserviamo che si decide di utilizzare l'operatore *max* (invece di una combinazione lineare ponderata tradizionale di pseudo-distanze) perché il massimo è correlato all'operatore "AND", più adatto al confronto soggettivo di forme complesse: in questo modo due oggetti sono simili se lo sono rispetto a tutte le proprietà della forma prese in considerazione.

Il **terzo passo** consiste nel richiedere all'utente di dare un *feedback* in merito alla rilevanza di alcune risposte fornite dal sistema.
A causa della complessità degli oggetti 3D e della varietà delle proprietà di forma, andiamo al di là della tradizionale classificazione binaria (rilevante / non rilevante) e chiediamo che l'utente esprima il proprio giudizio attraverso una scala numerica. L'utente, attraverso un'interfaccia dotata di una linea, può spostare il cursore lungo tale linea per esprimere la somiglianza tra due oggetti: la posizione del cursore viene poi trasformata in un valore numerico, che esprime la pseudo-distanza δ sulla coppia di oggetti del dataset Σ messi a confronto.

Siano q una query fissata e S un sottoinsieme del dataset Σ e supponiamo che i giudizi forniti dall'utente implichino la conoscenza della pseudo-distanza δ che rappresenta l'opinione dell'utente circa le diversità tra q e gli oggetti di S.
Questa conoscenza viene usata per inibire il ruolo delle pseudo-distanze d_i nella famiglia G che non sono compatibili con il giudizio dell'utente. Per fare questo l'idea è quella di riscalare d_i finchè non diventa compatibile con le informazioni fornite dall'utente: la nuova pseudo-distanza associa a ogni coppia di $\{q\} \times S$ un valore di diversità che non deve essere più grande del valore espresso dall'utente.
Matematicamente si vuole riscalare ogni pseudo-distanza d_i con una pseudo-distanza $\tilde{d}_i$

$$\tilde{d}_i = \lambda d_i \tag{3.4}$$

scegliendo λ come il più grande valore che soddisfa la disuguaglianza

$$\lambda d_i(q, x_j) \leq \delta(q, x_j), \quad x_j \in S \tag{3.5}$$

Per questo il **quarto step** è il calcolo dei pesi λ_i (con cui riscalare le pseudo-distanze d_i) che si può implementare attraverso una normalizzazione:

$$\lambda_i = min\{1; \frac{\delta(q, \bar{x})}{d_i(q, \bar{x})}\} \tag{3.6}$$

dove $\delta(q, \bar{x})$ è il giudizio numerico dell'utente circa la diversità tra la query e un oggetto $\bar{x}$ fornito dal primo retrieval.
Questa normalizzazione forza d_i a rispettare la disuguaglianza precedente e tale procedura corrisponde alla cancellazione delle pseudo-distanze che non sono compatibili con il giudizio dell'utente. Infatti, se d_i è molto più grande della valutazione di diversità fatta dall'utente, λ_i diventa molto piccolo e smorza la pseudo-distanza d_i in una nuova pseudo-distanza $\tilde{d}_i$ che ha un valore molto più basso.
Una volta calcolati i pesi λ_i si pesano le pseudo-distanze ottenendo una nuova famiglia di pseudo-distanze

$$\tilde{G} = \{\tilde{d}_1, ..., \tilde{d}_n\} \tag{3.7}$$

L'**ultimo passo** consiste nell'effettuare un nuovo retrieval utilizzando la pseudo-distanza

$$\tilde{D} = \max_{\tilde{d} \in \tilde{G}} \tilde{d} \tag{3.8}$$

come approssimazione della pseudo-distanza δ.

Capitolo 4

Lavoro svolto e risultati ottenuti

Come anticipato all'inizio di questo elaborato, è stato realizzato un progetto per la creazione di una macchina ad alta tecnologia per il supporto medico nell'individuazione di lesioni melanocitiche.
Occorre chiarire che questo progetto non mira a effettuare una diagnosi delle lesioni cutanee in esame, né tanto meno a sostituire il ruolo del medico, ma vuole far sì che il medico possa avvalersi di un supporto strumentale quando deve classificare una lesione di dubbia entità. Infatti l'idea è quella di fornire al medico le 10 immagini più "simili"alla lesione da classificare, recuperandole dal database della macchina (in cui sono già etichettate come melanomi piuttosto che nei) e restituendole in ordine di somiglianza.
Un problema riscontrato lavorando a questo progetto, che è il problema centrale affrontato in questa tesi, consiste nel fatto che spesso, eseguendo il retrieval rispetto a una determinata lesione (query), la macchina recupera 10 immagini che i medici giudicano poco rilevanti rispetto alla query in questione.
Per focalizzare meglio il problema analizziamo un esempio pratico considerando una lesione cutanea che sarà la nostra query e che, in questo caso, sappiamo essere una lesione maligna:

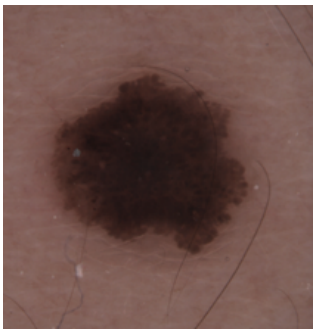

Figura 4.1: query, melanoma

A questo punto la macchina, attraverso meccanismi che descriveremo meglio successivamente, confronta la query con tutte le immagini del proprio dataset e fornisce all'utente (in questo caso il medico) le 10 immagini più simili; nel caso specifico di questo esempio osserviamo le prime 4 immagini restituite come output del retrieval:

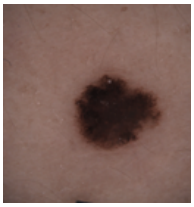

Figura 4.2: Prima immagine recuperata, melanoma

Figura 4.3: Seconda immagine recuperata, neo

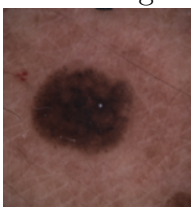

Figura 4.4: Terza immagine recuperata, neo

Figura 4.5: Quarta immagine recuperata, melanoma

Queste 4 immagini distano rispettivamente 0.077383, 0.080035, 0.086031 e 0.086287 dalla query e sono le immagini più vicine ad essa secondo i criteri di confronto della macchina. Tuttavia, quando è stato chiesto ai medici un giudizio di rilevanza su una scala da 0 a 3, essi hanno risposto con le seguenti valutazioni rispettive: 1, 2, 0 e 0.
Pertanto si osserva che i risultati del sistema non soddisfano il giudizio dei medici in quanto due lesioni hanno ottenuto voto 0 e nessuna delle prime 4 immagini recuperate ha ottenuto il

massimo dei voti; inoltre fra le lesioni successive (dalla quinta alla decima), alcune sono state valutate dai medici con voti superiori: in questo caso il sistema ha "ragionato"in modo diverso dai medici.
Affinché possa risultare affidabile e formativo, è importante che il retrieval automatico fornisca risultati in linea con quelli che otterrebbe un medico esperto; per questo motivo abbiamo deciso di lavorare su questo problema adottando la tecnica di relevance feedback precedentemente introdotta, con l'intento di far sì che il sistema venga "addestrato"secondo le indicazioni fornite da dermatologi esperti.

4.1 La macchina

Prima di poter descrivere il lavoro svolto e i risultati che ne sono conseguiti, occorre capire come effettivamente operi questa macchina e quali criteri di confronto utilizzi.
Il dermatoscopio IRSkin è un progetto a cura del Prof. Ivan Tomba, dell' Ing. Andrea Visotti e della ditta CA-MI S.r.l., con la supervisione del Prof. Massimo Ferri, leader del gruppo di ricerca di Matematica della Visione del Centro ARCES dell'Università di Bologna, e la collaborazione del Prof. Ignazio Stanganelli dell'Università di Parma, collaboratore dell'istituto IRST di Meldola; è un dermatoscopio costituito da:

- doppio monitor HD
- telecamera a 10 Mpx
- ingrandimenti fino a 100x
- Braccio meccanico
- Software high-tech
- Database di immagini diagnosticate

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.
Come prima cosa la telecamera rileva la lesione da analizzare e a questo punto inizia il lavoro vero e proprio sull'immagine acquisita, che viene prima di tutto privata dei peli (attraverso la

funzione Razor) in modo da eliminare alcune delle possibili fonti di rumore per i passi seguenti e successivamente sottoposta alla procedura di *segmentazione*. Il segmentatore determina il bordo della lesione, in modo da estrapolare la lesione vera e propria su cui andranno a lavorare i passi successivi: in particolare l'operazione di segmentazione crea una maschera, ossia una funzione a valori in $\{0, 1, 2\}$ che associa 0 ai punti esterni alla lesione, 1 a quelli interni e 2 a quelli del bordo.

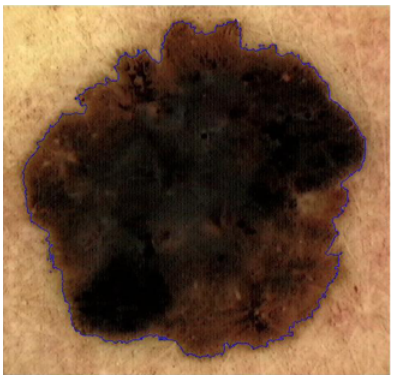

Figura 4.6: Lesione segmentata

Osserviamo che il bordo della lesione è una "curva di Jordan discreta"[12]: i pixel sul bordo hanno 2 e 2 soli pixel adiacenti secondo la 8-connessione; il bordo è 8-connesso, mentre l'interno e l'esterno sono 4-connessi.
A questo punto l'immagine segmentata deve essere confrontata con tutte quelle del database e per fare ciò occorre definire i descrittori di forma da utilizzare nel confronto.

