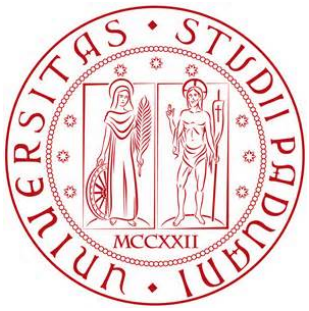
Università degli studi di Padova
Dipartimento di Scienze Statistiche
Corso di Laurea Magistrale in

Scienze Statistiche



Tesi di Laurea

Kernel Riemanniano per la classificazione di immagini

Relatore Prof. Livio Finos
Dipartimento di Psicologia dello Sviluppo e della Socializzazione

Laureanda Arianna Bellino
Matricola N 1104039

Anno Accademico 2016/2017

Indice

Introduzione

Fin dalla creazione della terra, la storia stessa ha provato che il cambiamento è indispensabile per la sopravvivenza. Il mondo attuale è fatto più che mai di cambiamento, e con le nuove possibilità di memorizzazione e con l'avanzamento delle tecnologie ha preso particolare importanza la classificazione delle immagini. La realtà informatica che circonda ognuno di noi è fatta di foto, immagini e documenti digitali, e la classificazione delle immagini è applicata in qualsiasi contesto: nel riconoscimento degli oggetti, dei volti, del testo, ed in qualsiasi ambito, da quello medico a quello geologico.
Un esempio dell'importanza del riconoscimento automatico è dato dal'individuazione degli elementi geomorfologici del paesaggio; mappare le forme sulla mappa geografica è un procedimento che implica un forte grado di soggettività ed un alto livello d'esperienza, quindi l'automatizzazione del processo di riconoscimento può velocizzare e ridurre l'incertezza nell'individuazione delle forme del paesaggio [Mendicelli 2007].
Ulteriore campo di utilizzo è, come detto, l'ambito clinico: il riconoscimento automatico è applicato nell'identificazione delle patologie a partire dalle radiografie; uno dei limiti maggiori dei metodi radiografici è la soggettività del giudizio visivo dello specialista connessa ai lunghi tempi di processamento, quindi l'automatizzazione della procedura non solo porta ad una riduzione dei tempi di attesa ma anche a una migliore identificazione delle patologie [Ricerche e Catone 2009].
Infine come ultimo esempio, ma lo spazio di applicazione è ben più ampio, si cita il riconoscimento automatico degli oggetti all'interno delle immagini volto alla creazione di sistemi di formulazione automatica di regole associative sulla base delle immagini analizzate.
Ovviamente data l'importanza del riconoscimento automatico delle immagini, mol-

ti dei modelli di classificazione e degli strumenti matematici e statistici vengono già utilizzati in tale contesto.
Lo scopo di questa tesi è quello di applicare l'algebra delle varietà ad alcuni degli strumenti di analisi supervisionata presenti in letteratura per migliorarne la precisione dell'individuazione della classe di appartenenza delle immagini considerate; più precisamente si vuole costruire un algoritmo di classificazione basato sulla minima distanza dalla media Riemanniana ed un modello basato sulla *Support Vector Machine* con l'aggiunta di un nucleo Riemanniano.
Il vantaggio dell'utilizzo della metrica Riemanniana è che permette sia l'elaborazione diretta dei dati sulle varietà sia la costituzione di *kernel* definiti positivi per l'elaborazione su spazi di alta dimensionalità.
Solitamente queste metriche vengono applicate su descrittori di caratteristiche che includono l'intensità e le sue derivate; uno degli aspetti innovativi che porta questa tesi è l'utilizzo delle immagini iniziali adeguatamente pre-processate evitando il calcolo del descrittore. Grande importanza viene data, oltre che al preprocessamento delle immagini iniziali, anche alla creazione di un algoritmo di individuazione della regione di interesse.
Ulteriore innovazione è quella di implementare la *Support Vector Machine* sia a partire dalla funzione di distanza Riemanniana sia riconducendosi al *kernel* a base radiale; nel primo caso si produce un metodo versatile che prende in *input* una funzione di distanza qualsiasi, mentre nel secondo si utilizza il logaritmo matriciale in modo tale da avvalersi dei metodi già presenti in letteratura.
Nel primo capitolo verrà fatto un breve *excursus* sull'analisi Riemanniana soffermandosi sulle matrici definite positive, sulla varietà e sulla distanza Riemanniana.
Nel secondo capitolo si darà un breve cenno sull'utilizzo della funzione *kernel* e sui modelli che la utilizzano per riprodurre lo spazio delle caratteristiche iniziali.
Nel terzo capitolo si spiegherà come nel caso dell'analisi Riemanniana esiste la necessità di operare su matrici definite positive e, per tale motivo, è d'obbligo il pre-processamento delle matrici di immagini ma anche l'utilizzo di metodi di regolarizzazione per completare il rango delle matrici. Inoltre si proporranno i due differenti metodi di classificazione, il primo che utilizza la minima distanza dalla media Riemanniana di ogni classe per effettuare la classificazione, mentre il secondo che applica il *kernel* Riemanniano ad una *Support Vector Machine* per calcolare il confine decisionale nella classificazione.

Seguirà un capitolo dedicato ai dati ed alla gestione del *dateset* considerato, focalizzando l'attenzione sull'identificazione della regione di interesse all'interno delle immagini.
Nell'ultimo capito, infine, si analizzeranno i risultati e gli errori di previsione per i diversi metodi proposti e si effettuerà un'analisi di sensibilità per confrontare al meglio gli algoritmi di classificazione considerati a partire da differenti preprocessamenti delle immagini.
L'obiettivo finale è perciò quello di osservare se e quali migliorie la metrica Riemanniana porta all'analisi.

Capitolo 1

Analisi Riemanniana

Molti studi attuali necessitano della raccolta di informazioni geometriche di un oggetto; in particolare ha assunto importanza negli anni ed in varie discipline l'analisi della forma atta al riconoscimento ed alla classificazione di oggetti all'interno di immagini.
La forma è costituita da tutte le informazioni geometriche che permangono quando da un'immagine vengono rimossi tutti gli effetti di rotazione, scala e locazione. Due oggetti risultano avere la stessa forma se venendo scalati, ruotati o traslati hanno esatta equivalenza e sono quindi invarianti sotto similarità euclidea.
L'analisi delle forme più famosa è l'analisi Procustiana, alla quale si fa riferimento in appendice A.1.
Il problema dell'analisi delle forme è che devono essere necessariamente effettuate precise operazioni di traslazione, rotazione e trasformazione di scala che portano ad un errore sistematico dovuto all'approssimazione.
L'analisi geometrica di Riemannian permette l'analisi della forma senza l'utilizzo della trasformazione di similarità evitando le operazioni di standardizzazione dell'immagine.
Per comprendere al meglio tale tipo di analisi è necessario definire alcuni concetti fondamentali riguardanti le metriche Riemanniane; in particolare si andrà a definire i concetti di matrici definite positive, di varietà e di distanza Riemanniana, che verranno successivamente impiegati nelle metodologie proposte.

1.1 Spazio delle Matrici Definite Positive

All'interno dello spazio delle matrici quadrate reali $M(n)$ è possibile definire lo spazio delle matrici simmetriche:

$$S(n) = \{S \in M(n), \quad S^T = S\} \tag{1.1}$$

Lo spazio della matrici definite positive è l'insieme di tutte le matrici simmetriche definite positive tali che:

$$P(n) = \{P \in S(n), u^T P u > 0, \forall u \in \mathbb{R}^n\} \tag{1.2}$$

Una matrice simmetrica definita positiva, *SPD*, è una matrice sempre diagonalizzabile con autovalori strettamente positivi [Barachant et al. 2012].
Per le matrici *SPD*, attraverso la decomposizione spettrale:

$$P = U \operatorname{Diag}(\sigma_1, \ldots, \sigma_n) U^T \tag{1.3}$$

è possibile ottenere la trasformazione esponenziale e logaritmica:

$$\begin{aligned} \exp(P) &= U \operatorname{Diag}(\exp(\sigma_1), \ldots, \exp(\sigma_n)) U^T \\ \log(P) &= U \operatorname{Diag}(\log(\sigma_1), \ldots, \log(\sigma_n)) U^T \end{aligned} \tag{1.4}$$

con $\sigma_1 > \sigma_2 > \cdots > 0$, σ_i autovalore corrispondente all'autovettore i-esimo di P e U matrice degli autovettori di P [Barachant et al. 2012].
Inoltre le matrici SPD godono delle seguenti proprietà [Barachant et al. 2012]:

1. $\forall P \in P(n)$, $det(P) > 0$;
2. $\forall P \in P(n)$, $P^{-1} \in P(n)$;
3. $\forall (P_1, P_2) \in P(n)^2$, $P_1 P_2 \in P(n)$;
4. $\forall P \in P(n)$, $\log(P) \in S(n)$;
5. $\forall P \in S(n)$, $\exp(S) \in P(n)$;
6. $A = P^{\frac{1}{2}}$ matrice simmetrica tale che $P = AA$.

1.2 Varietà

Una varietà è uno spazio topologico localmente simile ad uno spazio euclideo multidimensionale ma che globalmente può essere curvo ed assumere le forme più svariate. Ogni punto della varietà ha un vicino, il quale ha un omeomorfismo con lo spazio $\mathbb{R}^n$ con $n \geq 0$, ovvero esiste una corrispondenza biunivoca e continua tra i due spazi topologici.
Una varietà differenziabile è una varietà nella quale è possibile definire le derivate delle curve. La derivata sulla varietà al punto P giace su un piano T_p, tangente al punto stesso [Tuzel et al. 2007].
Una varietà di Riemannian M si ottiene attraverso la definizione di un prodotto interno variabile con continuità sullo spazio tangente alla varietà stessa [Xie 2013].
Lo spazio delle matrici simmetrice definite positive è differenziabile nella varietà Riemanniana ed ogni punto P della varietà ha derivata che giace nello spazio dei vettori $T_p \in S(n)$ tangente in P [Barachant et al. 2012].
La curva di distanza minima che connette due punti appartenenti alla varietà è chiamata Geodetica; la lunghezza della Geodetica è la distanza Riemanniana.
Per ogni punto $P \in P(n)$ si può definire uno spazio tangente costituito dai vettori tangenti in P. Ogni vettore tangente $S_i \in T_p$ può essere visto come la derivata a $t = 0$ della geodetica $\Gamma_i(t)$ tra P e un punto della mappa esponenziale $P_i = Exp_P(S_i)$. Quindi nello spazio delle matrici simmetriche definite positive, ove esiste il prodotto interno:

$$< S_i, P_i >_P = \mathrm{Tr}(S_i P^{-1} P_i P^{-1}) \tag{1.5}$$

si ottiene la mappa esponenziale associata alla metrica Riemanniana attraverso il calcolo della derivata della Geodetica:

$$\exp_P(S_i) = P^{\frac{1}{2}} \exp(P^{-\frac{1}{2}} S_i P^{-\frac{1}{2}}) P^{\frac{1}{2}} \tag{1.6}$$

con diffeomorfismo, ovvero con proprietà di differenziabilità, invertibilità e con inversa differenziabile e la mappa inversa, logaritmica, definita solo in un intorno di X:

$$\log_P(P_i) = P^{\frac{1}{2}} \log(P^{-\frac{1}{2}} P_i P^{-\frac{1}{2}}) P^{\frac{1}{2}} \tag{1.7}$$

Il calcolo delle precedenti si riduce all'utilizzo della decomposizione spettrale citata in equazione 1.3.

La relazione tra la varietà ed il piano tangente è presente in figura 1.1 dove viene visualizzata anche l'univocità tra la trasformazione esponenziale e logaritmica su un punto P della varietà, [Barachant et al. 2012].

