

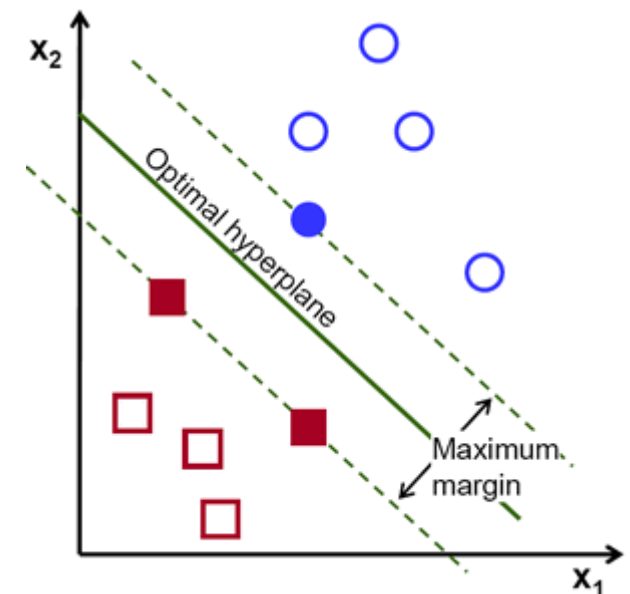
SPECTRAL ANALYSIS AND UNSUPERVISED SVM CLASSIFICATION FOR SKIN HYPER-PIGMENTATION CLASSIFICATION

*Sylvain Prigent 1 , Xavier Descombes 1 , Didier
Zugaj 2 , Josiane Zerubia 1*

Presentazione e codice di
Illuminato Luca Costantino (W82/000033)

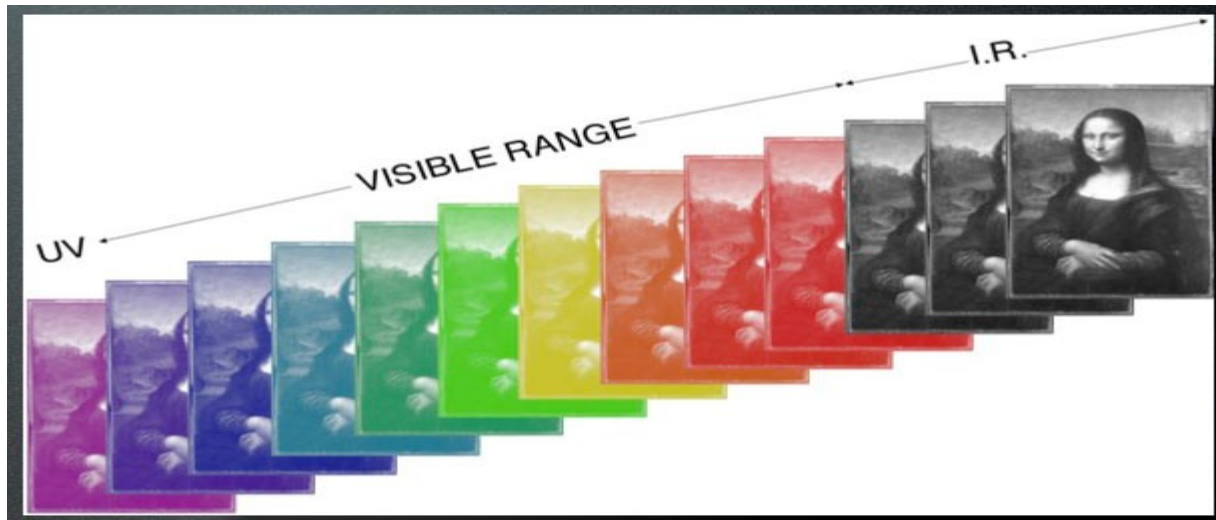
La problematica e la soluzione

- Si vuole analizzare delle immagini multispettrale della pelle umana e capire se è affetta da iperpigmentazione oppure no.
- Il migliore classificatore per questo genere di problemi è l'SVM (Support Vector Machine), ma è uno strumento **supervisionato**.
- Tramite l'analisi spettrale si fornisce un training set al classificatore, evitando così l'obbligo dell'interazione con un operatore umano.



Analisi spettrale

- Il dato che viene fornito in input è un'immagine multispettrale di una porzione di pelle contenente sia parti sane che parti malate.
- Un'immagine multispettrale è una matrice $m \times n \times k$, dove k è il numero dei livelli di lunghezza d'onda registrati.
- Per esempio, in una banda I possiamo memorizzare i valori di assorbimento della melanina alla lunghezza d'onda di 700 nanometri.