Per prima cosa si estrapolano le features "semplici":

- istogramma dei colori: è un vettore H di 512 componenti tale che $H[i]$ =numero di pixel del colore i-esimo
- fattore di forma
- circolarità di Haralick
- simmetria rispetto all'asse maggiore di inerzia
- simmetria rispetto all'asse minore di inerzia
- simmetria rispetto al baricentro
- ellitticità
- eccentricità

- diametro
- entropia del colore

Queste features sono scalari e generano 8 distanze ottenute dal modulo della differenza tra il valore assunto dalla query rispetto a una determinata feauture e il valore assunto dall'immagine con cui si confronta rispetto alla stessa feature; queste 8 distanze sono: Col, FF, CH, Sym, Elt, Ecc, Diam, Entr.

Successivamente l'intera lesione segmentata viene triangolata e su di essa vengono scelte delle funzioni filtranti che descrivano le seguenti features:

- intensità luminosa
- blue, green, red
- excess-blue (ExcB=2B-G-R), excess-green (ExcG=2G-B-R), excess-red (ExcR=2R-G-B)[11]
- filtrazione opposta alla luminosità $(1 - L)$

Così facendo si ottengono 8 grafi di taglia globali a risoluzione circa 200x200, che vengono poi tramutati in diagrammi di persistenza per poter confrontare la query con le altre immagini del database. La query, così come le altre immagini, presenta un diagramma di persistenza per ogni feature e il confronto avviene mediante l'uso della bottleneck distance tra i diagrammi di ogni feature. In questo modo si ottengono 8 distanze: L, B, G, R, ExcB, ExcG, ExcR, iL.

Allo stesso modo, considerando il bordo della lesione, si ottengono 2 grafi di taglia a risoluzione 800x800 le cui funzioni filtranti descrivono:

- distanza dal baricentro
- 1 - distanza dal baricentro

Dal confronto tra i rispettivi diagrammi di persistenza si ottengono 2 distanze: ∂, $i\partial$.

Infine l'ultima distanza si ottiene considerando un grafo di taglia a risoluzione maggiore sulla zona periferica della lesione, prendendo come funzione filtrante la luminosità. In questo caso la distanza è $L\partial$.

In questo modo abbiamo ottenuto 19 distanze, nell'ordine:
L, B, G, R, ExcB, ExcG, ExcR, iL, $L\partial$, ∂, $i\partial$, Col, FF, CH, Sym, Elt, Ecc, Diam, Entr.

Considerando un database di N immagini e confrontando ognuna di esse con tutte le altre rispetto a ogni distanza sopra elencata, si ottengono 19 matrici distanza di dimensioni NxN in cui l'elemento di posto (i, j) rappresenta la distanza tra l'i-esima e la j-esima immagine relativamente a quella particolare funzione filtrante. Pertanto ogni matrice è simmetrica con la diagonale principale nulla.

$$D_k = \begin{pmatrix} 0 & d^k_{1,2} & d^k_{1,3} & \dots & d^k_{1,N} \\ d^k_{1,2} & 0 & d^k_{2,3} & \dots & d^k_{2,N} \\ d^k_{1,3} & d^k_{2,3} & 0 & \dots & d^k_{3,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d^k_{1,N} & d^k_{2,N} & d^k_{3,N} & \dots & 0 \end{pmatrix}, \quad k = 1, \dots, 19 \tag{4.1}$$

Queste 19 matrici vengono calcolate dalla macchina nel momento *off-line* di lavoro, fase in cui la macchina prende in input i file con i valori caratteristici delle N immagini che vanno a formare il database, crea i diagrammi di persistenza per ogni immagine e per ogni feature, calcola le distanze e restituisce in output le 19 matrici distanza sopra descritte.
Il momento *real time* consiste nell'inserimento di un'immagine di una nuova lesione nel sistema, che viene confrontata con tutte le immagini già presenti nel database e per la quale 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 fatto di aver già calcolato le matrici distanza nella fase offline risulta particolarmente utile anche ai fini del 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 in quanto si eliminano le più distanti.

4.2 Sperimentazione

L' obiettivo di questa tesi è cercare una soluzione al problema di incompatibilità, che si verifica in alcune situazioni, tra il concetto di somiglianza elaborato dalla macchina e quello di rilevanza espresso dal medico.
A tal proposito abbiamo implementato due tipologie di relevance feedback:

- il primo metodo di relevance feedback segue la tecnica descritta nel terzo capitolo di questo elaborato, e si basa sull'utilizzo dei massimi e dei minimi;
- la seconda tecnica che abbiamo voluto testare, e che descriveremo meglio in seguito, si basa sulle somme pesate.

Una volta affinata l'idea matematica alla base delle due tecniche, le abbiamo implementate realizzando un programma in linguaggio C che è stato poi utilizzato in diverse varianti per effettuare i vari test che vedremo successivamente.
I database a nostra disposizione, su cui abbiamo effettuate tutte le prove seguenti, sono due:

1. il primo dataset è costituito da 295 immagini di lesioni di cui è nota la diagnosi e suddivise in 67 melanomi e 228 nei;
2. il secondo, il database PH2, è una raccolta di 200 immagini dermatologiche, anch'esse già classificate, che si dividono in 40 lesioni melanocitiche e 160 nei[13].

Per ognuno dei due dataset abbiamo ottenuto le 19 matrici di distanze reciproche, calcolate dalla macchina nella fase off-line di lavoro. La famiglia di distanze $G = \{d_1, d_2, \ldots, d_{19}\}$ formata da queste 19 matrici è la base di partenza per tutto il nostro lavoro.

4.2.1 Relevance feedback con il metodo dei massimi

Per quanto riguarda questo approccio allo studio del nostro problema, abbiamo seguito l'idea di relevance feedback di D. Giorgi, P. Frosini, M. Spagnuolo e B. Falcidieno che abbiamo descritto passo a passo nel terzo capitolo di questa tesi.
In particolare, partendo dalla famiglia G, abbiamo calcolato la distanza D sulla base della quale effettuare un primo retrieval scegliendo

$$D = \max_{i=1,\ldots,19} d_i \tag{4.2}$$

poiché $d_1, \ldots, d_{19}$ sono matrici $N \times N$ (dove N è il numero di elementi del database in questione), per calcolare il massimo abbiamo confrontato le 19 matrici, elemento per elemento, e costruito

una nuova matrice dei massimi D, anch'essa di dimensioni $N \times N$, avente in ogni posizione l'elemento massimo tra i corrispondenti elementi di ogni matrice di G.
A questo punto abbiamo scelto 30 lesioni di ogni database (e pertanto le 30 colonne della matrice D corrispondenti) come query su cui fare i test di recupero immagini; in particolare, per ogni immagine, abbiamo

- **fatto un retrieval iniziale rispetto alla distanza** D: ciò equivale a scegliere la colonna di D corrispondente alla query in questione e riordinarla in ordine crescente di distanza in modo da avere nelle prime posizioni le immagini meno distanti dalla query
- **utilizzato la tecnica di relevance feedback**: quindi abbiamo calcolato i pesi, pesato le 19 matrici ottenendo una nuova famiglia di distanze e calcolato nuovamente il massimo su di essa per ottenere una nuova distanza $\tilde{D}$
- **effettuato un nuovo retrieval rispetto alla distanza** $\tilde{D}$
- **confrontato il risultati ottenuti con i due retrieval**

In questo caso abbiamo scelto i pesi come

$$\lambda_i = min\{1; \frac{\delta(q, \bar{x})}{d_i(q, \bar{x})}\}, \quad i = 1, \ldots, 19 \tag{4.3}$$

dove $\delta(q, \bar{x})$ è il giudizio numerico dell'utente circa la distanza tra la query e un oggetto $\bar{x}$ fornito dal primo retrieval.

Nel momento dell'implementazione, abbiamo preso i giudizi numerici di somiglianza forniti dai medici e considerato le distanze delta come $\delta_i = 1 - voto_i$; infatti un voto alto equivale a due immagini simili, perciò poco distanti.

Primo test

Un test preliminare che abbiamo fatto, abbastanza grossolano, ma che dà subito un'idea sul buon funzionamento o meno dell'idea alla base di questa tecnica, consiste nell'attribuzione di votazioni alte (0.9 o 0.99) alle immagine "recuperate correttamente"dopo il primo retrieval e di votazione basse (0.1 o 0.01) a quelle "recuperate non correttamente".
Con l'espressione "recuperate correttamente"intendiamo che, se la query in questione è un neo, attribuiamo voti alti a tutti i nei recuperati e bassi a tutti i melanomi; viceversa, nel caso in cui la query sia un melanoma, associamo voti alti ai melanomi e bassi ai nei.