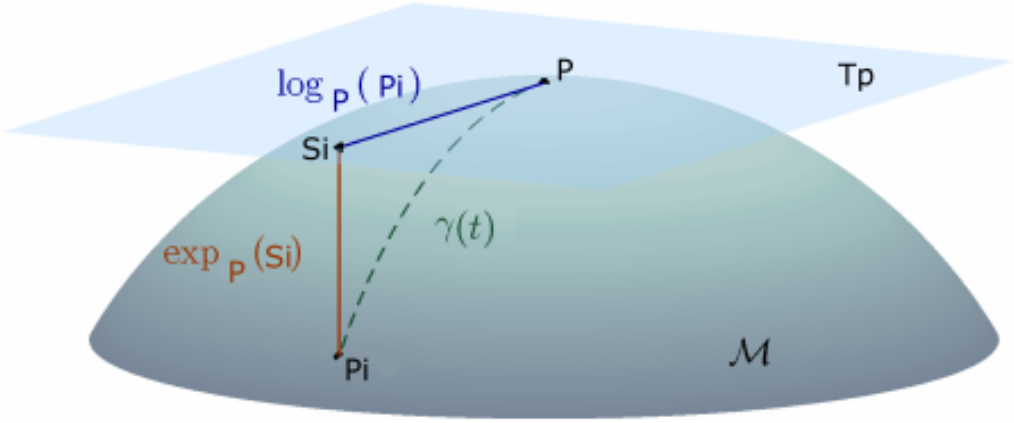


Figura 1.1: T_p spazio tangente nella varietà M al punto P, S_i vettore tangente a P_i e Γ geodetica tra P e P_i.

Considerando ora, come spiegato, due matrici definite positive $S_1, S_2 \in M$ una metrica sulla varietà delle matrici SPD è definita attraverso il prodotto scalare tra due matrici pari a:

$$\langle S_1, S_2 \rangle_P = \mathrm{Tr}(S_1 P^{-1} S_2 P^{-1}) \tag{1.8}$$

La metrica Riemanniana è definita attraverso la norma di Frobenius tale che:

$$\| S \|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{m} |s_{ij}|^2} = \sqrt{\mathrm{Tr}(SS)} \tag{1.9}$$

con n e m rispettivamente numero di righe e di colonne di S. Si ricorda che:

$$\| S \|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{m} |s_{ij}|^2} = \sqrt{\vec{S}^T \vec{S}} \tag{1.10}$$

1.3 Distanza Riemanniana

Considerando lo spazio delle matrici simmetriche definite positive e la mappa Riemanniana calcolabile su di esso si definisce la distanza Riemanniana tra due matrici SPD, P_1 e P_2, come:

$$d_R(P_1, P_2) = \| \log(P_1^{-1}P_2) \|_F = \| \log(P_1^{-\frac{1}{2}}P_2P_1^{-\frac{1}{2}}) \|_F = \Big[\sum_{i=1}^{n} \log^2 \lambda_i\Big]^{\frac{1}{2}} \tag{1.11}$$

con λ_i autovalori della matrice $P_1^{-1}P_2$.

La distanza Riemanniana di matrici simmetriche definite positive gode delle proprietà di:

- Simmetria: $d_R(P_1, P_2) = d_R(P_2, P_1)$;
- Invertibilità: $d_R(P_1^{-1}, P_2^{-1}) = d_R(P_2, P_1)$;
- Invarianza per proiezione: $\forall P_3 \in P(n) \quad t.c. \quad d_R(P_3^T P_1 P_3, P_3^T P_2 P_3) = d_R(P_1, P_2)$.

Dato un campione di $n \geq 1$ matrici definite positive è possibile calcolarne una stima della media tramite:

$$M_R(P_1, \ldots, P_n) = \arg\min_{P \in P(n)} \sum_{i=1}^{n} d_R^2(P, P_i) = \arg\min_{P \in P(n)} \sum_{i=1}^{n} \| \log(P_i^{-1}P) \|_F \tag{1.12}$$

Si ottiene poi una stima della varianza tramite:

$$\Sigma_R = \arg\inf_{\Sigma} \sum_{i=1}^{n} \| \log(P_i^{-\frac{1}{2}}\Sigma P_i^{-\frac{1}{2}}) \|^2 \tag{1.13}$$

Le soluzioni di tali equazioni si ottiengono attraverso procedure di *Gradient Descent*, differenziando la funzione di errore rispetto a P e a Σ rispettivamente e, cercandone il minimo locale [Dryden et al. 2009].

Un'alternativa alla distanza Riemanniana, presente in [Barachant et al. 2012], è pari a:

$$d_R(P, P_i) = \| \log_p(P_i) \|_P = \| \text{upper}(P^{-\frac{1}{2}} \log_p(P_i) P^{-\frac{1}{2}}) \|_2 = \| S_i \|_2 \tag{1.14}$$

dove upper$(\cdot)$ è l'operatore che considera solamente la parte triangolare superiore della matrice simmetrica vettorizzata a cui applica pesi unitari sugli elementi della diagonale e pesi pari a $\sqrt{2}$ sugli elementi fuori dalla diagonale principale.
Il vettore m dimensionale dello spazio tangente normalizzato S_i fa sì che:

$$\forall i, j \qquad d_R(P_i, P_j) \approx \| S_i - S_j \|_2 \tag{1.15}$$

Capitolo 2

Metodo Kernel

L'idea alla base di ogni metodo *Kernel* è di introdurre non linearità alla funzione di decisione, che costituisce il confine di scelta, nelle procedure di classificazione. La non linearità della frontiera fa sì che ci sia un adattamento migliore ai dati con confini decisionali più versatili.
Per comprendere al meglio il fenomeno, anche in questo caso, è necessario definire alcuni concetti fondamentali.
Uno spazio vettoriale è uno spazio con prodotto interno se esiste una funzione simmetrica bilineare per la quale vale che:

$$\langle X, X \rangle \geq 0 \tag{2.1}$$

Uno spazio vettoriale in cui esiste una distanza è uno spazio metrico. Uno spazio metrico completo, ovvero uno spazio in cui tutte le successioni fondamentali di Cauchy[1] di X convergono ad un elemento X nello spazio, in cui la distanza è indotta dal prodotto scalare, è uno spazio di Hilbert, H.
Uno spazio di Hilbert separabile possiede basi hilbertiane, ovvero basi ortonormali numerabili che permettono di rappresentare ogni elemento dello spazio in modo unico come somma delle basi hilbertiane moltiplicate per i coefficienti di Fourier di ogni elemento. Negli spazi di Hilbert separabili è possibile risolvere problemi di approssimazione mediante polinomi, polinomi trigonometrici o altre funzioni particolari.

[1] Una successione $\{X_n\}$ è una successione di Cauchy se $\forall \epsilon > 0$, $\exists$ un numero $\mathrm{N}(\epsilon) > 0$ t.c.: $d(X_n, X_m) < \epsilon \qquad \forall \mathrm{n}, \mathrm{m} > \mathrm{N}(\epsilon)$.

Considerando n variabili dipendenti con n esplicative, cercando una funzione f con una certa regolarità, è necessario ricondursi a un problema del tipo:

$$\min_{f \in H} F_\lambda(f) = \lambda \parallel f \parallel_H^2 + \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 \tag{2.2}$$

dove $\| \cdot \|$ è la norma in un opportuno spazio di funzioni con regolarità desiderata. Per ottenere esistenza e unicità della soluzione è necessario considerare un opportuno spazio di Hilbert con norma $\| \cdot \|$ e con proprietà di continuità dei funzionali di valutazione in ogni punto, ovvero:

$$|F_{x_i}(f)| \leq M \parallel f \parallel_H \qquad \forall \quad i = 1, \ldots, n \tag{2.3}$$

con funzionale di valutazione $F_{x_i}(f) = f(x_i)$ continuo rispetto alla norma.
Uno spazio di Hilbert i cui elementi sono funzioni per cui valga la proprietà sopra citata si dice *Reproducing Kernel Hilbert Space*, RKHS [Wahba et al. 1999].
In uno spazio RKHS si avrà che:

$$f(x) = F_x(f) = \langle K(x, \cdot), f(\cdot) \rangle_H \qquad \forall f \in H \tag{2.4}$$

per una certa funzione *kernel*. Il nucleo $K(x, y)$, simmetrico e definito positivo, è detto nucleo riproducente perché determina univocamente lo spazio RKHS tramite la proprietà:

$$K(x, y) = \langle K(x, \cdot), K(y, \cdot) \rangle_H \tag{2.5}$$

In poche parole, grazie al nucleo riproducente, un insieme di dati non separabile linearmente nello spazio dei numeri reali diventa separabile nello spazio di Hilbert di grandi dimensioni, come si può vedere in figura 2.1.

2.1 Densità kernel e classificazione

Dato un campione casuale semplice di $x_1, \ldots, x_N$ con densità $f_x(x)$ si può stimare f_X al punto x_0 con una stima locale della funzione data da:

$$\hat{f}_X(x_0) = \frac{1}{N\lambda} \sum_{i=1}^{N} K_\lambda(x_0, x_i) \tag{2.6}$$

con λ larghezza di banda per l'identificazione dei vicini più prossimi e funzione kernel che dipende da tale parametro, così da imporre dei pesi che diminuiscono

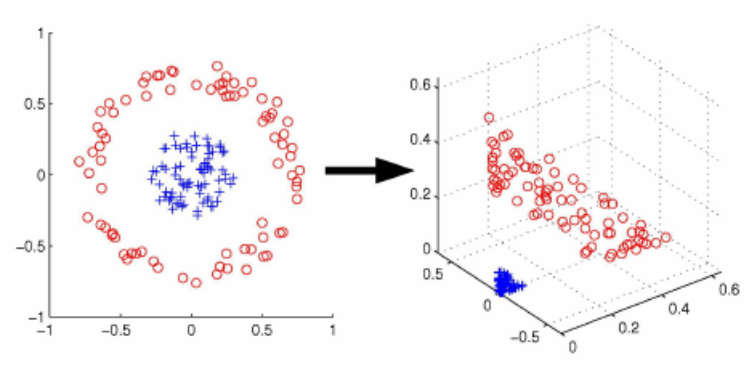


Figura 2.1: Esempio di trasformazione da spazio a 2 dimensioni a spazio a 3 dimensioni attraverso il *kernel* riproducente $K : \mathbb{R}^2 \to \mathbb{R}^3$ costituendo un RKHS separabile.

all'aumentare della distanza da x_0.
In questo caso è possibile utilizzare varie tipologie di funzioni *kernel*; la scelta più popolare è il *kernel* Gaussiano.
Si suppone ora di avere J classi con stima di densità non parametrica pari a $\hat{f}_j(X)$ e con stima della probabilità a priori $\hat{\pi}_j$ pari solitamente alla proporzione nel campione.
La stima della probabilità di appartenenza ad una classe dato un punto x_0 è:

$$\hat{Pr}(G = j|X = x_0) = \frac{\hat{\pi}_j \hat{f}_j(x_0)}{\sum_{k=1}^{J} \hat{\pi}_k \hat{f}_k(x_0)}. \tag{2.7}$$

Se lo scopo finale dell'analisi è la classificazione, non è necessaria l'identificazione separata di ogni densità per ogni classe, ma basta la stima della probabilità a posteriori nelle vicinanze del confine decisionale [Friedman et al. 2001].
Se ad esempio $J = 2$ allora la classificazione si ottiene tramite il valore della posteriori pari a:

$$x|Pr(G = 1|X = x) = \frac{1}{2} \tag{2.8}$$

2.2 Metodo kernel su varietà

La mappa esponenziale e quella logaritmica (citate nella sezione 1.2) vengono utilizzate per mappare i punti dalla varietà allo spazio tangente e viceversa. Tali operazioni richiedono il calcolo iterativo delle mappe con costo computazionale

particolarmente elevato e approssimano la vera distanza sulla varietà con la distanza euclidea con una conseguente riduzione di accuratezza.
Per risolvere tali problemi si può utilizzare il metodo *kernel* che incorpora la varietà in uno spazio di Hilbert (RKHS sopra citato).
La funzione di trasformazione che permette il passaggio ad uno spazio di Hilbert si basa su una funzione *kernel* definita positiva. Il *kernel* definito positivo più famoso è il *kernel* Gaussiano:

$$K_g(x_i, x_j) := \exp\left(\frac{||x_i - x_j||^2}{2\sigma^2}\right) \tag{2.9}$$

che utilizza la distanza euclidea tra due punti.
Risulta di naturale estensione sostituire la distanza Euclidea con la distanza Geodetica calcolata sulla varietà.
La matrice *kernel* costituita da tale sostituzione non risulta essere sempre definita positiva. In ogni caso la distanza Geodetica risulta essere la misura più naturale di similarità tra due matrici. Quindi, lo spazio costruito dal *kernel* definito positivo trasforma la varietà non lineare in uno spazio di Hilbert lineare e preserva la distribuzione dei dati originali [Jayasumana et al. 2013].