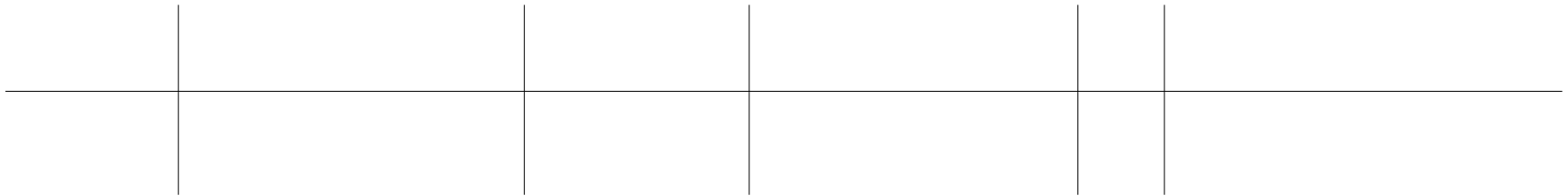


Analisi spettrale

- Il problema è che è un tipo di dato le cui dimensioni rallentano l'esecuzione di un qualunque algoritmo che debba analizzare pixel-per-pixel l'immagine.
- Inoltre, molti livelli contengono informazioni che variano poco da livello a livello. Cioè sono ridondanti e potrebbero essere eliminate (o perlomeno sintetizzate) senza grossi problemi.

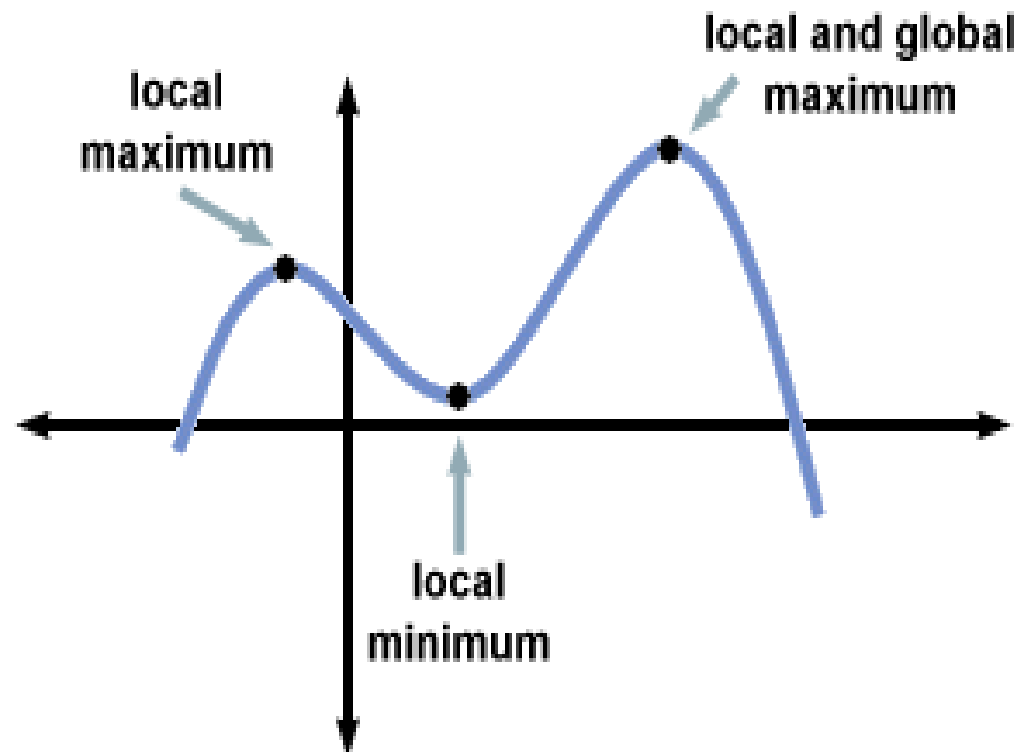
Analisi spettrale

- 1) Creiamo una mappa 3D delle variazioni di assorbimento tra due livelli contigui ($k-1, k$) e chiamiamola **FIS**.
- 2) Per ogni livello di FIS, memorizziamo a parte la variazione totale. Chiamiamo questo vettore **FISA**.
- 3) Determiniamo le bande dove sono presenti dei massimi locali e memorizziamone le locazioni.
- 4) Abbiamo fatto un primo partizionamento, abbastanza grossolano.