Riportiamo di seguito i risultati ottenuti da questa prima prova, che abbiamo ottenuta utilizzando i voti 0.1 e 0.9 (con relative distanze 0.9 e 0.1) su entrambi i database:

	Data295			Data200	
	primo retrieval	secondo retrieval		primo retrieval	secondo retrieval
N315	9	9	IMD002	10	10
N353	9	9	IMD008	9	9
M365	2	7	IMD016	10	10
N367	7	7	IMD020	10	10
N464	10	10	IMD035	10	10
N466	4	6	IMD040	10	10
N557	8	10	IMD041	8	10
N584	7	8	IMD057	10	10
M600	1	3	IMD058	4	6
N757	8	10	IMD061	9	10
N779	6	8	IMD076	1	6
N785	9	9	IMD078	6	8
M1098	0	0	IMD085	7	8
M1178	2	5	IMD088	3	4
N1205	9	9	IMD101	10	10
N1644	8	9	IMD103	10	10
N1709	8	9	IMD160	8	9
N1787	8	9	IMD182	10	10
N1854	8	10	IMD197	9	9
M2010	6	8	IMD199	10	10
N2151	6	8	IMD210	7	8
M2522	2	3	IMD211	0	0
M2898	1	1	IMD219	7	10
N4342	10	10	IMD242	1	3
N4496	8	9	IMD367	9	10
N4568	10	10	IMD371	10	10
N4643	6	9	IMD374	10	10
N4740	9	10	IMD418	6	9
N4755	7	8	IMD421	8	9
N4916_1	10	10	IMD423	5	8

Per ogni database, la prima colonna indica i nomi delle 30 immagini per cui è stato fatto il retrieval, mentre la seconda e la terza colonna riportano il numero di immagini recuperate correttamente rispettivamente dopo il primo e dopo il secondo retrieval.
Lo stesso test è stato fatto in maniera del tutto analoga su entrambi i database i voti 0.01 e 0.99 che hanno prodotto le corrispondenti distanze 0.99 e 0.01.
Riportiamo i risultati ottenuti, con le sperimentazioni appena descritte, su entrambi i database:

- Data 295:

	Voti 0.1/0.9		Voti 0.01/0.99	
	primo ret.	secondo ret.	primo ret.	secondo ret.
tot.imm."corrette"	198	233	198	231
miglioramento	35		33	

- Data PH2:

	Voti 0.1/0.9		Voti 0.01/0.99	
	primo ret.	secondo ret.	primo ret.	secondo ret.
tot.imm."corrette"	227	256	227	253
miglioramento	29		26	

Si osserva immediatamente un ragionevole incremento del numero totale di immagini "recuperate correttamente""che ci fa intuire che questo metodo effettivamente lavora nella giusta direzione e presenta una certa "stabilità": infatti, non solo a livello globale c'è un miglioramento, ma anche osservando singolarmente ogni query non si ha alcun tipo di peggioramento locale. Inoltre osserviamo che, per quanto riguarda il secondo database, il miglioramento è leggermente inferiore, ma giustificato dal fatto che il recupero di immagini corrette dopo il primo retrieval è già di per sé migliore rispetto a quello del database con 295 immagini, pertanto occorrono meno correzioni.

Secondo test

Dopo la precedente prova preliminare che ci ha dimostrato una certa efficacia, abbiamo deciso di testare questo metodo più nello specifico e in particolare abbiamo sfruttato le valutazioni reali di somiglianza (tra la query e lesioni recuperate dal retrieval) forniteci dal Prof. Stanganelli. Tali valutazioni sono fornite su una scala da 0 a 3, in modo tale che il voto 0 corrisponda a immagini con somiglianza minima o nulla, e il voto 3 sia attribuito a immagini ritenute molto simili.
In particolare, per le nostre prove, abbiamo considerato questi voti riscalandoli in modo tale da portarli tra 0 e 1: per fare questo, però, abbiamo voluto tenere conto del tipo di valutazioni

globali del medico sulle 10 immagini recuperate dal retrieval per ogni query; nello specifico, non abbiamo trasformato il voto i, $i = 0, 1, 2, 3$ in i/3, ma abbiamo normalizzato i voti reali in maniera adattiva. La motivazione di questa scelta dipende dal fatto che, per alcune query, le dieci immagini recuperate abbiano avuto valutazioni molto basse, quali al esempio tutti 0 e un 1; in questo caso riteniamo che quell' 1 sia una valutazione relativamente positiva rispetto all'1 ottenuto da un retrieval con tutti voti maggiori.
A tal proposito abbiamo deciso di agire come segue:

- trasformare il voto 0/3 in 0.01 e il voto 3/3 in 0.999
- considerare la somma dei voti medici delle 10 immagini recuperate dal primo retrieval e trasformato i voti 1/3 e 2/3 nel modo seguente:

somma voti	1/3	2/3
$[0; 5[$	0.85	0.9245
$[5; 10[$	0.8	0.8995
$[10; 15[$	0.7	0.8495
$[15; 20[$	0.6	0.7995
$[20; 25[$	0.5	0.7495
$[25, 30]$	0.4	0.6995

  Quindi, per quanto detto precedentemente, abbiamo dato a 1 un valore più alto quando la somma dei voti medici è minore e abbiamo progressivamente diminuito tale valore all'aumentare della somma dei voti.
  Per quanto riguarda i valori corrispondenti alla valutazione 2/3, li abbiamo ottenuti dal valore di 1/3 in questo modo:

$$v_1 + \frac{0.999 - v_1}{2} \tag{4.4}$$

  dove v_1 è il valore attribuito a 1/3.

Una volta stabilito il modo in cui assegnare i giudizi fornitici dal Prof. Stanganelli, abbiamo calcolate le rispettive distanze $\delta_i = 1 - voto_i$ e fatto girare il programma precedente per effettuare i nuovi test. Così facendo abbiamo potuto appurare che questo approccio mediamente ci dà i buoni risultati sperati (ovvero fa risalire le immagini con le votazioni più alte e scendere quelle con giudizi più bassi), ma in alcune situazioni mostra dei limiti. Mostriamo i risultati ottenuti lavorando su due lesioni del dataset da 295 immagini che spiegano più nel concreto quanto appena detto: N1644 e N353.

- Nel caso dell'immagine N1644 abbiamo ottenuto i seguenti risultati:

Primo retrieval			Secondo retrieval		
distanze	immagini	voti	distanze	immagini	voti
0	N1644		0	N1644	
0.287372	N1106	0	0.1005	N1695	2
0.308544	N3667	0	0.1005	N642	2
0.310402	M1914	1	0.116559	M1914	1
0.318204	N4784	0	0.130983	N4733	0
0.321169	N642	2	0.133425	N4916_2	0
0.321692	N5072	0	0.134369	N786	0
0.323742	N3663	0	0.134572	N147	0
0.324674	M2177	0	0.140924	N574	0
0.327808	N1891	0	0.141335	N4800	1
0.328332	N1695	2	0.14171	N1123	0

In questo test sono state soddisfatte tutte le nostre migliori aspettative: le due immagini con voto 2 sono risalite dalla decima e quinta posizione rispettivamente in prima e seconda, mentre quelle con votazione 0 sono scese permettendo di far salire tra le prime 10 immagini N4800 (con voto 1).

- Studiando il problema con la scelta dell'immagine N353 come query abbiamo ottenuto:

Primo retrieval			Secondo retrieval		
distanze	immagini	voti	distanze	immagini	voti
0	N353		0	N353	
0.204502	M334	0	0.001	N1033	3
0.223929	N255	0	0.001	N1372	3
0.2494	N4755	1	0.001	N3667	3
0.255367	N991	0	0.001207	M334	0
0.25663	N3667	3	0.001334	N1144	0
0.270475	N1372	3	0.001377	N255	0
0.274174	N1144	0	0.001517	N425	0
0.292656	N1033	3	0.001586	M911	0
0.299698	N778	0	0.001588	M365	1
0.305681	N270	0	0.001609	N4755	1

In questo caso, come ci aspettavamo, le immagini con valutazione massima sono salite nelle prime tre posizioni; tuttavia, nonostante la somma dei voti sia rimasta aumentata da 10 a 11 dopo il secondo retrieval, l'immagine N4755 (con voto 1) è scesa dalla terza alla decima posizione lasciando salire immagini con voto 0 (come ad esempio N1144 che è salita di due posizioni).

Nonostante questa problematica, abbiamo comunque riscontrato, dopo aver testato l'algoritmo su tutte le 30 immagini precedenti del database con 295 lesioni, un aumento globale delle valutazioni mediche del retrieval post feedback rispetto al primo retrieval; in particolare dopo il primo retrieval la somma dei voti era 156, mentre dopo il feedback abbiamo ottenuto un aumento di 27 punti, raggiungendo un totale di 183.

Terzo test

Sebbene il test precedente abbia dato mediamente i risultati auspicati e abbia avuto un aumento globale delle valutazioni, abbiamo anche riscontrato in esso alcuni limiti (come mostrato nello studio dell'immagine N353). Per questo motivo, abbiamo ritenuto opportuno ragionare diversamente sull'assegnazione delle distanze δ_i sulla base dei voti; fino ad ora le abbiamo calcolate considerando $\delta_i = 1 - voto_i$, ma per evitare che dopo il feedback le immagini con valutazioni 1 o 2 scendessero al di sotto di immagini con valutazione 0 abbiamo pensato di calcolare tali distanze come segue:
scelta una query q, il retrieval restituisce le 10 immagini $x_i,\ i = 1, \dots, 10$ più vicine rispetto alla distanza globale D, ognuna con una valutazione v_i fornita dai medici

distanze	immagini	voti
d_1	x_1	v_1
d_2	x_2	v_2
d_3	x_3	v_3
d_4	x_4	v_4
d_5	x_5	v_5
d_6	x_6	v_6
d_7	x_7	v_7
d_8	x_8	v_8
d_9	x_9	v_9
d_{10}	x_{10}	v_{10}

con $d_1 < d_2 < d_3 < \cdots < d_{10}$ e $v_i = 0, 1, 2, 3$.

Abbiamo allora posto, $\forall i = 1, 2, \ldots, 10$:

- $\delta_i = 2d_{10}$, se $v_i = 0$;
- $\delta_i = d_1/2$, se $v_i = 1$;
- $\delta_i = d_1/4$, se $v_i = 2$;
- $\delta_i = d_1/8$, se $v_i = 3$;

in modo che ai voti più alti corrispondano le distanze minori e viceversa.

Abbiamo allora considerato le 30 lesioni precedenti del database con 295 immagini, utilizzato le loro valutazioni mediche per ricavarne le distanze δ_i con il metodo sopra descritto e ripetuto il test.
Riportiamo i risultati ottenuti studiando le due lesioni prese in esame anche nel test precedente: N1644 e N353.

- Con la scelta della lesione N1644 come query abbiamo ottenuto:

Primo retrieval			Secondo retrieval		
distanze	immagini	voti	distanze	immagini	voti
0	N1644		0	N1644	
0.287372	N1106	0	0.071843	N1695	2
0.308544	N3667	0	0.071843	N642	2
0.310402	M1914	1	0.083323	M1914	1
0.318204	N4784	0	0.093634	N4733	0
0.321169	N642	2	0.095379	N4916_2	0
0.321692	N5072	0	0.096054	N786	0
0.323742	N3663	0	0.0962	N147	0
0.324674	M2177	0	0.10074	N574	0
0.327808	N1891	0	0.101034	N4800	1
0.328332	N1695	2	0.101302	N1123	0

Poiché per questa lesione anche il test precedente dava i risultati auspicati, le immagini recuperate dopo il feedback con questa scelta delle distanze sono le stesse ottenute con il metodo precedente.
Pertanto le due immagini con valutazione di somiglianza maggiore salgono nelle prime posizioni, mentre scendono quelle con voto 0.