2.2.1 Kernel definito positivo

Data M varietà, (M, d) spazio metrico e $K : (M \times M) \to \mathbb{R}$ con:

$$K(x_i, x_j) := \exp\left(-\frac{d^2(x_i, x_j)}{2\sigma^2}\right) \tag{2.10}$$

allora K è un *kernel* definito positivo per tutti i $\sigma > 0$ se e solo se esiste uno spazio V di prodotti interni e una funzione $\Psi : M \to V$ per cui valga la metrica:

$$d(x_i, x_j) = \| \Psi(x_i) - \Psi(x_j) \|_V \tag{2.11}$$

ed il quadrato di essa sia definito negativo.
Considerando ora il caso di matrici simmetriche definite positive, la vera distanza Geodetica è la distanza Log-Euclidea; risulta comunque possibile identificare ulteriori metriche riportate in tabella 2.1. Tuttavia la distanza Log-Euclidea è quella che meglio identifica la distanza Riemanniana nella varietà; infatti, tale metrica

utilizza operatori algebrici e matriciali che evitano le approssimazioni dovute all'utilizzo delle mappe di trasformazione Riemanniane presentate in 1.2.
Date $P_i, P_j \in SPD$ si calcola la Geodetica sotto struttura Log-Euclidea come:

$$\gamma(t) = \exp\{(1-t)\log(P_i) + t\log(P_j)\} \qquad t \in [0,1] \tag{2.12}$$

La distanza geodetica tra P_i e P_j è pari a:

$$d_g(P_i, P_j) = \| \log(P_i) - \log(P_j) \|_F \tag{2.13}$$

dove $\| \cdot \|_F$ è la norma di Frobenius indotta dal prodotto interno $< \cdot, \cdot >_F$ della matrice di Frobenius e $\log(\cdot)$ è il logaritmo matriciale di una matrice simmetrica definita positiva. [2]

Dato $K_R : (P \times P) \to \mathbb{R}$ tale che

$$K_R(P_i, P_j) := \exp\left(- \frac{d_g^2(P_i, P_j)}{2\sigma^2} \right) \tag{2.14}$$

con $d_g(P_i, P_j) = \| \log(P_i) - \log(P_j) \|_F$ allora K_R è definito positivo per ogni $\sigma \in \mathbb{R}$.
Sono disponibili ulteriori metriche che definiscono *kernel* definiti positivi per ogni σ o solo per alcuni valori di essi, come è possibile vedere in tabella 2.1.
Imporre costrizioni al valore di σ non è mai opportuno infatti molti modelli di classificazione richiedono che la funzione sia continua e definita per tutti i valori del parametro [Jayasumana et al. 2013].

[2] Nel caso di matrici quadrate di dimensione 2:
$\| A \|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} = \sqrt{trace(A^T A)} = \sqrt{\sum_{i=1}^{min\{m,n\}} \sigma_i^2}$
con σ_i valori singolari di A e $< A, A >_F = a_{11} + a_{12} + a_{21} + a_{22}$.

Tabella 2.1: Possibili Kernel Gaussiani

METRICHE DISPONIBILI	FORMULA	DISTANZA GEODETICA	KERNEL DEFINITO POSITIVO
Log-Euclidea	$\| \log(P_1) - \log(P_2) \|_F$	X	X
Affine-Invariante	$\| \log(P_1^{-\frac{1}{2}} P_2 S_1^{-\frac{1}{2}}) \|_F$	X	
Cholesky	$\| chol(P_1) - chol(P_2) \|_F$		X
Power-Euclidean	$\frac{1}{\lambda} \| P_1^{\lambda} - P_2^{\lambda} \|_F$		X
Root-Stein Divergence	$[\log det(\frac{1}{2}P_1 + \frac{1}{2}P_2) - \frac{1}{2} \log det(P_1 P_2)]^{\frac{1}{2}}$		solo per alcuni σ

2.3 Algoritmi basati sul kernel

Si considerano una funzione *kernel* $K(\cdot, \cdot)$, uno spazio H di tipo *RKHS*, generato da K e $\Phi(X)$, vettore di caratteristiche di H, dove viene mappato $X \in SPD$ tramite il *kernel* K.
Come già precedentemente spiegato, grazie all'utilizzo del *kernel*, $\Phi(X)$ non ha bisogno di calcolo esplicito perché i dati vengono considerati nell'analisi solo tramite il nucleo.
Vari metodi possono essere implementati con l'inserimento di un *kernel*, utilizzato come nucleo riproducente all'interno dello spazio, se ne elencano alcuni:

- *Support Vector Machine*;
- *Multiple Kernel Learning*;
- *Discriminative Nearest Neighbor*;
- *Analisi Componenti Principali*;
- *K-Means*.

2.3.1 Support Vector Machine

Data $y_i \in (-1, 1)$, una certa etichetta che identifica la classe di appartenenza nel caso di classificazione bidimensionale, la *Support Vector Machine* (SVM) cerca l'iperpiano in H che separa, con margine massimo, il vettore di caratteristiche tra le due classi. La classe viene determinata in base alla posizione di $\Phi(x)$ sull'iperpiano separato.
Un modello di regressione o di classificazione può essere riformulato in forma di

rappresentazione duale nella quale il *kernel* sorge spontaneamente. La rappresentazione duale è molto importante nel caso della Support Vector Machine, perché a partire da essa è possibile calcolare il problema di massimo e minimo per l'identificazione della soglia [Anzai 2012].
Considerando ad esempio un modello di regressione i cui parametri sono determinati tramite la minimizzazione della somma del quadrato degli errori regolarizzata, del tipo:

$$\min_{f \in H} \left[\sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda J(f) \right] \tag{2.15}$$

con $L(y_i, f(x_i))$ funzione di perdita e $J(f)$ funzione di penalità.
La funzione di penalità può essere definita in termini di funzione *kernel*. Utilizzando il *kernel* con rappresentazione in autofunzione:

$$K(x, x_1) = \sum_{i=1}^{\infty} \gamma_i \phi_i(x) \phi_i(x_1) \tag{2.16}$$

dove $\gamma_i \geq 0$ e $\sum_{i=1}^{\infty} \gamma_i^2 < \infty$, si ottiene per gli elementi dello spazio H un'equazione del tipo:

$$f(x) = \sum_{i=1}^{\infty} c_i \phi_i(x) \tag{2.17}$$

con vincolo $\| f \|_H^2 := \sum_{i=1}^{\infty} c_i^2 / \gamma_i < \infty$.
La funzione di penalità è allora $J(f) = \| f \|_H^2$ ed implica che all'aumentare del valore degli autovalori nella rappresentazione in autofunzione (2.16) diminuisce la penalizzazione.
La soluzione al problema di minimo:

$$\min_{f \in H} \left[\sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda ||f||_H^2 \right] \tag{2.18}$$

è data da $f(x) = \sum_{i=1}^{N} \alpha_i K(x, x_i)$ di dimensione finita.
La funzione di basi $h_i(x) = K(x, x_i)$ è la valutazione di x_i in H.
Se $f \in H$ vale che $f(x_i) = \langle K(\cdot, x_i), f \rangle_H$, perciò:

$$J(f) = \sum_{i=1}^{N} \sum_{j=1}^{N} K(x_i, x_j) \alpha_i \alpha_j \qquad con \qquad f(x) = \sum_{i=1}^{N} \alpha_i K(x, x_i) \tag{2.19}$$

Le proprietà del *kernel* fanno sì che un problema di infinite dimensioni si riduca ad un problema di ottimizzazione di finite dimensioni risolvibile attraverso algoritmi

numerici.

Utilizzando ora il moltiplicatore di Lagrange, è possibile considerare $f(x)$ come:

$$f(x) = h(x)^T\beta + \beta_0 = \sum_{i=1}^{N} \alpha_i y_i \langle h(x), h(x_i)\rangle + \beta_0 \tag{2.20}$$

dato il vettore α, β_0 è determinato grazie al vincolo $y_i f(x_i) = 1$ per ogni x_i per cui vale $0 < \alpha_i < \gamma$.
La funzione di basi $h(x)$ viene coinvolta solo tramite il prodotto interno, per tale motivo non è necessario specificare tutte le basi ma basta conoscere la funzione *Kernel*:

$$K(x, x_1) = \langle h(x), h(x_1)\rangle \tag{2.21}$$

che calcola i prodotti interni nello spazio trasformato.
Data la trasformazione in auto-funzioni della funzione *kernel* e la funzione di perdita della SVM pari a:

$$L(Y_i, f(x_i)) = [1 - y_i f(x_i)]_+ \tag{2.22}$$

è facile ottenere una soluzione di dimensione finita al problema di minimo della (2.15) della forma:

$$f(x) = \beta_0 + \sum_{i=1}^{N} \alpha_i K(x, x_i) \tag{2.23}$$

attraverso un ottimizzatore numerico con il quale stimare i parametri β_0 e α_i [Friedman et al. 2001]. [3]
I *kernel* più conosciuti sono:

- d-polinomiale: $K(x, x') = (1 + \langle x, x'\rangle)^d$;
- a base radiale: $K(x, x') = exp(-\gamma||x - x'||^2)$;
- rete neurale: $K(x, x') = tanh(\kappa_1\langle x, x'\rangle + \kappa_2)$.

Per l'estensione al caso multivariato è necessario considerare una classificazione binaria per ogni classe presente nel dataset.

[3] Nel caso bidimensionale, $y_i = \{-1, +1\}$ e la soluzione al problema di minimo è:
$f(x) = \begin{cases} +1, & se \quad Pr(Y = +1|X) \geq \frac{1}{2} \\ -1, & altrimenti \end{cases}$

2.3.2 Multiple Kernel Learning

L'idea che sta alla base del *Multiple Kernel Learning* è di calcolare diversi *kernel* su diversi insiemi di esplicative. Con questo metodo è possibile considerare diverse tipologie di rappresentazione e trasformazione delle osservazioni ottenendone una combinazione ottimale attraverso l'uso di una diversa base *kernel* per ognuna di esse.
Dati m insiemi di esplicative $X_i \in X$, di $y_i \in Y$ classi e di funzioni generatrici ${g_j}_1^N$ tali che $g_j : X \to SPD$ si costituisce una funzione di classificazione $f : X \to Y$.
La funzione f si costruisce tramite la combinazione ottimale dei differenti descrittori generati dalle funzioni generatrici g_j.
$K^{(j)}$ è la matrice *kernel* generata dalla funzione K e dalle funzioni generatrici:

$$k_{pq}^{(j)} = k(g_j(x_p), g_j(x_q)) \tag{2.24}$$

La convergenza dell'algoritmo è garantita solo in presenza di *kernel* definiti positivi, perciò viene considerato:

$$K^* = \sum_{j=1}^{N} \lambda_j K^{(j)} \tag{2.25}$$

con $\lambda_j \leq 0$ per $j = 1, \ldots, N$ che comporta la positività di K^*. I pesi λ_j sono i parametri di penalizzazione e devono poi essere calcolati tramite una procedura di ottimizzazione. Inoltre è possibile considerare i pesi come valore di appartenenza ad una classe e calcolarli tramite *clustering*.
L'estensione al caso multiclasse si basa sulla gestione di un metodo a due passi; il primo passo in cui si ottimizza la migliore combinazione dei vari *kernel* ed il secondo volto alla classificazione con *Support Vector Machine* in base alla combinazione e ai pesi stimati al passo precedente [Bucak et al. 2014].