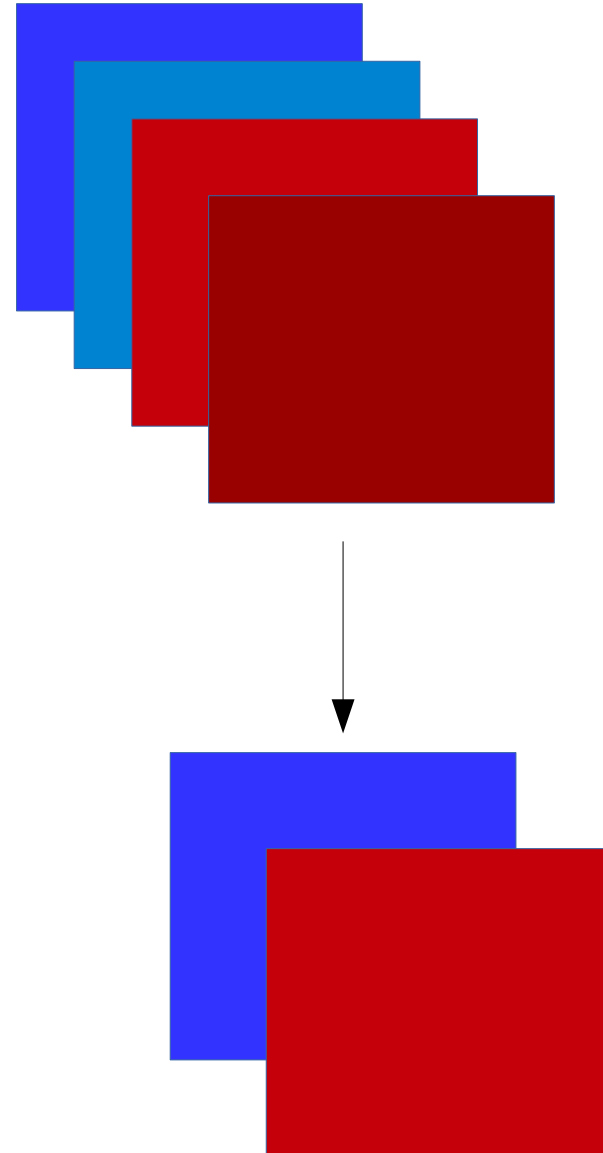
Analisi spettrale

- Scegliamo casualmente un bordo tra i K disponibili e spostiamolo di 1 posizione verso destra.
- Se la varianza intraclass complessiva è diminuita, accettiamo il cambiamento e aggiorniamo la locazione dei bordi.
- Altrimenti rifiutiamo questa proposta e continuiamo a iterare. Iteriamo un numero grande di volte.
Man mano che si itera, convergeremo al partizionamento ottimale.



Analisi spettrale

- A questo punto, per ognuno dei $K+1$ gruppi, abbiamo dei livelli che ci raccontano più o meno la stessa cosa! Possiamo benissimo fondere i livelli di ogni gruppo in uno solo.
- Così facendo abbiamo ridotto i livelli dell'immagine multispettrale di $N_b - (k+1)$ elementi.



Costruzione di un classificatore

- Un classificatore SVM, per poter operare, ha bisogno di un training set adeguato per potersi addestrare.
- Pertanto, scegliamo il livello di FIS più interessante, cioè quello con la variazione totale massima e chiamiamo questa mappa **FISK**.
- Determiniamo una soglia di assorbimento, superata la quale essa diventa patologica e chiamiamola **T1**.
- $T1 = \text{media}(\text{FISK}) + t * \text{varianza}(\text{FISK})$
- **t** è un parametro di controllo che nel paper viene impostato a 0.5

Costruzione di un classificatore

- Scegliamo N pixel il cui valore è più vicino alla soglia T_1 , sia da destra che da sinistra.
- I pixel con valore inferiore a T_1 saranno i nostri esempi di pelle sana, gli altri saranno gli esempi di pelle malata.
- Così come per gli iris di Fisher avevamo 150 osservazioni per 4 features, qui abbiamo N osservazioni per $K+1$ features (i valori per ogni livello dopo il partizionamento spettrale).
- Passiamo questo training set al classificatore.