- Nel caso della lesione N353, invece, si ha:

Primo retrieval			Secondo retrieval		
distanze	immagini	voti	distanze	immagini	voti
0	N353		0	N353	
0.204502	M334	0	0.025563	N1033	3
0.223929	N255	0	0.025563	N1372	3
0.2494	N4755	1	0.025563	N3667	3
0.255367	N991	0	0.030863	M334	0
0.25663	N3667	3	0.034106	N1144	0
0.270475	N1372	3	0.035197	N255	0
0.274174	N1144	0	0.038778	N425	0
0.292656	N1033	3	0.040531	M911	0
0.299698	N778	0	0.040603	M365	1
0.305681	N270	0	0.041138	N4755	1

Anche in questo caso il retrieval dopo il feedback è del tutto analogo a quello ottenuto nella prova precedente scegliendo i voti da attribuire alle immagini in modo adattivo.

Questa cosa non deve far pensare a un fallimento della prova, tanto che a livello globale c'è un ulteriore miglioramento. Rimane tuttavia qualche problema, in particolare nei casi in cui abbiamo un primo retrieval con numerosi 1 e qualche 2 o 3: i voti alti finiscono nelle primissime posizioni, ma i voti bassi tendono a uscire dalle prime 10 facendo però spazio a immagini ancora peggiori. Un chiaro esempio di questo comportamento lo riscontriamo studiando N315:

Primo retrieval			Secondo retrieval		
distanze	immagini	voti	distanze	immagini	voti
0	N315		0	N315	
0.198573	N1242	2	0.042719	1077	1
0.222989	N1077	1	0.049643	N1242	2
0.227188	N4700	2	0.049643	N373	2
0.24611	N1144	1	0.049643	N4670	2
0.258078	N373	2	0.049643	N4700	2
0.275427	N4670	2	0.062499	N533	0
0.281354	M2686	0	0.062993	N1144	1
0.293472	N496	0	0.064485	N411	0
0.307808	N5072	0	0.065455	N4835	2
0314491	N705	0	0.06911	M334	0

In questo particolare caso l'immagine N1144 (già presente nelle 10 immagini recuperate dal primo retrieval e con voto 1) scende lasciando salire non solo le immagini con voto più alto, ma anche N533 che ha voto 0 e che viene "ripescata"tra le immagini che erano fuori dalle prime 10.

Con ogni probabilità questo sarà quasi impossibile da evitare, poichè a priori l'algoritmo non può sapere se spostare indietro le immagini con voto 1/3 faccia migliorare o peggiorare la situazione (avendo dato una valutazione solo alle prime 10 immagini recuperate).

4.2.2 Relevance feedback con il metodo delle somme pesate

Come accennato precedentemente, abbiamo deciso di provare un metodo diverso con cui approcciare la tecnica di relevance feedback; in particolare abbiamo voluto utilizzare, non più i massimi e i minimi, ma le somme pesate.
Più nello specifico, l'idea alla base di questa tecnica che proponiamo è la seguente:

1. avendo a disposizione la famiglia $G = \{d_1, d_2, \ldots, d_{19}\}$ delle 19 matrici distanza prodotte nel momento offline della macchina, otteniamo una distanza globale D (rispetto a cui effettuare il primo retrieval) in questo modo:

$$D = \frac{\sum_{i=1}^{19} d_i}{19} \tag{4.5}$$

2. richiediamo all'utente un feedback in merito alla rilevanza dei primi 10 output del retrieval fatto rispetto a D

3. calcoliamo i pesi utilizzando la distanza δ ottenuta dalla valutazione dell'utente:
supponiamo che $\delta = \delta(q, x)$ sia la distanza tra la query e l'output x e sfruttiamo le 19 matrici $d_1, \ldots, d_{19}$ per ottenere delle variabili ausiliarie $q_1, \ldots, q_{19}$ tali che

$$q_i = |\delta - d_i(q, x)|, \quad i = 1, \ldots, 19 \tag{4.6}$$

A questo punto imponiamo che i pesi abbiano somma 1 e in particolare risolviamo l'equazione

$$\frac{h}{q_1} + \frac{h}{q_2} + \frac{h}{q_3} + \cdots + \frac{h}{q_{16}} + \frac{h}{q_{17}} + \frac{h}{q_{18}} + \frac{h}{q_{19}} = 1 \tag{4.7}$$

dalla quale otteniamo h che ci permette di ricavare i pesi

$$\lambda_i = \frac{h}{q_i}, \quad i = 1, \ldots, 19 \tag{4.8}$$

4. pesiamo le matrici $d_1, d_2, \ldots, d_{19}$ con i pesi calcolati ottenendo una nuova famiglia di distanze $\tilde{G} = \{\tilde{d_1}, \tilde{d_2}, \ldots, \tilde{d_{19}}\}$ in cui $\tilde{d_i} = \lambda_i d_i$, i=1,2,...,19

5. infine otteniamo una nuova distanza globale $\tilde{D}$ come media pesata delle 19 matrici di partenza:

$$\tilde{D} = \frac{\sum_{i=1}^{19} \lambda_i d_i}{19} = \frac{\sum_{i=1}^{19} \tilde{d_i}}{19} \tag{4.9}$$

Abbiamo voluto testare questo metodo per verificare l'attendibilità dell'idea che ne è alla base, adattando l'algoritmo di volta in volta a seconda di quello che ci premeva verificare.
Inoltre possiamo osservare che il procedimento teorico sopra descritto riguarda l'assegnazione di un unico voto da parte dell'utente: a partire da quel voto si ricava la distanza δ che, rispettando le regole precedentemente elencate, permette di ottenere $\lambda_1, \ldots, \lambda_{19}$; tuttavia, il nostro intento è quello di tener conto delle valutazioni fornite dai medici per quanto riguarda la somiglianza tra la query e tutte le prime 10 immagini del retrieval, e pertanto ci troviamo a dover gestire 10 valutazioni e, per ognuna, le conseguenti 10 distanze $\delta_1, \ldots, \delta_{10}$. Per fare ciò abbiamo pensato a due approcci diversi che vedremo meglio all'interno delle varie prove effettuate.

Primo test

Parallelamente a quanto fatto per il relevance feedback basato sui massimi, anche in questo caso abbiamo effettuato un test preliminare che ci potesse dare un'idea sul buon funzionamento o meno di questa idea; per fare ciò abbiamo assegnato voto 0.9 (e successivamente 0.99) alle immagini "recuperate correttamente" e 0.1 (poi 0.01) a quelle "non corrette": in questo modo abbiamo ottenuto, per ogni query, 10 valori di distanze $\delta_1, \ldots, \delta_{10}$ calcolate a partire dai voti come $\delta_i = 1 - voto_i$.
In questo caso, per ogni distanza δ_i abbiamo effettuato le differenze $|\delta_i - d_1|, \ldots, |\delta_i - d_{19}|$ ottenendo $q_1^{(i)}, \ldots, q_{19}^{(i)}$; i pesi vengono poi ricavati prendendo

$$q_1 = \frac{q_1^{(1)} + q_1^{(2)} + \cdots + q_1^{(19)}}{10}$$
$$q_2 = \frac{q_2^{(1)} + q_2^{(2)} + \cdots + q_2^{(19)}}{10}$$
$$\vdots$$
$$q_{19} = \frac{q_{19}^{(1)} + q_{19}^{(2)} + \cdots + q_{19}^{(19)}}{10}$$

e procedendo al calcolo di h e conseguentemente di $\lambda_1, \ldots, \lambda_{19}$.

L'algoritmo è stato testato sulle stesse immagini utilizzate in precedenza in modo da poter fare un confronto tra i due metodi. In questo caso i risultati ottenuti attribuendo i voti 0.1 e 0.9 sono riportati nella seguente tabella dove, per ogni database, la prima colonna indica i nomi delle 30 immagini studiate mediante il retrieval, mentre la seconda e la terza colonna riportano il numero di immagini recuperate correttamente rispettivamente dopo il primo e dopo il secondo retrieval.