2.3.3 Discriminative Nearest Neighbor

L'idea che sta alla base del *Discriminative Nearest Neighbor* è quella di considerare solo i vicini di un punto e, sull'insieme così ottenuto, effettuare un'analisi tramite *Support Vectore Machine* che riesca a preservare la distanza sulla varietà. Tale procedura elimina il problema di poca adattabilità del metodo dei vicini più vicini e permane efficiente nei casi in cui, in presenza di un numero elevato di classi, la *Support Vector Machine* risulta essere poco efficiente e computazionalmente

non trattabile.
Il metodo proposto è composto da 5 passi:

1. si trova l'insieme dei T vicini attraverso una funzione di distanza opportuna;
2. si calcola una funzione di distanza più accurata all'interno degli insiemi di vicini ottenuti nel primo passo per considerare gli insiemi dei vicini più prossimi;
3. per ogni nuova osservazione si calcola la distanza con gli insiemi dei vicini più prossimi ottenuti al secondo passo;
4. si converte la matrice di distanze calcolata al punto 3 in una matrice *kernel* attraverso l'equazione:

$$\begin{aligned} K(x,y) &= \langle x,y\rangle \\ &= \frac{1}{2}(\langle x,x\rangle + \langle y,y\rangle - \langle x-y,x-y\rangle) \\ &= \frac{1}{2}(d(x,0) + d(y,0) - d(x,y)) \end{aligned} \tag{2.26}$$

   con d distanza.
5. si applica una *Support Vector Machine* multiclasse sulla matrice *Kernel* e si assegna alla nuova osservazione una classe in base al risultato della classificazione.

Nel caso in cui la cardinalità dell'insieme dei vicini più vicini è piccola, la procedura risulta essere un semplice classificatore *Nearest Neighbor*. Quando invece il numero degli insieme dei vicini più prossimi è pari al numero delle osservazioni del campione, allora la procedura è equivalente ad una *Support Vector Machine* standard [Zhang et al. 2006].

2.3.4 Altri metodi

L'analisi delle componenti principali con *kernel* è un metodo di riduzione della dimensionalità non lineare. Questa analisi estrae dai dati iniziali un numero di dimensioni maggiore rispetto alla dimensione dello spazio dato in input.
Tutti i punti $X_i \in SPD$ vengono riprodotti tramite lo spazio delle caratteristiche

in H, formando l'insieme delle caratteristiche trasformate $\Phi(X_i)$. Per calcolare la matrice di covarianza di $\Phi(X_i)$, è necessario valutare la matrice *kernel* sui dati originali. Una rappresentazione dei dati è ottenuta calcolando gli autovettori della matrice *kernel* così generata.
Questo tipo di rappresentazione si avvicina molto ad una rappresentazione euclidea dei dati originali sulla varietà [Jayasumana et al. 2013].
Il metodo *Kernel K-Means*, invece, è un metodo di apprendimento non supervisionato volto a ridurre la dimensione del campione. Attraverso il kernel vengono riprodotti i punti su uno spazio di Hilbert e si utilizza un algoritmo *K-Means* per raggruppare lo spazio delle caratteristiche e per conservare una rappresentazione ottimale dei dati.
Una volta considerato lo spazio di Hilbert tramite kernel a partire dalla varietà, viene utilizzato un algoritmo di *clustering* non supervisionato il quale, dato un insieme di X_i dello spazio RKHS e un numero di *cluster*, assegna ogni X_i al gruppo con cui ha distanza minima dal centroide. Tale procedimento viene ripetuto iterativamente fino a quando la somma dei quadrati delle distanze di ogni $\Phi(X_i)$ è minore di un preciso valore specificato a priori [Jayasumana et al. 2013].

Capitolo 3

Metodologia proposta

Dopo aver approfondito i concetti di metrica Riemanniana e di *kernel* riproducente è di interesse coniugare le varie nozioni e produrre algoritmi per raggiungere gli obiettivi iniziali.
L'analisi Riemanniana richiede l'utilizzo di matrici simmetriche definite positive. Nel caso in oggetto ogni matrice definisce un'immagine ed ogni cella della matrice si riferisce ad un pixel.
In letteratura viene spesso utilizzata una trasformazione della matrice iniziale che oltre a considerare la matrice di intensità di ogni immagine valuta anche il gradiente per ogni direzione costituendo un descrittore di dati del tipo:

$$\mathrm{H} = [\mathrm{I}, \quad \mathrm{I}_x, \quad \mathrm{I}_y, \quad \mathrm{I}_{xx}, \quad \mathrm{I}_{xy}, \quad \mathrm{I}_{yy}] \tag{3.1}$$

dove I è la matrice delle intensità dell'immagine e I_d è il derivata in d calcolata tramite l'operatore di Sobel [Kanopoulos et al. 1988].
Vettorizzando poi il descrittore H e calcolandone la matrice di covarianza o la matrice di correlazione, si ottiene per ogni immagine una matrice 6×6 definita positiva a rango pieno, utilizzabile direttamente nella metrica Riemanniana. Quest'ultima tipologia di pre-processamento, oltre ad avere un grande onere computazionale, comporta un'indesiderabile perdita di informazione dovuta all'utilizzo di una misura di sintesi del descrittore, altrimenti non utilizzabile a causa della grandi dimensioni; si vuole evitare tale procedimento con lo scopo di considerare direttamente la matrice di intensità dell'immagine iniziale.

L'idea è allora quella di partire dalla matrice di intensità X_i corrispondente all'immagine i ed ottenere una matrice simmetrica ad esempio con:

1. $S_i(X_i) = X_i X_i^T$;
2. $S_i(X_i) = COV(X_i) = E[(X_i - E[X_i])(X_i - E[X_i])^T] = \Sigma_i$;
3. $S_i(X_i) = Corr(X_i) = (\text{Diag}(\Sigma_i))^{-\frac{1}{2}} \Sigma_i (\text{Diag}(\Sigma_i))^{-\frac{1}{2}}$;
4. $S_i(X_i) = \left(\frac{(X_i - \mu_{x_i})}{sd(X_i)}\right)\left(\frac{(X_i - \mu_{x_i})}{sd(X_i)}\right)^T$;

o moltiplicando una qualsiasi trasformazione della matrice X_i per la sua trasposta.

Per ottenere matrici definite positive è necessario avere tutte le colonne linearmente indipendenti all'interno della matrice X_i. Effettuando il centramento delle matrici di intensità si avrà necessariamente un autovalore nullo, inoltre i dati a disposizione sono immagini di scarsa qualità che hanno subito operazioni di ridimensionamento: a causa di ciò le matrici considerate sono composte da colonne linearmente dipendenti e sono semi-definite positive.

Per ovviare a tale problema vi sono vari approcci: il primo tra questi è quello di non considerare le colonne linearmente dipendenti delle matrici, e ridurre la dimensione delle matrici iniziali. Utilizzando questo approccio si ottengono matrici di dimensioni differenti e ciò porta alla creazione di osservazioni con diverso numero di variabili.

Una possibile soluzione è quella di utilizzare un metodo di regoralizzazione volto ad effettuare il completamento del rango della matrice iniziale. [Zhang et al. 2015]. In letteratura esistono diversi metodi di regolarizzazione; nel caso in oggetto l'approccio considerato è quello di sommare una matrice diagonale alle matrici simmetriche calcolate:

$$P_i(X_i) = S_i(X_i) + \lambda I \tag{3.2}$$

In alternativa è possibile sommare matrici derivanti dai dati o da trasformazioni di essi in modo tale da completare le matrici in oggetto con una quantità che dipende dalle immagini iniziali [Ledoit e Wolf 2004].

Attraverso prove preliminari si è deciso di considerare nei metodi proposti il preprocessamento tramite la matrice di covarianza (in lista 3.2) e tramite la standardizzazione globale della matrice iniziale (in lista 3.4). Nel caso in cui si considera

come trasformazione di partenza la covarianza, la matrice diagonale aggiunta ha diverso peso in base alle diverse immagini; considerando invece una misura standardizzata della matrice iniziale si porta stessa penalizzazione in ogni osservazione. Ricapitolando, gli algoritmi di classificazione proposti per utilizzare matrici simmetriche semi-definite positive, come primo passo, effettuano il pre-processamento delle immagini attraverso una delle misure sopra citate; come secondo passo, per completare il rango delle matrici aggiungono una matrice diagonale moltiplicata per un parametro di regolarizzazione.
Vista però la non invarianza rispetto a traslazioni della distanza Riemanniana non è possibile sommare una matrice diagonale ad ogni matrice pre-processata senza aggiungere distorsione; è necessario perciò individuare, tramite convalida incrociata, il parametro di penalizzazione che ottiene risultati migliori in termini di errore e che risulta essere un buon compromesso tra varianza e distorsione.
Per costruire il *dataset* ed il descrittore sopra citato, per il pre-processamento dei dati tramite una delle misure sopra elencate e per effettuare regolarizzazione, sono stati implementati algoritmi attraverso l'ambiente R [R Core Team 2012], utilizzando alcune delle funzioni del pacchetto base.

3.1 Minima Distanza dalla Media Riemanniana

Un primo approccio è quello di costituire un semplice e pratico algoritmo di classificazione attraverso il quale è possibile associare ogni immagine alla rispettiva categoria utilizzando la distanza Riemanniana come misura di similarità, chiamato Minima Distanza dalla Media Riemanniana (MDMR).
L'algoritmo che viene proposto, una volta calcolate matrici definite positive a rango pieno calcola la media delle matrici pre-processate per ogni categoria tramite una procedura di *Gradient Descent*, ovvero tramite un algoritmo di ottimizzazione iterativo di primo ordine della funzione:

$$M_R(P_1, \ldots, P_n) = \arg\min_{P \in P(n)} \sum_{i=1}^{n} d_R^2(P, P_i) \tag{3.3}$$

utilizzato per ottenere il minimo locale di P. Tale algoritmo afferma che per una funzione data $d_R^2(P, P_i)$, la direzione di massima discesa di una matrice P assegnata, è determinata dall'opposto del suo gradiente in P.

Il metodo del gradiente parte da un valore iniziale scelto arbitrariamente per la funzione e procede iterativamente aggiornando la soluzione come somma del passo precedente e direzione di discesa del passo attuale moltiplicata per la lunghezza del cammino di discesa.
Utilizzando come cammino di discesa il gradiente, viene garantita la convergenza ad un minimo locale della funzione; se la funzione è convessa il minimo locale coincide con il minimo globale e la procedura di *Gradient Descent* converge alla soluzione globale [Friedman et al. 2001].
Le previsioni risultano poi facili da ottenere, infatti computando la distanza Riemanniana tra la media di ogni categoria M_i e una nuova matrice P_j:

$$d_R(P_j, M_i) = ||\log(P_j^{-1} M_i)||_F \tag{3.4}$$

si associa ogni nuova immagine P_j alla classe con la quale ha distanza minore. Gli errori vanno poi valutati confrontando le previsioni con i valori osservati.
Per comprendere al meglio l'algoritmo se ne riporta una versione in pseudo-codice in 3.1; si noti che il parametro λ è definito tramite convalida incrociata.
Per costituire l'algoritmo di classificazione spiegato si è utilizzato l'ambiente R [R Core Team 2012], con l'ausilio della funzione `estcov` del pacchetto `Shapes` per la definizione della media Riemanniana di ogni classe.