Riduzione della dimensionalità

- In generale, quando si hanno N features, i nostri dati *vivono* in uno spazio dove ogni feature è una coordinata (spazio delle feature).
- Come dicevamo prima, il nostro problema è riuscire a rinchiudere in regioni separate gli elementi di classi diverse (segmentazione).
- Ma più feature abbiamo e più la dimensione del dataset da gestire aumenta. Per giunta, non sempre più feature si hanno e meglio è: alcune di esse non solo non sono determinanti per distinguere due classi, ma aumentano solo la confusione. Per esempio, se vogliamo classificare delle persone in base al sesso, non ha senso rilevare caratteristiche come il colore degli occhi.

Riduzione della dimensione

- Allora, prima di andare avanti con la nostra classificazione, dobbiamo ridurre la dimensionalità del nostro input, scegliendo in maniera oculata quali feature mantenere e quali buttare via.
- Formalmente, ci chiediamo qual'è il migliore sottospazio proiettivo che approssimi in maniera ottimale il nostro dataset riducendone la complessità.

Riduzione della dimensione

- Questa ricerca di un sottospazio a dimensionalità più piccola rispetto all'originaria che però ne preservi almeno le informazioni più importanti si chiama in inglese **Projection Pursuit**.
- Una tecnica che si può usare è la **massimizzazione della divergenza di Kullback-Leibler**.
- Questa particolare tecnica è nata per effettuare feature selection. Dati N feature rilevate su elementi di X classi (nel nostro caso 2), vogliamo mantenere solo quelle che massimizzano la mutua informazione tra le classi.
- Dato che nel nostro caso **#feature = #livelli**, stiamo effettuando un ulteriore riduzione della dimensione della nostra immagine.
- Su questa immagine sarà effettuata la classificazione vera e propria.

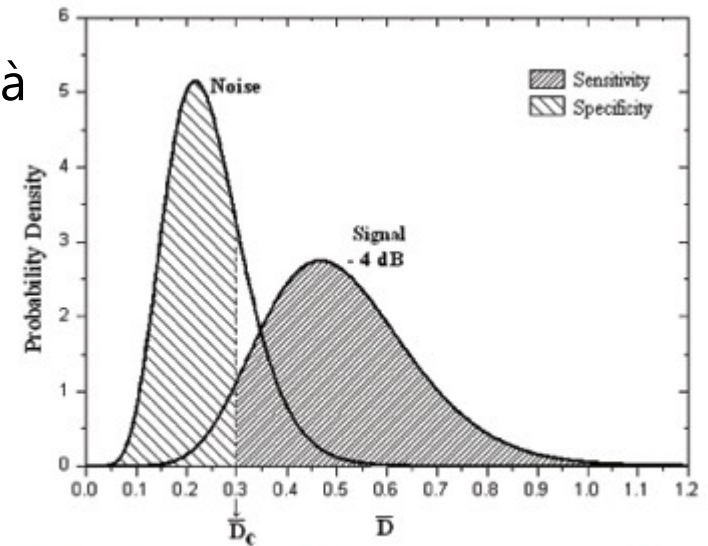


FIG. 4: Areas corresponding to specificity and sensitivity under probability densities $f(\bar{D}|-)$ (for simulated noise only) and $f(\bar{D}|+)$ (for simulated signal with SNR=-4dB).

Riduzione della dimensione

- Quante riduzioni abbiamo fatto nel complesso?
- Due: con la prima riduzione, durante l'analisi spettrale, abbiamo eliminato l'informazione **ridondante**; con la seconda riduzione, durante il project pursuit, abbiamo eliminato l'informazione **poco significativa** (quella che non ci aiutava a discriminare le classi).
- Se N_b è il numero originale di livelli, K è il numero di livelli dopo la prima fase e K' è il numero di livelli dopo la seconda, abbiamo ridotto l'immagine di $N_b - K'$ livelli, un numero molto alto!
- Non solo non abbiamo perso informazione importante, ma l'abbiamo pure estratta e isolata rispetto a tutta quella disponibile. Questo fa capire la teorica potenza di questa doppia riduzione ai fini della classificazione.

Ricapitolando...