	Data295			Data200	
	primo retrieval	secondo retrieval		primo retrieval	secondo retrieval
N315	8	8	IMD002	10	10
N353	7	7	IMD008	10	10
M365	3	2	IMD016	10	10
N367	7	7	IMD020	8	8
N464	10	10	IMD035	10	10
N466	2	2	IMD040	10	10
N557	9	9	IMD041	10	10
N584	5	5	IMD057	7	8
M600	3	3	IMD058	5	6
N757	7	9	IMD061	9	9
N779	8	8	IMD076	6	6
N785	9	9	IMD078	7	7
M1098	1	2	IMD085	6	7
M1178	1	1	IMD088	5	5
N1205	7	7	IMD101	10	10
N1644	9	9	IMD103	10	10
N1709	9	10	IMD160	9	10
N1787	9	9	IMD182	9	10
N1854	10	10	IMD197	9	9
M2010	5	5	IMD199	10	10
N2151	7	7	IMD210	7	7
M2522	2	3	IMD211	0	0
M2898	4	4	IMD219	10	9
N4342	9	9	IMD242	1	1
N4496	6	6	IMD367	10	10
N4568	10	10	IMD371	9	9
N4643	8	9	IMD374	10	10
N4740	10	10	IMD418	9	9
N4755	9	9	IMD421	9	9
N4916_1	10	10	IMD423	9	9

Anche assegnando i voti 0.01 e 0.99 la situazione non varia particolarmente e si ha una tabella di risultati analoga alla precedente:

	Data295			Data200	
	primo retrieval	secondo retrieval		primo retrieval	secondo retrieval
N315	8	9	IMD002	10	10
N353	7	7	IMD008	10	10
M365	3	3	IMD016	10	10
N367	7	9	IMD020	8	10
N464	10	9	IMD035	10	10
N466	2	7	IMD040	10	10
N557	9	10	IMD041	10	10
N584	5	8	IMD057	7	10
M600	3	1	IMD058	5	6
N757	7	10	IMD061	9	4
N779	8	9	IMD076	6	7
N785	9	9	IMD078	7	9
M1098	1	2	IMD085	6	8
M1178	1	2	IMD088	5	3
N1205	7	9	IMD101	10	10
N1644	9	9	IMD103	10	10
N1709	9	10	IMD160	9	10
N1787	9	10	IMD182	9	10
N1854	10	10	IMD197	9	9
M2010	5	7	IMD199	10	9
N2151	7	10	IMD210	7	8
M2522	2	2	IMD211	0	1
M2898	4	5	IMD219	10	8
N4342	9	9	IMD242	1	2
N4496	6	8	IMD367	10	10
N4568	10	10	IMD371	9	10
N4643	8	10	IMD374	10	10
N4740	10	9	IMD418	9	7
N4755	9	10	IMD421	9	9
N4916_1	10	8	IMD423	9	4

Riassumiamo risultati ottenuti, con le sperimentazioni appena descritte, su entrambi i database:

- Data 295:

	Voti 0.1/0.9		Voti 0.01/0.99	
	primo ret.	secondo ret.	primo ret.	secondo ret.
tot.imm."corrette"	204	209	204	231
miglioramento	5		27	

- Data PH2:

	Voti 0.1/0.9		Voti 0.01/0.99	
	primo ret.	secondo ret.	primo ret.	secondo ret.
tot.imm."corrette"	244	248	244	244
miglioramento	4		0	

Come prima cosa, possiamo osservare che il primo retrieval, effettuato prendendo come distanza globale la media delle 19 distanze iniziali, è migliore rispetto a quello effettuato col metodo dei massimi: abbiamo 204 immagini "recuperate correttamente"(anziché 198) nel database con 295 elementi e 244 (invece di 227) nel dataset PH2.
Questo fattore ci ha fatto subito capire che dovevamo aspettarci, partendo da una situazione migliore, dei miglioramenti leggermente minori.
Effettivamente così è stato: a livello globale ci sono stati miglioramenti, seppur lievi, o comunque non ci sono stati peggioramenti; tuttavia lo stesso non si può dire a livello locale: se consideriamo ad esempio la query IMD061 del secondo database e il test con voti 0.01 e 0.99, dopo il primo retrieval la macchina recuperava 9 immagini con la stessa diagnosi della query, ma dopo il feedback questo numero è sceso a 4.
Per capire le motivazioni di questi locali peggioramenti abbiamo osservato i pesi $\lambda_1, \ldots, \lambda_{19}$ e abbiamo riscontrato in essi un notevole appiattimento:
λ_1=0.049830; λ_2=0.034508; λ_3=0.043265; λ_4=0.056508; λ_5=0.054642; λ_6=0.051923; λ_7=0.047263; λ_8=0.040084; λ_9=0.041300; λ_{10}=0.054443; λ_{11}=0.038570; λ_{12}=0.054743; λ_{13}=0.028490; λ_{14}=0.078283; λ_{15}=0.085753; λ_{16}=0.082496; λ_{17}=0.077353; λ_{18}=0.050053; λ_{19}=0.030492.
Riteniamo che questo appiattimento non permetta alle immagini recuperate dal retrieval "eccessivi movimenti", ma spostamenti verso l'alto o verso il basso di poche posizioni: questo fa sì che i miglioramenti siano limitati e che entrino nelle prime 10 immagini quelle che si trovavano subito sotto provocando, in alcune situazioni, dei peggioramenti.

Secondo test

Per evitare questo appiattimento dei pesi abbiamo provato diverse soluzioni:

a) abbiamo calcolato i q_i come descritto precedentemente, calcolato i quadrati dei q_i e successivamente h risolvendo l'equazione $\frac{h}{q_1^2} + \frac{h}{q_2^2} + \cdots + \frac{h}{q_{19}^2} = 1$; infine abbiamo ottenuto i pesi come $\lambda_i = \frac{h}{q_i^2}$

b) abbiamo calcolati i q_i, non più come media dei $q_i^{(k)}$, $\quad k = 1, \ldots, 19$, ma come media pesata scegliendo i pesi come $\frac{\delta_i}{numero\ di\ output}$

c) abbiamo unito le due soluzioni precedenti

La soluzione a) è quella che ci ha dato i risultati migliori che riportiamo di seguito; anche in questo caso abbiamo attribuito valutazioni 0.1 e 0.9 (o 0.01 e 0.99) ripetendo il test precedente e sfruttando i quadrati dei q_i.
I valori ottenuti con 0.1 e 0.9 sono:

	Data295			Data200	
	primo retrieval	secondo retrieval		primo retrieval	secondo retrieval
N315	8	8	IMD002	10	10
N353	7	8	IMD008	10	10
M365	3	2	IMD016	10	10
N367	7	7	IMD020	8	8
N464	10	9	IMD035	10	10
N466	2	3	IMD040	10	10
N557	9	9	IMD041	10	10
N584	5	5	IMD057	7	8
M600	3	3	IMD058	5	6
N757	7	9	IMD061	9	9
N779	8	8	IMD076	6	5
N785	9	9	IMD078	7	7
M1098	1	2	IMD085	6	7
M1178	1	1	IMD088	5	5
N1205	7	8	IMD101	10	10
N1644	9	9	IMD103	10	10

N1709	9	10	IMD160	9	10
N1787	9	8	IMD182	9	10
N1854	10	10	IMD197	9	9
M2010	5	5	IMD199	10	10
N2151	7	8	IMD210	7	8
M2522	2	3	IMD211	0	0
M2898	4	4	IMD219	10	8
N4342	9	10	IMD242	1	1
N4496	6	6	IMD367	10	10
N4568	10	10	IMD371	9	9
N4643	8	9	IMD374	10	10
N4740	10	9	IMD418	9	9
N4755	9	9	IMD421	9	8
N4916_1	10	10	IMD423	9	9

In maniera analoga abbiamo ripetuto il test attribuendo le valutazioni 0.01 e 0.99. Globalmente la situazione può essere riassunta come segue:

- Data 295:

	Voti 0.1/0.9		Voti 0.01/0.99	
	primo ret.	secondo ret.	primo ret.	secondo ret.
tot.imm."corrette"	204	211	204	213
miglioramento	7		9	

- Data PH2:

	Voti 0.1/0.9		Voti 0.01/0.99	
	primo ret.	secondo ret.	primo ret.	secondo ret.
tot.imm."corrette"	244	246	244	248
miglioramento	2		4	

Quello che si osserva rispetto al test precedente, è che a livello globale permane un miglioramento (in alcuni casi minore di prima, in altri maggiore) e a livello locale la situazione migliora. Tuttavia rimangono ancora casi isolati in cui si riscontra un peggioramento del retrieval dopo il feedback rispetto al primo; questo evidenzia una maggiore instabilità di questo metodo rispetto al metodo di relevance feedback che utilizza il massimo: in questo caso qualunque sistema

adottiamo per il calcolo dei pesi produce qualche peggioramento sporadico, cosa che invece non succedeva con l'altra tecnica di relevance feedback.

Un'altra osservazione che è doveroso fare è la seguente: sebbene con questo metodo il primo retrieval, globalmente parlando, sia migliore (e quindi ci si aspettava un miglioramento numericamente inferiore), una volta effettuato il feedback il primo metodo recupera correttamente più immagini rispetto a quest'ultimo.

Terzo test

Avendo appurato che l'approccio precedente, ovvero l'utilizzo dei quadrati dei q_i, è quello che con questa tecnica di relevance feedback ci ha dato i risultati migliori, abbiamo deciso di testare proprio quest'ultimo utilizzando i voti reali fornitici dai medici.
In particolare, analogamente a quanto fatto nell'altra tecnica di relevance feedback, abbiamo voluto prendere i 10 voti con cui i medici hanno valutato le prime 10 immagini del retrieval, e per ogni voto v_i, $\quad i = 1, \ldots, 10$, abbiamo ottenuto un valore di distanza δ_i come segue:

- $\delta_i = 2d_{10}$, se $v_i = 0$;
- $\delta_i = d_1/2$, se $v_i = 1$;
- $\delta_i = d_1/4$, se $v_i = 2$;
- $\delta_i = d_1/8$, se $v_i = 3$;

così facendo i valori δ_i vengono distribuiti tra $\frac{d_1}{8}$ (dove d_1 è la distanza minore) e $2d_{10}$ (dove d_10 è la distanza maggiore) in modo che al voto 3 corrisponda una distanza molto piccola, al voto 0 corrisponda la distanza maggiore e ai voti 1 e 2 corrispondano i valori intermedi.
A partire da questi valori δ_i sono stati calcolati i q_i, i loro quadrati e successivamente i pesi $\lambda_1, \ldots, \lambda_{19}$ con $\lambda_i = \frac{h}{q_i^2}$.

Per poter fare un paragone con i risultati ottenuti con il metodo dei massimi andiamo a prendere in esame le stesse due lesioni: N1644 e N353.