Listing 3.1: "Distanza minima dalla media con metrica Riemanniana"

```
INPUT:
- X[1],...,X[n] immagini insieme di stima
- X1[1],...,X1[m] immagini insieme di verifica
- nclas numero delle classi di immagini
OUTPUT:
- y[1],...y[m] previsioni
- errore di classificazione
PASSI:
- calcolo lambda parametro di penalizzazione
  tramite convalida incrociata su insieme di stima:
  - for (h in 1:n) C[h]=CovarianzaPenalizzata(lambda,X[h])
  - for (i in 1:nclas)
                    media[i]=RiemannianMean(C[1],...C[n])
- for (k in 1:m) C1[k]=CovarianzaPenalizzata(lambda,X1[k])
- for (j in 1:m)
   {
     for (l in 1:nclas)
           dist[l]=RiemannianDistance(media[l],C1[l])
     y[j]=min(dist)
   }
- calcolo misura di errore su insieme di verifica
```

3.2 Kernel Support Vector Machine Riemanniana

Un secondo approccio è quello di considerare un *kernel* Riemanniano nei metodi presenti in letteratura; nel caso specifico si vuole modificare una *Support Vector Machine* e ottenere una *Support Vector Machine* con *kernel* Riemanniano (SVMR). L'idea è di partire dalla funzione della distanza e calcolare il nucleo riproducente Riemanniano della SVM per ottenere il confine decisionale dei dati considerati.
Le variabili esplicative vengono date in *input* solo attraverso la matrice *kernel* calcolata ed il confine decisionale dipenderà da quest'ultima e dalle variabili dipendenti che identificano la classe dell'insieme di stima. Ottenuto il confine deci-

sionale, vengono classificate le immagini dell'insieme di verifica solo in base alla loro posizione rispetto all'iperpiano separatore costituito dai dati dell'insieme di stima.
Grazie all'uso del *kernel* riproducente non è necessario trasformare tutti i punti dello spazio ma basta calcolare la matrice *kernel* corrispondente; infatti anche all'aumentare della dimensione della matrice iniziale la *Support Vector Machine* considera solamente un numero ridotto di unità tramite le combinazioni lineari dati dal prodotto interno presente nel nucleo senza la necessità di onerosi calcoli computazionali e senza problemi di memoria.
Riassumendo, sul *dataset* di stima verrà calcolato il confine di decisione con l'utilizzo di un numero ridotto di osservazioni (punti di supporto) mentre sul *dataset* di verifica verrà effettuata la classificazione e il calcolo dell'errore tramite l'utilizzo del confine decisionale valutato; è importante ricordare che la *Support Vector Machine* utilizza i dati dell'insieme di stima anche per la fase di classificazione sull'insieme di verifica [Anzai 2012].
Per una definizione più formale del metodo proposto si precede come segue.
Dati $y_1, \ldots, y_n$ etichette delle classi e $x_1, \ldots, x_n$ esplicative del *dataset* di stima, si calcola la distanza Riemanniana ed il nucleo relativo per ogni osservazione tramite:

$$K(X_i, X_j) = \exp\left(- \frac{d_R^2(X_i, X_j)}{2\sigma} \right) = \exp\left(- \frac{\| \, log(X_i) - log(X_j) \, \|^2}{2\sigma} \right) \tag{3.5}$$

Si ricorda che è necessario considerare ogni classe come un problema bivariato ed associare ogni nuova osservazione in base alla maggiore propensione di appartenenza alle classi.
Nel caso bivariato si ha che:

$$f(X) = w^T \phi(X) + b \tag{3.6}$$

e la classificazione viene fatta in base al segno di $f(X)$. Imponendo $f(X) = 0$ si ottiene il confine di decisione.
La distanza di un punto X_i dalla superficie di decisione è:

$$\frac{y_i f(Xi)}{\| \, w \, \|} = \frac{y_i (w^T \phi(X_i) + b)}{\| \, w \, \|} \tag{3.7}$$

che compone un problema di ottimizzazione del tipo:

$$arg \max_{w,b} \left\{ \frac{1}{\| \, w \, \|} \min_n (y_i (w^T \phi(X_i) + b)) \right\} \tag{3.8}$$

senza soluzione esplicita.
Per i punti vicini alla superficie di decisione vale che:

$$y_i(w^T\phi(X_i) + b) = 1 \tag{3.9}$$

quindi il problema diventa:

$$arg\max_{w,b} \left\{ \frac{1}{2} \parallel w \parallel^2 \right\} \tag{3.10}$$

Il parametro di intercetta b è scomparso dal problema considerato, ma un cambiamento in b va a modificare il valore di w, ed un cambiamento in w può essere compensato da b senza mostrare alcuna differenza. È necessario quindi introdurre e considerare il moltiplicatore di Lagrange:

$$L(w, b, a) = \frac{1}{2} \parallel w \parallel^2 - \sum_{i=1}^{n} a_i\{y_i(w^T\phi(X_i) + b) - 1\} \tag{3.11}$$

trasformato di segno per effettuare minimizzazione rispetto w e b e massimizzazione rispetto ad a, con $a = (a_1, \ldots, a_n)$ ed n numero di osservazioni dell'insieme di stima. Calcolando la derivata del moltiplicatore di Lagrange per w e b e ponendolo uguali a zero si ottengono le condizioni:

$$\begin{aligned} w &= \sum_{i=1}^{n} a_i y_i \phi(X_i) \\ 0 &= \sum_{i=1}^{n} a_i y_i \end{aligned} \tag{3.12}$$

Sostituendo tali condizioni nell'equazione 3.11 per eliminare w e b dal moltiplicatore si ottiene la rappresentazione duale del tipo:

$$\begin{aligned} \tilde{L}(a) = \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j y_i y_j K(X_i, X_j) \\ s.v. \qquad a_i \geq 0 \\ \sum_{i=1}^{n} a_i y_i = 0 \end{aligned} \tag{3.13}$$

da massimizzare per ottenere la stima di a.
Per effettuare la massimizzazione ed ottenere una stima del vettore a, una volta considerato $\tilde{L}(a)$, è necessario calcolare la matrice di Gram pari a:

$$G_{i,j} = y_i y_j K(X_i, X_j) \tag{3.14}$$

e risolvere il problema di programmazione quadratica dalla forma:

$$\begin{aligned} \max_{0 \leq a \leq C} \; & a\mathbf{1} - \frac{1}{2} a^T G a \\ con \quad & ay = 0 \\ & G = ZZ^T \\ & w = Z^T a \end{aligned} \tag{3.15}$$

dove $\mathbf{1}$ è un vettore di 1 di dimensione n.
Solo alcuni a saranno non nulli; i punti X_i che corrispondono agli a non nulli sono i punti di supporto considerati nell'analisi.
Attraverso complementarità o dualità è poi possibile calcolare una stima dell'intercetta b; una soluzione stabile è data dalla media:

$$\hat{b} = \frac{1}{n} \sum_{i=1}^{n} \Big(y_i - \sum_{j=1}^{n} a_j y_j K(X_i, X_j) \Big) \tag{3.16}$$

dove n è il numero totale dei punti di supporto.
Per ogni nuova osservazione dell'insieme di verifica X^*, è quindi possibile ottenere il valore di $f(X^*)$ utilizzando le stime di a e b, tramite:

$$\hat{f}(X^*) = \hat{b} + \sum_{i=1}^{n} \hat{a}_i y_i K(X^*, X_i) \tag{3.17}$$

Nel caso bivariato basta valutare il segno di $f(X^*)$ per classificare l'osservazione come appartenente o meno ad una classe.
Nel caso multivariato è necessario calcolare la probabilità di appartenenza ad ogni classe. Per fare ciò, visto che si considera ogni classe come variabile con risposta binaria, una possibilità è quella di ricondursi alla funzione logistica, tramite:

$$\hat{f}(X^*) = \log \frac{\hat{Pr}(y_i = +1|X^*)}{1 - \hat{Pr}(y_i = +1|X^*)} = \log \frac{\hat{Pr}(y_i = +1|X^*)}{\hat{Pr}(y_i = -1|X^*)} \tag{3.18}$$

e quindi la probabilità di appartenza alla classe i è data da:

$$\hat{Pr}(y_i = +1|X^*) = \frac{1}{1 + \exp(\hat{b} + \sum_{j=1}^{n} \hat{a}_j y_j K(X^*, X_j))} \tag{3.19}$$

Permane anche per questo modello di classificazione la necessità di utilizzare metodi di regolarizzazione tramite convalida incrociata, vista la necessità di considerare

matrici definite positive a rango pieno.
Ricapitolando, l'algoritmo effettua la stima dei parametri sul *training set* ed utilizza le stime calcolate per ottenere la probabilità di appartenenza ad ogni classe per ogni osservazione del *test set*. Al termine dell'associazione di ogni osservazione alla classe in base alla maggiore propensione si calcola la misura di errore.
Il metodo sopra descritto è versatile, infatti è adattabile a partire da una qualsiasi funzione di distanza iniziale; nel caso in oggetto si considera solamente la metrica Riemanniana.
Si ricorda inoltre che è necessaria la scelta tramite convalida incrociata del parametro σ della funzione *kernel* presentata nell'equazione 3.5.
In letteratura non esistono strumenti che consentono l'applicazione di una *Support Vector Machine* a partire da una funzione di distanza selezionata, per questo motivo si è resa necessaria la completa implementazione del metodo. A causa del grande costo computazionale, in termini di tempo e di memoria, dato dalla necessità della costruzione della matrice *kernel* a partire della funzione di distanza, ma anche dato dal pre-preocessamento iniziale e dal calcolo del logaritmo matriciale, si è reso necessario l'utilizzo della libreria `RcppArmadillo` del linguaggio di programmazione C++ [Sanderson e Curtin 2016]. Inoltre è importante specificare che per trovare la soluzione al problema di massimizzazione si è utilizzata la funzione `solveQP` del pacchetto `quadprog` dell'ambiente R [R Core Team 2012] con l'aggiunta di un errore pari a $5e-04$.