ANALISI SPETTRALE

Partizionamento dei livelli in K gruppi mediante analisi della varianza intraclassa
Fusione dei livelli di ogni gruppo in uno solo



COSTRUZIONE CLASSIFICATORE SVM

Costruzione del training set (gli N pixel più vicini ad una soglia T1)
Etichettatura pixel di training $> T1$ come malati, altrimenti come sani
Addestramento classificatore



DATA REDUCTION

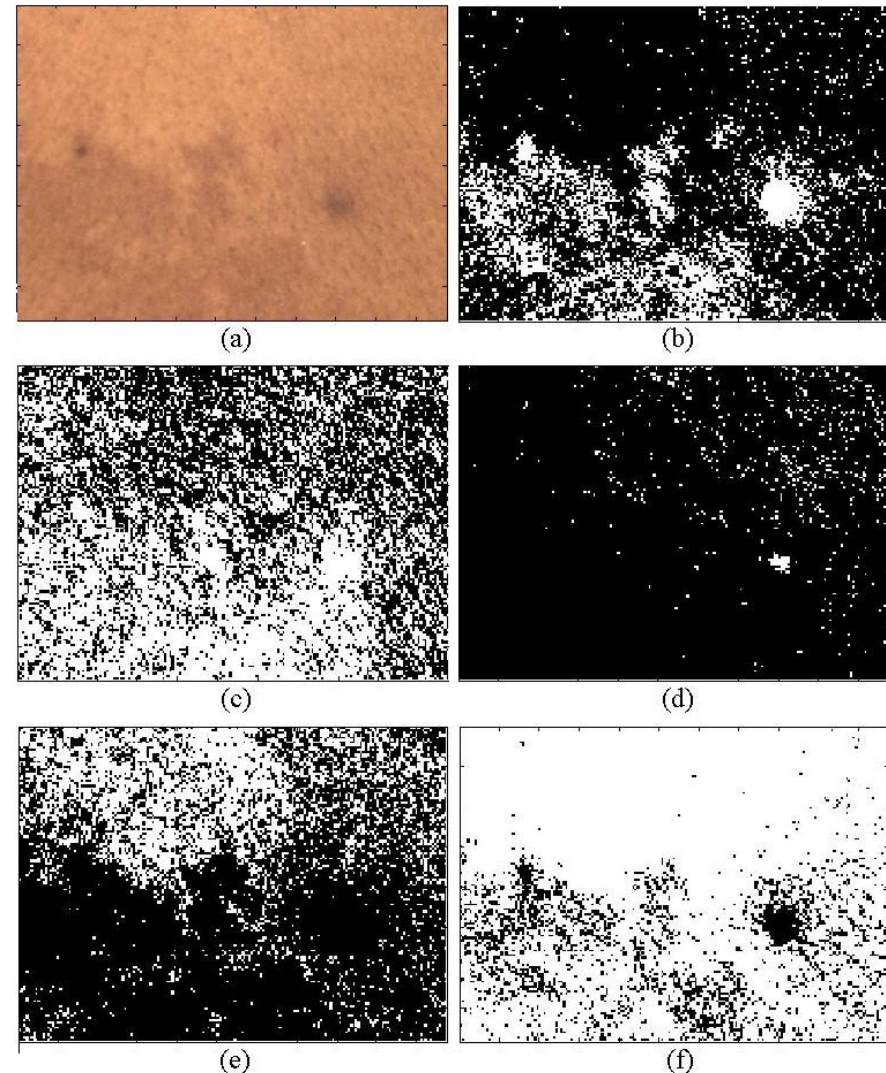
Feature selection mediante massimizzazione della distanza di Kullback-Leibler
Eliminazione dei livelli corrispondenti alle feature scartate



CLASSIFICAZIONE !

Classificatore in azione

- Gli autore del paper hanno usato come test un'immagine multispettrale a 18 bande, di 25 nm ciascuna, dai 405 nm a 970 nm.
- L'analisi spettrale ha garantito un partizionamento ottimale in $K=5$ gruppi e si è evidenziato che nelle bande a 470, 660 e 700 nm (dove l'azione della melanina è più visibile) erano presenti delle aree sospette.
- Il livello che ha fornito i pixel di training aveva come indice il bordo del quinto gruppo (rispetto al partizionamento spettrale).



Considerazioni

- Il PP così effettuato si presta bene al calcolo parallelo perchè contiene molte operazioni su livelli e gruppi indipendenti.
- Il poter usare SVM con una supervisione automatizzata è sicuramente ottimo, soprattutto considerando che stiamo analizzando un'immagine senza particolari pattern.
- Con qualche piccola variazione (basato sulla scelta della banda adiacente a quella in esame) si può estendere questo procedimento alle immagini iperspettrali.

Risorse

- <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5652072>
 - Il paper originale.
- <https://github.com/lucaluke88/codici-vari/tree/master/ClassificatoreSVM4MelanomaDiagnosis>
 - La mia implementazione (in aggiornamento)
su Github

**Grazie
per
l'attenzione!**