- Con la scelta della lesione N1644 come query abbiamo ottenuto:

Primo retrieval			Secondo retrieval		
distanze	immagini	voti	distanze	immagini	voti
0	N1644		0	N1644	
0.111466	N147	0	0.120422	N1709	2
0.111701	M1914	1	0.124211	N2068	1
0.119105	N1842	0	0.129583	M1914	1
0.120053	N5072	0	0.131657	N147	0
0.120793	N2068	1	0.133388	N2185	1
0.127602	N1709	2	0.141921	N1842	0
0.130153	N4916_1	0	0.144144	N4639	0
0.132918	N4851	0	0.146263	N265	0
0.133736	N801	0	0.146315	N3663	0
0.13691	N1464	0	0.148073	N5072	0

In questo caso i risultati ottenuti sono esattamente quelli che ci aspettavamo: l'immagine con valutazione di somiglianza maggiore è salita in prima posizione, seguita immediatamente dalle due immagini con voto 1 recuperate anche dal primo retrieval; il fatto che quelle con voto 0 siano scese ha permesso che l'immagine N2185 (con voto1) salisse tra le prime 10 immagini e in particolare si posizionasse in quinta posizione.

- Nel caso della lesione N353, invece, si ha:

Primo retrieval			Secondo retrieval		
distanze	immagini	voti	distanze	immagini	voti
0	N353		0	N353	
0.120429	N1372	2	0.114612	N1372	2
0.123201	N1033	3	0.119721	N1033	3
0.129344	N1218	0	0.131655	N265	2
0.135424	M334	0	0.132128	N764	1
0.138741	N764	1	0.134798	N1218	0
0.140742	N4755	1	0.138032	M334	0
0.14134	M911	0	0.140399	N1123	0
0.141475	M2958	0	0.140985	N425	0
0.141785	N265	2	0.142679	N786	0
0.142707	N425	0	0.14296	M2958	0

In questo frangente, sebbene le immagini con i voti maggiori vengano fatte salire nelle prime posizioni, l'immagine N4755 con voto 1 viene fatta scendere fuori dalle prime 10 posizionandosi al di sotto di immagini con voto 0.

Anche in questo caso, come nel relevance feedback che sfrutta i massimi, ci sono situazioni in cui l'input dei medici può non venire rispettato: a volte accade che un'immagine con voto basso (0 o 1) rimanga davanti ad una con un voto alto (2 o 3) anche dopo il feedback.
Nel caso dell'algoritmo dei massimi questo accade di rado, ma comunque accade; in questo caso si verifica anche più spesso, probabilmente a causa di una maggiore mescolanza: analizzando gli output ottenuti su tutte le 30 lesioni del database con 295 immagini, infatti, abbiamo riscontrato che l'algoritmo dei massimi dà risultati migliori, seppur lievissimamente.

Quarto test

Considerati i risultati ottenuti con i test precedenti e appurato che, anche nella migliore delle ipotesi, non raggiungono la bontà di quelli ottenuti con la tecnica dei massimi, abbiamo deciso di cambiare il metodo di calcolo dei pesi cercando di ottimizzarli e quindi di trovare la miglior soluzione possibile.
A tal proposito abbiamo creato un *problema di ottimizzazione*, che prevede tre elementi:

1. *le variabili decisionali*, ovvero quelle da ottimizzare: nel nostro caso sono i pesi, con i quali abbiamo creato un vettore di lunghezza 19

$$\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_{18}, \lambda_{19}) \tag{4.10}$$

2. *la funzione obiettivo*, ovvero la relazione funzionale tra le variabili decisionali e altre variabili che deve essere massimizzata o minimizzata; nel nostro caso, poiché vorremmo che le nuove 19 distanze (ottenute dopo aver pesato le 19 distanze iniziali) rispecchiassero il più possibile i valori di distanza δ_i espressi dal medico, andremo a minimizzare la funzione

$$f(\lambda) = ||d \cdot \lambda - \delta||^2 \tag{4.11}$$

dove $\delta = (\delta_1, \ldots, \delta_{10})$ è il vettore di lunghezza 10 in cui vengono inserite le distanze δ_i corrispondenti ai voti espressi dai medici sulle prime 10 immagini, mentre d è una matrice di dimensioni 10×19 che nella colonna i-esima contiene i 19 valori di distanza tra la query e la i-esima immagine rispetto ai 19 descrittori di forma

3. l'*insieme ammissibile*, cioè l'insieme delle alternative ammissibili per la scelta delle variabili decisionali; in questo caso richiediamo che i pesi siamo tutti non negativi perciò

l'insieme ammissibile è quello che esprime il vincolo

$$\lambda_i \geq 0 \quad \forall i = 1, \ldots, 19 \tag{4.12}$$

Pertanto il nostro problema di ottimizzazione è il seguente:

$$\begin{cases} \min f(\lambda) = \min ||d \cdot \lambda - \delta||^2 \\ \lambda_i \geq 0 \quad \forall i = 1, \ldots, 19 \end{cases} \tag{4.13}$$

Innanzitutto abbiamo calcolato la distanza globale D (come media aritmetica delle 19 matrici iniziali) e, sulla base di questa, abbiamo effettuato un primo retrieval ottenendo le prime 10 immagini più vicine alla query per le quali i medici hanno fornito le valutazioni $v_1, \ldots, v_{10}$

distanze	immagini	voti
d_1	x_1	v_1
d_2	x_2	v_2
d_3	x_3	v_3
d_4	x_4	v_4
d_5	x_5	v_5
d_6	x_6	v_6
d_7	x_7	v_7
d_8	x_8	v_8
d_9	x_9	v_9
d_{10}	x_{10}	v_{10}

con $d_1 < d_2 < d_3 < \cdots < d_{10}$ e $v_i = 0, 1, 2, 3$.

Per queste 10 immagini abbiamo ottenuto le distanze $\delta_1, \delta_2, \ldots, \delta_{10}$ basandoci sui voti nel modo seguente:

- $\delta_i = d_{10} + 0.1(d_{10} - d_1)$, se $v_i = 0$;
- $\delta_i = d_{10} - 3 \cdot 0.1(d_{10} - d_1)$, se $v_i = 1$;
- $\delta_i = d_{10} - 7 \cdot 0.1(d_{10} - d_1)$, se $v_i = 2$;
- $\delta_i = d_{10} - 11 \cdot 0.1(d_{10} - d_1)$, se $v_i = 3$;

e costruito il vettore δ.

A questo punto abbiamo ottimizzato la funzione $f(\lambda)$ e risolto il sistema di ottimizzazione precedentemente descritto implementando un metodo del gradiente proiettato con regola di Armijo.

Abbiamo poi testato questo metodo sulle 30 lesioni del database con 295 immagini ottenendo risultati decisamente migliori e soddisfacenti:

- Con la scelta della lesione N1644 come query abbiamo ottenuto:

Primo retrieval			Secondo retrieval		
distanze	immagini	voti	distanze	immagini	voti
0	N1644		0	N1644	
0.111466	N147	0	0.119163	N1709	2
0.111701	M1914	1	0.120727	N4972	0
0.119105	N1842	0	0.129054	N2068	1
0.120053	N5072	0	0.129757	M1914	1
0.120793	N2068	1	0.135188	N470	0
0.127602	N1709	2	0.137516	N4942	0
0.130153	N4916_1	0	0.139114	N147	0
0.132918	N4851	0	0.139357	N5072	0
0.133736	N801	0	0.139375	N1464	0
0.13691	N1464	0	0.139413	N1842	0

In questo caso la somma dei voti è la stessa sia dopo il primo retrieval che dopo il secondo, ma si osserva che le immagini con voto più alto salgono nelle prime posizione lasciando scendere quelle con voto 0.

- Nel caso della lesione N353, invece, si ha:

Primo retrieval			Secondo retrieval		
distanze	immagini	voti	distanze	immagini	voti
0	N353		0	N353	
0.120429	N1372	2	0.118283	N1033	3
0.123201	N1033	3	0.126334	N3667	3
0.129344	N1218	0	0.126959	N1372	2
0.135424	M334	0	0.127203	N265	2
0.138741	N764	1	0.136031	N764	1
0.140742	N4755	1	0.136207	N4755	1
0.14134	M911	0	0.143701	N1123	0
0.141475	M2958	0	0.144121	N317	0
0.141785	N265	2	0.144663	M334	0
0.142707	N425	0	0.14479	M2958	0

In questo caso il miglioramento è ancora più evidente: non solo le immagini con voto alto salgono nelle prime posizioni, ma aumenta anche la somma dei voti delle prime 10 immagini, che dopo il primo retrieval è 9 e dopo il feedback sale a 12; questo incremento è dovuto al fatto che, come ci aspettavamo, l'algoritmo va a recuperare l'immagine N3667 (che prima si trovava molto in basso) con voto 3 portandola nelle primissime posizioni.

Questo comportamento è stato riscontrato praticamente in tutte e 30 le immagini analizzate e ci ha permesso di dire che sia a livello locale che globale c'è stato un notevole miglioramento di questo metodo rispetto alla prima tecnica di relevance feedback studiato in questa tesi:

- a livello globale il miglioramento è giustificato da un notevole aumento complessivo delle valutazioni del secondo retrieval rispetto al primo; in particolare la somma dei voti nei due retrieval è:

Somma voti	
primo retrieval	secondo retrieval
174	237

  pertanto con questo approccio si ottiene un miglioramento di 63 voti.

- a livello locale, osservando l'andamento dell'algoritmo su ogni immagine, abbiamo osservato il comportamento descritto per le due immagini N1644 e N353: la somma dei voti del secondo retrieval è rimasta invariata o addirittura aumentata rispetto a quella precedente al feedback e le immagini con voto alto sono salite sempre nelle prime posizioni lasciando scendere quelle con valutazioni minime.