Un'alternativa possibile alla creazione della *Support Vector Machine* a partire dalla funzione di distanza è quella di ricondurre il *kernel* Riemanniano al *kernel* a base radiale già utilizzato dai metodi standard presenti in letteratura.
Questo approccio ci consente di utilizzare gli strumenti esistenti che richiedono come *input* la matrice di dati senza il calcolo precedente della matrice *kernel*.
Anche in questo caso è obbligatorio considerare matrici definite positive ed a rango pieno tramite una trasformazione della matrice iniziale e un metodo di regolarizzazione.
Considerando un *kernel* Riemanniano, definito positivo per ogni σ, di tipo Log-Euclideo:

$$K_R(P_i, P_j) := \exp\left(-\frac{d_R^2(P_i, P_j)}{2\sigma^2} \right) : (P \times P) \to \mathbb{R} \tag{3.20}$$

con $d_R(P_i, P_j) = ||\Psi(P_i) - \Psi(P_j)||_F$ e $\Psi(\cdot) = \log(\cdot)$, come spiegato nella sezione 2.2.1, è possibile ricondursi alla distanza Riemanniana tra due matrici definite positive.
A partire da:

$$d_R(P_1, P_2) = \| \log(P_1^{-1} P_2) \| \tag{3.21}$$

ed utilizzando le proprietà del logaritmo matriciale valide su matrici simmetriche definite positive quali:

1. $\log(P_1 P_2) = \log(P_1) + \log(P_2)$;
2. $\log(P_1^{-1}) = -\log(P_1)$;

è facile ottenere la distanza d_R considerata all'interno della funzione *kernel*.
Infatti il *kernel* a base radiale è dato da:

$$K(P_i, P_j) := \exp\left(-\gamma \| P_i - P_j \|^2 \right) \tag{3.22}$$

mentre il *kernel* Riemanniano calcolato tramite le proprietà del logaritmo matriciale è dato da:

$$\begin{aligned} K_R(P_i, P_j) := \exp\left(-\frac{\| \log(P_1^{-1} P_2) \|^2}{2\sigma^2} \right) = \\ = \exp\left(-\frac{\| log(P_i) - log(P_j) \|^2}{2\sigma^2} \right) \end{aligned} \tag{3.23}$$

Si ricorda che la differenza tra matrici è uguale alla differenza delle matrici vettorizzate e che la norma di una matrice è pari alla norma della matrice vettorizzata, quindi:

$$\| log(P_i) - log(P_j) \|^2 = \| \overrightarrow{log(P_i)} - \overrightarrow{log(P_j)} \|^2 \tag{3.24}$$

Inoltre le variabili esplicative, all'interno di una *Support Vector Machine*, come spiegato, vengono considerate solo nella formazione del nucleo.
Ponendo $\gamma = \frac{1}{2\sigma^2}$ e utilizzando la vettorizzazione del logaritmo matriciale di ogni matrice pre-processata ci si riconduce al *kernel* a base radiale della 3.22. Una volta ottenuti i logaritmi delle matrici e la vettorizzazione corrisponente è possibile applicare la *Support Vector Machine* a base radiale sul *dataset* in cui ogni riga corrisponde ad un'immagine pre-processata adeguatamente.

Per l'implementazione sono state pre-processate le matrici corrispondenti alle immagini iniziali sulle quali è stato calcolato il logaritmo matriciale e la vettorizzazione tramite l'uso della libreria `RcppArmadillo` del linguaggio C++ [Sanderson e Curtin 2016] e si è considerato il pacchetto R `e1071` [Dimitriadou et al. 2005] per l'applicazione della *Support Vector Machine* a base radiale che stima la probabilità a posteriori di appartenza alla classe utilizzando l'ottimizzazione quadratica.

Sia nel caso della *Support Vector Machine* applicata a partire dalla matrice di distanze sia nel caso in cui ci si riconduce al *kernel* a base radiale per svolgere al meglio l'analisi è necessario stimare:

- il valore di regolizzazione, che completa il rango della matrice pre-processata, che risulta essere un buon compromesso tra varianza e distorsione, attraverso convalida incrociata;
- il valore di σ o γ che costituisce il *kernel* Riemanniano che tramite convalida incrociata da migliori risultati senza sovra-adattare il modello.

Successivamente alla stima dei parametri di regolarizzazione si utilizza il *dataset* di verifica per valutare l'errore di previsione.

Capitolo 4

Dati Caltech 101

I dati a nostra dispozione appartengono ad un famoso *set* di immagini, "Caltech 101", utilizzato per testare gli algoritmi di classificazione per il riconoscimento di oggetti e renderli comparabili ad altre soluzioni proposte.
Le immagini sono scaricabili dal sito https://www.vision.caltech.edu/Image_Datasets/Caltech101/, mentre, il *dataset* viene formato attraverso operazioni di apertura e pulitura per ogni immagine.
Il dataset si compone di 9144 immagini divise in 102 categorie. Per ogni categoria ci sono da un minimo di 40 ad un massimo di 800 immagini. La dimensione di ogni immagine è approssimativamente di 300 per 200 *pixel*.
Una parte molto importante dell'analisi Riemanniana è il pre-processamento delle immagini. Infatti, vista la necessità di considerare matrici simmetriche definite positive, non è possibile utilizzare direttamente le immagini presenti nel *dataset*.
Per rendere più agevoli le operazioni ogni immagine sarà codificata in scala di grigi; inoltre, saranno riscalate per considerare un numero accessibile di variabili, vista la numerosità elevata dei *pixel* di ogni immagine.
È di particolare importanza sottolineare che all'interno del *dataset* "Caltech 101" esistono molte categorie confondenti, costituite da immagini difficili da catalogare e facilmente assegnabili a più categorie.
Un esempio è la categoria *Background Google* costituita da 467 immagini di vario genere; alcune di queste raffigurano volti e per tale motivo potrebbero essere coerentemente assegnate alla classe *Faces*, figura 4.1.

Figura 4.1: Immagini della categoria *Background Google* raffiguranti volti

Prima immagine

Seconda immagine

Terza immagine

Esistono inoltre categorie a cui appartengono le stesse immagini ma leggermente ritagliate come ad esempio *Faces* e *Simply Faces*, figura 4.2, ovviamente difficili da classificare.

Figura 4.2: Immagini della categoria *Faces* e della categoria *Simply Faces*

Prima *Faces*

Seconda *Faces*

Prima *Simply Faces*

Seconda *Simply Faces*

Infine esistono categorie riguardanti il corpo e la testa di animali: è il caso ad esempio della categoria *Cougar Body* in contrapposizione alla categoria *Cougar Head*; in figura 4.3 si mostrano quattro immagini difficili da associare ad una o all'altra categoria in quanto tutte includono sia il corpo che la testa di un puma.

Figura 4.3: Immagini della categoria *Cougar Body* e della categoria *Cougar Head*

Prima *Cougar Body*

Seconda *Cougar Body*

Prima *Cougar Head*

Seconda *Cougar Head*

4.1 Regione di Interesse

Il *dataset* "Caltech 101" è composto da immagini raffiguranti oggetti, persone, animali; la maggior parte delle immagini presenta un primo piano dell'oggetto da classificare con sfondo bianco, esistono però anche alcune immagini che sono frammenti di un paesaggio o che raffigurano più oggetti, si mostrano alcuni esempi in figura 4.4.
Con lo scopo di migliorare l'accuratezza dell'analisi è quindi possibile ridurre la superficie dell'immagine ed identificarne una sotto regione, la regione di interesse [Bosch et al. 2007].
Per fare ciò esistono diversi metodi:

- Identificazione dei marcatori della figura: procedura manuale e soggettiva che richiede lunghe operazioni di processamento, utilizzata nell'analisi Procuste presente in appendice A.1;

- Identificazione dei contorni della figura: procedura versatile e utilizzabile a partire da trasformazioni dell'intensità delle immagini;

- Identificazione dei contorni attraverso l'algoritmo di Canny: algoritmo che effettua la sogliatura di isteresi in cui si considerano due valori soglia, alto

e basso, e si calcola e valuta il gradiente di ogni punto dell'immagine; un punto se ha gradiente con valore minore della soglia bassa non fa parte del contorno, se ha valore maggiore della soglia alta è parte del contorno e se invece ha valore compreso tra le due soglie viene accettato solamente se è contiguo ad un punto che costituisce il contorno [Green 2002].

Per semplicità si è deciso di generare un algoritmo di individuazione della regione di interesse, in ambiente R [R Core Team 2012], che attraverso il riconoscimento dei contorni della figura estrae un sotto riquadro di immagine che raffigura esclusivamente l'oggetto; nel caso in cui l'area considerata è una porzione troppo piccola di immagine, l'algoritmo estrae l'immagine iniziale privata di una parte di contorno di 10 *pixel* per lato.
Non è certo che individuando la regione di interesse si ottenga errore di previsione minore, quindi in fase classificazione verrà utilizzato sia il *dataset* iniziale sia quello generato tramite identificazione della regione di interesse.

Figura 4.4: Immagini di varie classi

Immagine classe *Chair*

Immagine classe Pizza

Immagine classe *Motorbike*

Immagini classe *Scorpion*

4.2 Stima e verifica

Generalmente all'aumentare della complessità dei modelli si nota un incremento della varianza e un decremento della distorsione. Tipicamente è necessario scegliere la complessità del modello in modo tale da ottenere un compromesso tra varianza e distorsione [Friedman et al. 2001].
Per poter calcolare adeguatamente l'errore di classificazione è necessario dividere i dati in un insieme di stima e in uno di verifica. I modelli considerati verranno allenati sull'insieme di stima. Gli errori di classificazione verranno invece misurati sull'insieme di verifica.
L'approccio stima e verifica permette il calcolo dell'errore di previsione; infatti calcolando l'errore di classificazione sul *dataset* di verifica, non utilizzato per la stima del modello, si riesce a dare una misura dell'errore accurata evitando i problemi di sovradattamento ai dati.
Se venissero utilizzate tutte le osservazioni sia per la stima del modello che per la misurazione dell'errore si otterrebbero risultati troppo ottimistici. Infatti, all'aumentare dell'adattamento ai dati, il modello si adatta anche alla parte di errore dovuta al caso e le previsioni sui dati futuri sarebbero erratiche [Friedman et al. 2001].
Di particolare importanza è anche la presenza di categorie sbilanciate, a tal proposito è fondamentale effettuare le analisi tenendo conto di tale sbilanciamento. Verrà allora considerato sia l'insieme di stima che un sotto-campione con un numero di osservazioni bilanciato per ogni classe. Nel secondo caso il numero di osservazioni considerate sarà minore, ma vista la presenza di classi bilanciate si otterrà una quota di varianza delle stime simile per ognuna di esse.

4.3 Misure dell'errore di classificazione

Un ruolo importante nei problemi di classificazione di immagini è il calcolo della misura d'errore; in base all'errore è infatti possibile confrontare i vari algoritmi.
La prima misura di errore di previsione calcolabile è il rapporto tra le immagini classificate scorrettamente ed il numero totale; questa misura è confrontabile con l'errore di previsione nullo dato dall'assegnazione di tutte le immagini alla classe più ricorrente, pari a $(1 - f_i(\mu o))$ con f_i frequenza relativa e μo classe modale.

Ulteriore misura predittiva importante ed utilizzata in letteratura è l'indice di *Mean Average Precision* (MAP). Il MAP identifica la precisione di classificazione di una sequenza come:

$$\text{MAP}_K = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} \frac{P(k)}{\min(n_i, k)} \tag{4.1}$$

dove n è il numero di osservazioni dell'insieme di verifica, n_i è il numero di osservazioni di ogni sequenza ad un *cut-off* K e $P(k)$ è la precisione ad un *cut-off* di osservazioni k, ovvero è il numero di osservazioni classificate correttamente su $K \leq n$ totali.
Nel caso considerato il MAP, ad ogni passo, non fa altro che confrontare la classe dell'insieme di verifica con il valore predetto dal modello di classificazione, perciò non ci sono differenze tra l'indice considerato e l'indice di precisione calcolato attraverso il rapporto delle osservazioni correttamente classificate ed il totale.
Per tale motivo, nel confronto dei risultati dei modelli considerati, si utilizzano il *mean average precision* e l'errore di classificazione equivalentemente, infatti vale che:

$$\text{MAP} = 1 - \text{ERRORE DI PREVISIONE} \tag{4.2}$$

per qualsiasi valore di $K \leq n$.

Capitolo 5

Risultati

5.1 Minima distanza dalla media Riemanniana

5.1.1 Classificazione con 102 categorie

Dato l'intero *dataset* di immagini formato da 102 classi, si calcola per ogni immagine la covarianza regolarizzata nei casi di bilanciamento e non del *dataset* di stima.
Considerando il caso bilanciato senza individuazione della regione di interesse e testando l'algoritmo tramite convalida incrociata per la scelta ottimale del parametro di regolarizzazione λ, si è individuato il minimo errore di classificazione ottenibile nell'insieme di stima (figura 5.1).