L'unica situazione in cui abbiamo riscontrato un lieve peggioramento riguarda lo studio della lesione N584:

Primo retrieval			Secondo retrieval		
distanze	immagini	voti	distanze	immagini	voti
0	N584		0	N584	
0.068653	N778	1	0.072433	M975	3
0.088535	M1964	0	0.078521	N541	0
0.090029	N313	1	0.079263	N985	0
0.09009	M975	3	0.086252	N411	0
0.091369	N437	1	0.087037	N1014	1
0.093847	N3852	1	0.088603	N778	1
0.096918	M3710	0	0.088679	N3852	1
0.099784	M911	1	0.088863	N368	1
0.101749	M828	1	0.08956	M3580	0
0.104486	N3959	0	0.090482	M2632	0

In questa particolare situazione, la somma dei voti dopo il secondo retrieval è diminuita rispetto a quella calcolata dopo il primo retrieval (da 9 a 7), nonostante l'immagine con voto massimo sia comunque salita in prima posizione.
Questo comportamento si è verificato perché il primo retrieval effettuato su questa query ha prodotto un'immagine con una valutazione massima 3 e diverse immagini con valutazione 1; questo ha fatto sì che l'ottimizzazione facesse scendere alcune immagini con voto 1 nel tentativo di recuperarne altre con valutazione più alta, senza però trovarne.

Nonostante questo lieve peggioramento i risultati mostrati da questo approccio sono decisamente molto buoni e soddisfacenti in assoluto e comunque preferibili rispetto a quelli, seppur buoni, ottenuti dal relevance feedback basato sui massimi:

Somma voti					
rel. feed. massimi			rel. feed. somme		
primo ret.	secondo ret.	miglioramento	primo ret.	secondo ret.	miglioramento
173	190	17	174	237	63

Si osserva da questi risultati che questo secondo metodo di relevance feedback risulta essere migliore del precedente sia perché il totale dei voti è notevolmente maggiore, sia perché il miglioramento che si ottiene è più elevato nonostante la base di partenza sia leggermente più elevata nel secondo caso rispetto al primo.

Conclusioni

In questa tesi abbiamo proposto due diversi modelli per la tecnica di relevance feedback, e in particolare per il calcolo dei pesi $\lambda_1, \ldots, \lambda_{19}$ necessari a migliorare il recupero di immagini dermatologiche da un database tenendo conto anche del giudizio di somiglianza espresso dai medici.
Questi modelli sono risultati adatti allo scopo di migliorare la situazione attuale, in modo particolare il secondo metodo basato sulle somme pesate che, dopo diversi tentativi di ottenere pesi mirati al nostro scopo, ha prodotto risultati migliori del primo metodo basato sui massimi. Infatti, sebbene i primi risultati fossero già di per sé positivi, il secondo metodo si è dimostrato migliore sia perché la somma globale dei voti dopo il feedback è maggiore in questo caso rispetto al primo, sia perché il miglioramento dei voti tra il primo e il secondo retrieval è notevolmente maggiore. Inoltre, esaminando i retrieval ottenuti dopo il feedback effettuato con il secondo algoritmo, abbiamo osservato il comportamento sperato: le immagini con i voti maggiori si sono collocate nelle primissime posizioni e quelle con voto minimo sono scese lasciando che, in molte situazioni, venissero recuperate immagini con voti più alti che non comparivano dopo il primo retrieval.
In vista di possibili sperimentazioni future, si potranno sicuramente pensare e testare altri schemi di individuazione di pesi ottimali al fine di migliorare ulteriormente la resa feedback, così come abbiamo parzialmente fatto in questo lavoro di tesi.
Inoltre, poiché momentaneamente l' abbiamo solo simulato, potrà essere fatta una sperimentazione accurata di vero e proprio relevance feedback: con questa sperimentazione reale si potranno provare anche due fasi di relevance feedback anziché una sola, come fatto fino ad ora.

Appendice A

Il codice C

A.1 Relevance feedback col metodo dei massimi

Per testare il metodo di relevance feedback che sfrutta i massimi abbiamo, come prima cosa, calcolato la matrice dei massimi: abbiamo letto le 19 matrici distanza $N \times N$ dai file Mat01.tx, ..., Mat19.txt e stampato sul file MaxMat.txt la matrice dei massimi ottenuta confrontando le 19 matrici precedenti elemento per elemento e prendendo il massimo in ogni posizione (i, j).
A questo punto abbiamo scelto una generica colonna "index" della matrice dei massimi come query rispetto alla quale abbiamo fatto il retrieval sfruttando la funzione di riordinamento Quicksort:

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>

//Riordina il vettore di double x ristretto fra l'indice left e l'indice right in ordine
void Quicksort3(double *x, int *sigma, int left, int right)
{
    int i = left;
    int j = right;
    double tmp;
    int tmp2;
```

```
    double pivot = x[(left + right)>>1];

    while (i <= j)
    {
        while (x[i] < pivot)
            i++;
        while (x[j] > pivot)
            j--;
        if (i <= j)
        {
            tmp = x[i];
            tmp2 = sigma[i];
            x[i] = x[j];
            sigma[i] = sigma[j];
            x[j]   = tmp;
            sigma[j] = tmp2;
            i++;
            j--;
        }
    }
    if (left < j)
        Quicksort3(x, sigma, left, j);
    if (i < right)
        Quicksort3(x, sigma, i, right);
}

int main()
{

    /************************************************************************
     *           scelta di una colonna della matrice dei massimi            *
     ************************************************************************/

    double Colonna[295];
```

```
double* matmax;
matmax=(double*)calloc(295*295, sizeof(double));
FILE *f;    //leggo la matrice MaxMat
f=fopen("/Users/Eleonora/Desktop/programmazioneC/matmax/MaxMat.txt","r");
int i,j;
for (i=0; i<295; i++)
{
    for (j=0; j<295; j++)
        fscanf( f, "%lf;", &matmax[i+j*295] );
}
fclose(f);


int index=224;     //prendo una colonna di MaxMat
for (i=0; i<295; i++)
    Colonna[i]=matmax[i+295*index];

/*********************************
 *   Relevance della colonna  *
 *********************************/

int sigma[295];
for (i=0; i<295; i++)
    sigma[i]=i;

Quicksort3(Colonna,sigma,0,294);
```

A questo punto abbiamo creato i pesi $\lambda_1, \ldots, \lambda_{19}$ con cui modificare le 19 matrici distanza iniziali sfruttando i voti. Nel codice che segue vediamo l'assegnazione di voti casuali tra 0 e 1 alle prime 10 immagini recuperate dal retrieval e le conseguenti distanze ottenute come $\delta_i = 1 - voto_i$. Nel corso dei vari test effettuati, i voti e le distanze sono stati scelti in modo mirato come descritto nel capitolo 4.

```
 /************************************
 *  Assegnazione di voti tra 0 e 1  *
 ************************************/
 for(i=0; i<10; i++)
    Colonna[i]=rand()/(double)RAND_MAX;
/************************
 *  Creazione dei pesi  *
 ************************/
int l,k;
FILE *fp;
double lambda[19];
for (i=0; i<19; i++)
    lambda[i]=1;
double* mat_dist;
mat_dist=(double*)malloc(19*295*295*sizeof(double));
char num;
char* str;
char* string;
string=(char*)malloc(500*sizeof(char));
int len=(int)sizeof("/Users/Eleonora/Desktop/programmazioneC/AllMatrix2/Mat");
double* newmat;
newmat=(double*)malloc(19*295*295*sizeof(double));

for (l=1; l<11; l++) //per ogni voto creo 19 pesi lambda_i
{
    for (k=1; k<20; k++)
    {

        if (k<10)
        {
            num=k+48;
            str=(char*)malloc(500*sizeof(char));
            strncpy(str,"/Users/Eleonora/Desktop/programmazioneC/AllMatrix2/
            Mat\0",len);
            strcat(str,"0\0");
```

```
    strcat(str, &num);
    strcat(str,"\0");
    strcat(str,".txt\0");
    strncpy(string,str,len+6);
    free(str);
}
else
{
    num=k+38;
    str=(char*)malloc(500*sizeof(char));
    strncpy(str,"/Users/Eleonora/Desktop/programmazioneC/AllMatrix2/
    Mat\0",len);
    strcat(str,"1\0");
    strcat(str,&num);
    strcat(str,"\0");
    strcat(str,".txt\0");
    strncpy(string,str,len+6);
    free(str);
}


fp=fopen(string,"r");
for (i=0; i<295; i++)
{
    for (j=0; j<295; j++)
        fscanf( fp, "%lf;", &mat_dist[i+j*295+(k-1)*295*295] );
}
fclose(fp);
double x=1-Colonna[l];
double y=mat_dist[sigma[l]+index*295+(k-1)*295*295];
if (x/y < 1)  //prendo il minimo tra lambda_k e x/y
{ if (lambda[k-1]< x/y)
        lambda[k-1]=lambda[k-1];
    else
        lambda[k-1]=x/y;}
```

```
            /************************************
             *   creo le nuove distanze pesate    *
             ************************************/
            for (i=0; i<295*295; i++)
                newmat[i+(k-1)*295*295]=lambda[k-1]*mat_dist[i+(k-1)*295*295];
        }
    }
    free(string);
    free(mat_dist);
    free(matmax);
```

Una volta creata la nuova famiglia con le distanze modificate, abbiamo ricalcolato e stampato su file la nuova matrice dei massimi NewMaxMat.txt, sulla quale abbiamo poi effettuato i retrieval post-feedback confrontandoli con quelli precedenti al feedback.

```
/*******************************
*  creo il vettore dei massimi  *
*******************************/
double* newmatmax;
newmatmax=(double*)malloc(295*295*sizeof(double));
int alpha,h;
double v[19];
double massimo;
for (h=0; h<295*295; h++)
{
    for (alpha=0; alpha<19; alpha++)
        v[alpha]=newmat[h+alpha*295*295];
    massimo= v[0];
    for(i=0; i<19; i++)
    {
        if ( v[i]>massimo )
            massimo=v[i];
    }
```

```
        newmatmax[h]=massimo;
    }

    //creo un file con la nuova matrice dei massimi

    FILE *fmax;
    fmax=fopen("/Users/Eleonora/Desktop/programmazioneC/Feedback/NewMaxMat.txt","w");
    for (i=0; i<295; i++)
    {
        for (j=0; j<295; j++)
            fprintf( fmax, "%lf;", newmatmax[i+j*295]);
        fprintf(fmax, "\n");
    }
    fclose(fmax);

    free(newmatmax);
    free(newmat);
    return 0;
}
```

Il codice appena descritto è quello che abbiamo utilizzato per i test sul database con 295 immagini, ma è del tutto analogo il codice sfruttato per studiare il database con 200 immagini: infatti è sufficiente sostituire 295 con 200 oppure inizializzare una variabile N da sostituire nel codice al posto di 295 e attribuire alla variabile stessa il valore 295 piuttosto che 200 a seconda del database che si sta studiando.

A.2 Relevance feedback col metodo delle somme pesate

Per effetture i test con il metodo di relevance feedback con le somme pesate abbiamo innanzitutto calcolato la matrice media: anche in questo caso abbiamo letto le 19 matrici distanza $N \times N$ dai file Mat01.tx, ..., Mat19.txt e stampato sul file Max_Media.txt la matrice media calcolata sommando termine a termine le 19 matrici precedenti e dividendo per 19 il totale ottenuto in ogni posizione (i, j).
A questo punto abbiamo scelto una generica colonna "index" della matrice media come query rispetto alla quale abbiamo fatto il retrieval sfruttando, analogamente al caso precedente, la funzione di riordinamento Quicksort.

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>

int main()
{

    /*********************************************************
     *        scelta di una colonna della matrice media      *
     *********************************************************/
    double Colonna[295];
    double* matmed;
    matmed=(double*)calloc(295*295, sizeof(double));
    FILE *f;    //leggo la matrice MaxMat
    f=fopen("/Users/Eleonora/Desktop/programmazioneC/TestFeedback/Data295/
    Mat_Media.txt","r");
    int i,j;
    for (i=0; i<295; i++)
    {
        for (j=0; j<295; j++)
            fscanf( f, "%lf;", &matmed[i+j*295] );
    }
    fclose(f);
```

```
int index=2;     //prendo una colonna di Mat_Media
for (i=0; i<295; i++)
    Colonna[i]=matmed[i+295*index];

/*******************************
 *   Relevance della colonna     *
 *******************************/
int sigma[295];
for (i=0; i<295; i++)
    sigma[i]=i;

Quicksort3(Colonna,sigma,0,294);
```

Per poter creare i pesi $\lambda_1, \ldots, \lambda_{19}$ con cui pesare le 19 matrici distanza iniziali abbiamo sfruttato i voti (dapprima scelti in modo casuale tra 0 e 1 e poi modificati sulla base di quelli reali per i vari test) delle prime 10 immagini recuperate dal retrieval ottenendo le conseguenti distanze δ_i.

```
/*********************************
 *  Assegnazione di voti tra 0 e 1  *
 *********************************/
/*for(i=0; i<10; i++)
 Colonna[i]=rand()/(double)RAND_MAX;*/

/************************
 *  Creazione dei pesi  *
 ************************/
int l,k;
FILE *fp;
double lambda[19];
for (i=0; i<19; i++)
    lambda[i]=0;
double* mat_dist;
mat_dist=(double*)malloc(19*295*295*sizeof(double));
```

```
char num;
char* str;
char* string;
string=(char*)malloc(500*sizeof(char));
int len=(int)sizeof("/Users/Eleonora/Desktop/programmazioneC/AllMatrix2/Mat");
double* newmat;
newmat=(double*)malloc(19*295*295*sizeof(double));

for (l=1; l<11; l++) //per ogni voto creo 19 pesi lambda_i
{
    for (k=1; k<20; k++)
    {
        if (k<10)
        {
            num=k+48;
            str=(char*)malloc(500*sizeof(char));
            strncpy(str,"/Users/Eleonora/Desktop/programmazioneC/AllMatrix2/
            Mat\0",len);
            strcat(str,"0\0");
            strcat(str, &num);
            strcat(str,"\0");
            strcat(str,".txt\0");
            strncpy(string,str,len+6);
            free(str);
        }
        else
        {
            num=k+38;
            str=(char*)malloc(500*sizeof(char));
            strncpy(str,"/Users/Eleonora/Desktop/programmazioneC/AllMatrix2/
            Mat\0",len);
            strcat(str,"1\0");
            strcat(str,&num);
            strcat(str,"\0");
            strcat(str,".txt\0");
```

```
            strncpy(string,str,len+6);
            free(str);
        }


        fp=fopen(string,"r");
        for (i=0; i<295; i++)
        {
            for (j=0; j<295; j++)
                fscanf( fp, "%lf;", &mat_dist[i+j*295+(k-1)*295*295] );
        }
        fclose{fp};
    }
}
double q[19];
for(i=0;i<19;i++)
    q[i]=0;

/******************
*   creo i pesi   *
*******************/
for (l=1;l<11;l++)
{
    double delta= 1-Colonna[l];
    //double delta= Delta[l-1];
    for (k=0;k<19;k++)
    {
        double d= mat_dist[sigma[l]+index*295+k*295*295];
        if (delta-d >0)
            q[k]=q[k]+(delta-d);
        else
            q[k]=q[k]+(d-delta);
    }
}
```

```
//Poi divido per 10 rispetto a tutti i k
int s=0;
for (k=0;k<19;k++)
{
    if (q[k]>0.0)
        s++;
    q[k]/=10;
}

//h risolve l'equazione h/q_1+.....+h/q_19=1
if (s==0)
{
    for (k=0;k<19;k++)
        lambda[k]=(double)1.0;
}
else if (s<19)
{
    for (k=0;k<19;k++)
    {
        if (q[k]>0.0)
            lambda[k]=(double)0.0;
        else
            lambda[k]=(double)(1.0)/(double)(s);
    }
}
else
{
    double temp=0.0;
    double h;
    for (k=0;k<19;k++)
        temp+=(double)(1.0)/q[k];
        h=(double)(1.0)/temp;
    for (k=0;k<19;k++)
        lambda[k]=h/q[k];
}
```

```
FILE *pesi;
pesi=fopen("/Users/Eleonora/Desktop/Pesi.txt","w");
for (k=0; k<19;k++)
    fprintf(pesi,"%lf;\n",lambda[k]);
fclose(pesi);

 /************************************
 *   creo le nuove distanze pesate    *
 ************************************/
for (k=0;k<19;k++)
{
    for (i=0; i<295*295; i++)
        newmat[i+k*295*295]=lambda[k]*mat_dist[i+(k)*295*295];//peso le distanze
}
```

Dopo aver ottenuto la nuova famiglia con le distanze modificate, abbiamo ricalcolato e stampato su file la nuova matrice dei media New_Mat_Media.txt, rispetto alla quale abbiamo poi effettuato i retrieval post-feedback confrontandoli con quelli precedenti al feedback.

```
 /*************************************
 *  creo la nuova matrice media pesata *
 *************************************/
double* newmatmed;
newmatmed=(double*)calloc(295*295,sizeof(double));
for(i=0; i<295*295; i++)  //creo la "matrice" media
{
    for(k=0; k<19; k++)
        newmatmed[i]=(newmatmed[i]+newmat[i+k*295*295]);
    newmatmed[i]=newmatmed[i]/19;
}
```

```
    //stampo su file la matrice media
    FILE *fm;
    //fm=fopen("/Users/Eleonora/Desktop/programmazioneC/TestFeedback/Data295/Feedback_sc
    for (i=0; i<295; i++)
    {
        for (j=0; j<295; j++)
            fprintf( fm, "%lf;", newmatmed[i+j*295]);
        fprintf(fm, "\n");
    }
    fclose(fm);

    free(string);
    free(mat_dist);
    free(matmed);
    free(newmatmed);

    return 0;
}
```

Anche questo codice, ideato per effettuare i test sul database da 295 immagini, può essere modificato in modo molto semplice e adattato per testare il database PH2: basta sostituire 295 con 200 o inizializzare una variabile N a cui attribuire il valore 295 piuttosto che 200 a seconda dei test che si desidera eseguire.

Bibliografia

[1] http://www.airc.it/cancro/tumori/melanoma-cutaneo

[2] http://www.humanitas.it/malattie/melanoma

[3] http://www.stanganelliignazio.it/

[4] I. Stanganelli
Il sole e la pelle
Dall'Emilia-Romagna un modello di ricerca e di intervento educazionale
Edizioni Minerva Medica, 2015

[5] F. Foschi, E. Loli Piccolomini
Valutazione di un sistema di recupero di immagini dermatologiche
Tesi di Laurea in Topologia Algebrica, Bologna, Anno Accademico 2013/2014

[6] F. Di Dio
Algoritmi di omologia persistente k-dimensionale per la diagnosi di lesioni melanocitiche
Tesi di Laurea in Topologia Algebrica, Bologna, Anno Accademico 2010/2011

[7] M. Ferri
L'incredibile ubiquità della topologia persistente
http://maddmaths.simai.eu/divulgazione/focus/lincredibile-ubiquita-della-topologia-persistente/, 2015/2016

[8] M. Ferri
Persistent topology for natural data analysis-A survey
https://arxiv.org/abs/1706.00411, 2017

[9] D. Giorgi, P. Frosini, M. Spagnuolo, B. Falcidieno
3D relevance feedback via multilevel relevance judgements
The Visual Computer, vol. 26 (2010), 1321-1338

[10] M. D'Amico
A new optimal algorithm for computing size function of shapes
Proc. CVPRIP Algorithms III, International Conference of Computer Vision, Pattern Recognition and Image Processing (2000), 107/110

[11] M. Guijarro, G. Pajaresb, I. Riomorosc, P. J. Herrerad, X. P. Burgos-Artizzue, A. Ribeiroe
Automatic segmentation of relevant textures in agricultural images
Computers and Electronics in Agriculture 75 (1) (2011), 75-83

[12] M. Ferri, I. Tomba, A. Visotti, I. Stanganelli
A feasibility study for a persistent homology based k-Nearest Neighbor search algorithm in melanoma detection
J. Math. Imaging Vis. 57 (2017), 324-339

[13] https://www.fc.up.pt/addi/ph2%20database.html