All'aumentare di λ l'errore medio di classificazione nell'insieme di stima subisce dapprima una graduale diminuzione ed in un secondo tempo un aumento sostanziale. Con λ pari a 30 si ottengono migliori risultati, tale valore quindi viene considerato nella stima dell'errore di previsione che risulta essere pari a 0.748 con relativo indice di precisione MAP pari a 0.262.
In tabella 5.1, si mostrano i risultati dell'algoritmo di classificazione nel caso in cui le immagini vengano pre-processate anche tramite l'individuazione della regione di interesse, con o senza bilanciamento del *dataset* di stima.

Figura 5.1: Errore di classificazione nel *dataset* di stima per la scelta del parametro di regolarizzazione - caso 102 classi

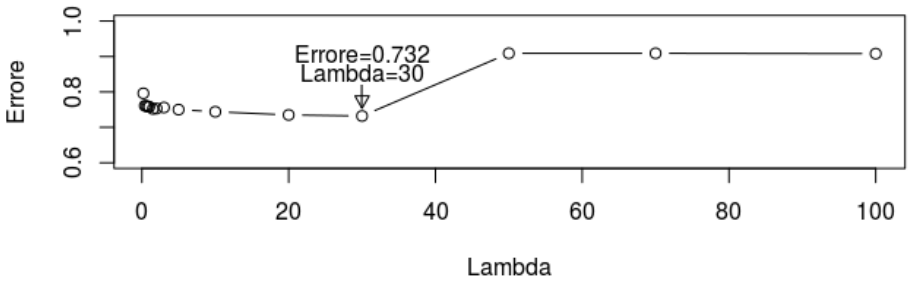


Tabella 5.1: Errore di classificazione con o senza individuazione della regione di interesse - caso 102 categorie

METODO	REGIONE DI INTERESSE	NUMEROSITA'	ERRORE	ERRORE NULLO
Bilanciamento	No	1428	0.748	0.990
Bilanciamento	Sì	1428	0.837	0.990
No Bilanciamento	No	5486	0.925	0.972
No Bilanciamento	Sì	5486	0.928	0.972

Effettuando il bilanciamento delle categorie nel *dataset* si ottengono minori errori di previsione in entrambi i casi, per tale motivo si sceglie di applicare bilanciamento nel *dataset* di stima in tutte le successive analisi.
Al contrario di quanto ci si poteva aspettare l'individuazione della regione di interesse all'interno dell'immagine porta a un peggiore errore di previsione. L'algoritmo considerato trova il baricentro Riemanniano di ogni classe e associa ogni nuova immagine in base alla vicinanza dal baricentro calcolato. Una possibile spiegazione è che se viene estrapolata una regione di interesse non adeguata, il baricentro della classe a cui appartiene l'immagine subisce una grossa variazione e ciò porta a risultati fuorvianti. Per ovviare a questo problema si potrebbe modificare l'algoritmo di selezione della regione di interesse e prestare maggiore attenzione nella selezione dei contorni della figura in ogni immagine.

5.1.2 Classificazione con 3 categorie

I dati Caltech 101 sono spesso utilizzati per testare gli algoritmi di classificazione di immagini. Per confrontare l'approccio proposto con altri già presenti nella letteratura è necessario considerare 3 delle 102 classi del *dataset*.
A tale scopo vengono considerate le classi:

1. *Snoopy*;
2. *Strawberry*;
3. *Crocodile*.

Per dimostrare la poca stabilità dei risultati ottenibili con la matrice di covarianza regolarizzata usata per rendere simmetrica la matrice iniziale, si considera anche la matrice standardizzata globalmente, cioè ottenuta sottraendone la media e dividendola poi per la deviazione standard totale, come annunciato nel capitolo 3, ottenendo matrici simmetriche definite positive pari a:

- $P_i(X_i) = Cov(X_i) + \lambda I$ =COV;
- $P_i(X_i) = (\frac{X_i - \mu_{X_i}}{\sigma_{X_i}})(\frac{X_i - \mu_{X_i}}{\sigma_{X_i}})^T + \lambda I$ =STAND.

Tabella 5.2: Confronto dell'indice di precisione medio MAP nel *dataset* di stima con i due tipi di pre-processamento COV e STAND in MDMR - caso 3 categorie

LAMBDA	MDMR_{COV}	MDMR_{STAND}
0.01	0.72	0.75
0.50	0.75	0.82
1	0.78	0.82
2	0.82	0.82
10	0.88	0.82
20	0.92	0.85
50	0.82	0.88
100	0.58	0.88
1000	0.65	0.85
5000	0.72	0.82
10000	0.75	0.82

Nella tabella 5.2 si mostra l'indice di precisione medio al variare del paramentro di regolazione, ottenuto attraverso convalida incrociata e si evidenzia come utilizzando la covarianza si ottengano risultati poco stabili, infatti pur avendo indice di

precisione maggiore, al variare di λ si hanno valori contrastanti, che seguono un trend oscillatorio. Si preferisce quindi considerare la matrice standardizzata per avere risultati più stabili.
In tabella 5.3 si mostrano gli indici di precisione e gli errori di previsione ottenuti nel *dataset* di verifica considerando il parametro di regolarizzazione che massimizza l'indice di precisione medio nel *dataset* di stima.

Tabella 5.3: Misure di errore nel *dataset* di verifica con λ scelto tramite convalida incrociata in MDMR - caso 3 categorie

MISURA D'ERRORE	$\text{MDMR}_{COV,\lambda=20}$	$\text{MDMR}_{STAND,\lambda=50}$
MAP	0.87	0.83
ERRORE DI PREVISIONE	0.13	0.17

I MAP calcolati sono confrontabili con l'indice di precisione massimo pari a 0.60 ottenuto da Bucak et al. 2014 tramite *Multiple Kernel Learning* con 10 iterazioni. Confronti con altri modelli presenti in letteratura e che utilizzano i dati Caltech 101 non sono possibili in quanto non viene chiaramente esposta la tecnica ed il *dataset* utilizzato durante la classificazione.
L'algoritmo proposto quindi, se pur semplice, porta a sostanziali miglioramenti nella precisione della classificazione.

5.2 Kernel Support Vector Machine Riemanniana

Come per la minima distanza dalla media Riemmaniana, anche per il secondo approccio, è necessario rendere simmetriche ed effettuare regolarizzazione sulle matrici di immagini, per avere matrici definite positive a rango pieno.
Si mostreranno i risultati sia utilizzando il *dataset* ottenuto tramite l'individuazione della regione di interesse sia il *dataset* originale. Inoltre verrà considerato sia il caso di pre-processamento con la matrice di covarianza per ogni immagine, sia il caso di pre-processamento con la matrice globalmente standardizzata.

5.2.1 Classificazione con 102 categorie

In tabella 5.4, si comprende come l'utilizzo della covarianza durante il pre-processamento porti a risultati meno stabili, in linea con quanto detto per la minima distanza dalla media Riemanniana. Visto il miglioramento dell'errore di classificazione medio nel *dataset* di stima solamente con un grosso incremento del parametro λ, è necessaria una più accurata scelta del parametro di regolarizzazione.
Individuando la regione di interesse si ha un indice di precisione minore: si suppone che per alcune immagini all'interno di una o più classi, sia più difficile individuare correttamente i contorni e si abbia quindi errore maggiore, introducendo incertezza nell'analisi.
Rispetto al caso del pre-processamento tramite covarianza, una volta considerata la regolarizzazione del parametro λ, con il pre-processamento tramite standardizzazione globale si ottengono risultati migliori, infatti si raggiunge un MAP pari a 0.30.
Questo approccio porta a precisione migliore anche rispetto alla minima distanza dalla media Riemanniana analizzata in 5.1, raggiungendo un errore di previsione minimo pari a 0.70 (tabella 5.5).

Tabella 5.4: MAP medio in Kernel Support Vector Machine Riemanniana - scelta parametro di regolarizzazione valutato nell'insieme di stima - caso 102 classi

PRE-PROCESSAMENTO	$\lambda = 1$	$\lambda = 10$	$\lambda = 50$	$\lambda = 100$	$\lambda = 500$	$\lambda = 1000$	$\lambda = 5000$	$\lambda = 10000$
$\text{COV}_{X_i} = COV(X_i) + \lambda_i$	0.23	0.28	0.28	0.28	0.30	0.30	0.31	0.30
$\text{COV}_{X_{ROI}} = COV(X_{ROI}) + \lambda_i$	0.23	0.24	0.25	0.26	0.26	0.27	0.27	0.26
$\text{STAND}_{X_i} = \frac{(X_i-\mu)}{\sigma}\frac{(X_i-\mu)}{\sigma}^T + \lambda_i$	0.30	0.33	0.33	0.33	0.33	0.33	0.31	0.30
$\text{STAND}_{X_{ROI}} = \frac{(X_{ROI}-\mu)}{\sigma}\frac{(X_{ROI}-\mu)}{\sigma}^T + \lambda_i$	0.31	0.32	0.31	0.31	0.31	0.31	0.30	0.29

Tabella 5.5: Errore di Previsione minimo in Kernel Support Vector Machine Riemanniana - caso 102 classi

PRE-PROCESSAMENTO	ERRORE DI PREVISIONE	ERRORE NULLO
$\text{COV}_{X_i} = COV(X_i) + \lambda$	0.733	0.972
$\text{COV}_{X_{ROI}} = COV(X_{ROI}) + \lambda$	0.759	0.972
$\text{STAND}_{X_i} = \frac{(X_i-\mu)}{\sigma}\frac{(X_i-\mu)}{\sigma}^T + \lambda$	0.700	0.972
$\text{STAND}_{X_{ROI}} = \frac{(X_{ROI}-\mu)}{\sigma}\frac{(X_{ROI}-\mu)}{\sigma}^T + \lambda$	0.716	0.972

Per confrontare al meglio la *Support Vector Machine* Riemanniana e mostrare che l'utilizzo della metrica Riemanniana porta a risultati più soddisfacenti, si calcolano l'errore di previsione ed il *mean average precision* di una *Support Vector Machine* con *kernel* a base radiale, presentata in sezione 2.3.1; in tabella 5.6 se ne riportano i risultati.

Tabella 5.6: Misure d'errore in Kernel Support Vector Machine con nucleo a base radiale- 102 classi

PRE-PROCESSAMENTO	ERRORE	MAP
X_i	0.712	0.288
X_{ROI}	0.720	0.280

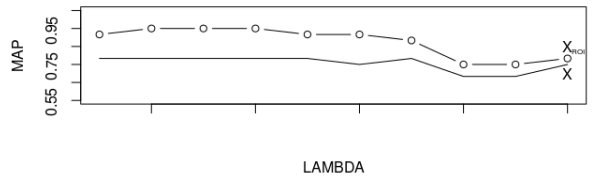
Con il pre-processamento dato dalla standardizzazione globale della matrice iniziale e con l'utilizzo della metrica Riemanniana si ottengono risultati più soddisfacenti rispetto al metodo standard valutato in tabella 5.6, ma soprattutto rispetto ai risultati ottenuti dalla minima distanza dalla media Riemanniana in 5.1; si ricorda inoltre che l'errore di classificazione nullo all'interno di un campione di 102 classi bilanciato risulta essere pari a 0.972.

5.2.2 Classificazione con 3 categorie

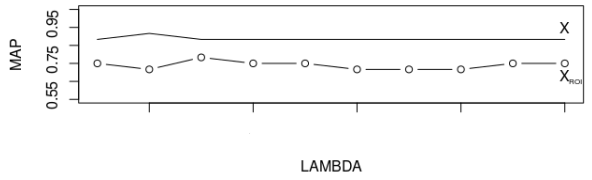
Considerando ora le 3 categorie, precedentemente elencate in 5.1, si confronta la differenza tra gli indici di *mean average precision* medi (complemento ad uno degli errori di classificazioni medi) nelle due tipologie di pre-processamento nell'insieme di stima, procedimento atto al riconoscimento del più adeguato valore di λ. Nella figura 5.2 si mostra la scelta del parametro di regolarizzazione nell'insieme di stima: utilizzando la standardizzazione si ha un andamento più stabile delle misure d'errore ma si ottengono valori più bassi, rispetto al caso della matrice di covarianza, per l'indice di precisione (e quindi valori più alti per l'errore di previsione).

In tabella 5.7, nella quale si riportano gli errori di previsione e l'indice calcolato nell'insieme di verifica dato il parametro λ selezionato con convalida incrociata sull'insieme di stima, si evidenzia come l'individuazione della regione di interesse

Figura 5.2: Map e errore di previsione nei diversi pre-processamenti - Valutazione del parametro di regolazione - 3 categorie



(a) Map medio con pre-processamento tramite covarianza nell'insieme di stima con e senza individuazione della regione di interesse



(b) Map medio con pre-processamento tramite matrice standardizzata nell'insieme di stima con e senza individuazione della regione di interesse

porti a migliore previsione nel caso in cui si utilizzi il pre-processamento tramite covarianza; situazione opposta nel caso in cui venga considerata la standardizzazione globale della matrice iniziale; inoltre, si conferma che il calcolo della matrice di covarianza porta a migliori risultati rispetto al pre-processamento con standardizzazione globale, ma non è detto che questo fenomeno valga per ogni categoria considerata.

Tabella 5.7: Minimo errore di previsione e MAP con o senza individuazione della regione di interesse nel pre-processamento con covarianza o con standardizzazione globale - caso 3 categorie

PRE-PROCESSAMENTO	ERRORE PREVISIONE	MAP
$\text{COV}_{X_i} = COV(X_i) + \lambda$	0.27	0.73
$\text{COV}_{X_{ROI}} = COV(X_{ROI}) + \lambda$	0.11	0.89
$\text{STAND}_{X_i} = \frac{(X_i-\mu)}{\sigma}\frac{(X_i-\mu)}{\sigma}^T + \lambda$	0.13	0.87
$\text{STAND}_{X_{ROI}} = \frac{(X_{ROI}-\mu)}{\sigma}\frac{(X_{ROI}-\mu)}{\sigma}^T + \lambda$	0.27	0.73

Viene spontaneo chiedersi a questo punto se la scelta di considerare solamente 3 classi sulle 102 totali del *dataset* non sia fuorviante. Infatti è possibile che le classi prese in considerazione siano quelle maggiormente distinte e, per tale motivo, è possibile che i risultati ottenuti siano particolarmente ottimistici.
Vista la presenza di un numero elevato di classi si può procedere ad un'analisi di sensibilità considerando 10 differenti campioni. Effettuando casualmente la scelta delle classi per ogni sotto campione non si considera solo il caso in cui le categorie sono ben distinte e si riescono ad ottenere risultati più robusti.
Dalla figura 5.3 e dalla figura 5.4 si nota come non ci sia un andamento stabile dei risultati e quindi in base al campione o in base al pre-processamento utilizzato si ottengono risultati differenti. In generale, a differenza del caso con 102 classi, si evince che l'individuazione della regione di interesse può migliorare la classificazione, infatti in molti casi si ottiene *mean average precision* medio maggiore o vicino al caso delle matrici originali, ma si suppone che sia necessario un algoritmo più accurato di selezione della regione stessa.
Inoltre si conferma che operando un pre-processamento di tipo standardizzato si ricavano risultati peggiori ma si ha maggiore stabilità, infatti è possibile che l'aggiunta del parametro di regolarizzazione, sulla matrice di varianza e covarianza non standardizzata, possa includere, in alcune osservazioni, troppa distorsione nell'analisi e compromettere i risultati.

Figura 5.3: Analisi di sensibilità - MAP medio pre-processamento con covarianza - Valutazione parametro di regolarizzazione - caso 3 categorie

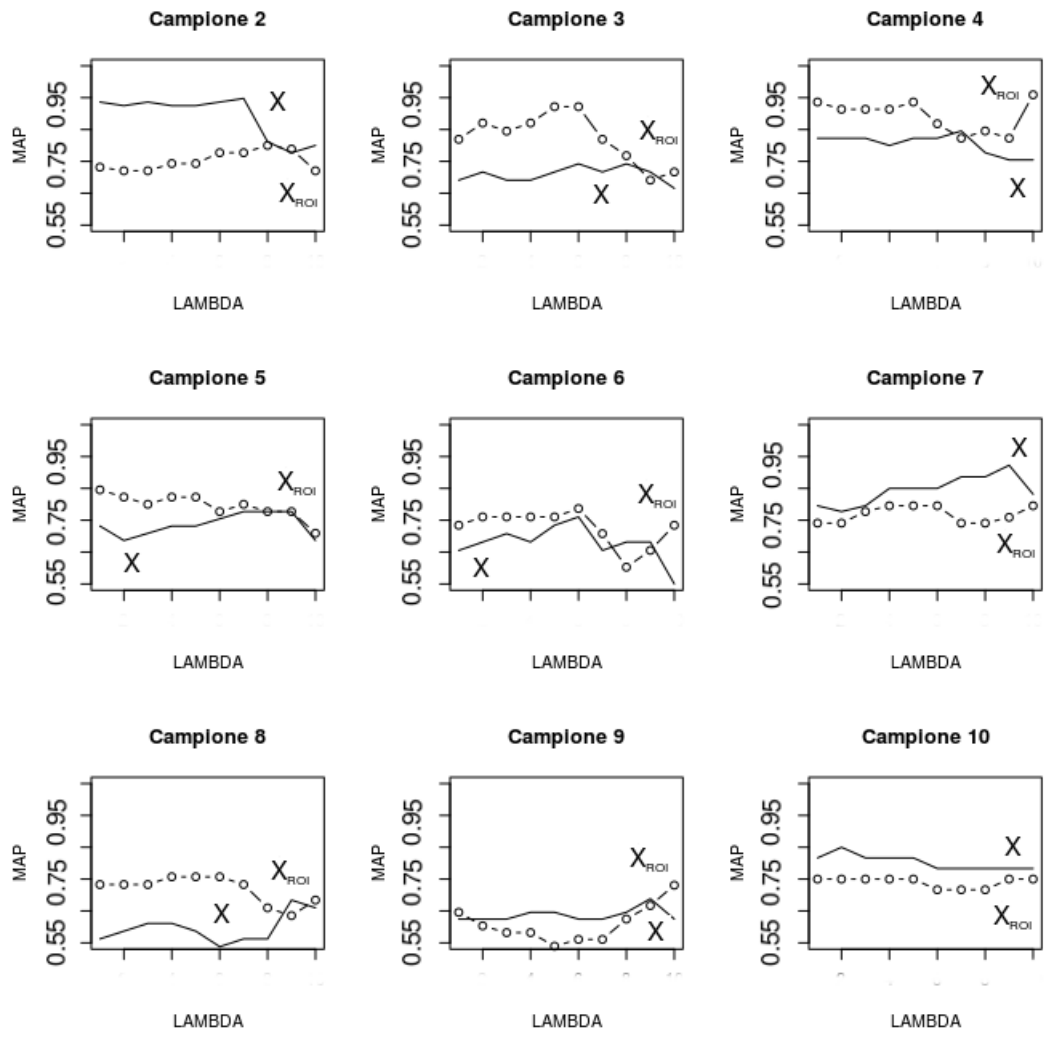
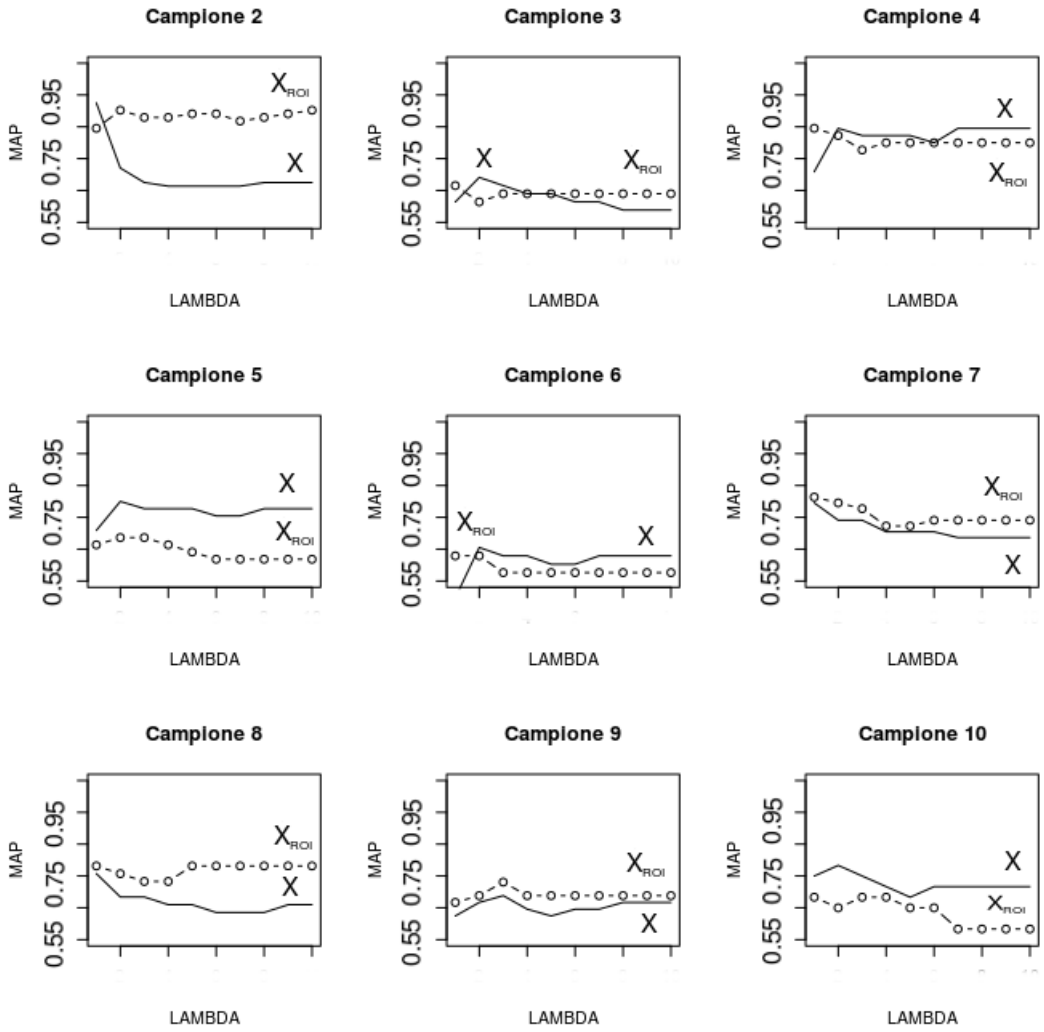


Figura 5.4: Analisi di sensibilità - MAP medio pre-processamento con standardizzazione - Valutazione parametro di regolarizzazione - caso 3 categorie



In tabella 5.8 si riportano infine i minimi errori di previsione raggiungibili dai diversi modelli di classificazione una volta considerato il valore del parametro di regolarizzazione λ che riporta errore medio di classificazione minore nell'insieme di stima. Gli errori ottenuti sono confrontabili con l'errore di previsione di una *Support Vector Machine* con *kernel* a base radiale.

Tabella 5.8: Minimo errore di previsione di 10 sotto campioni - Confronto tra *Support Vector Machine* Riemanniana e *Support Vector Machine* con *kernel* a base radiale - Analisi di sensibilità

	SVMR: $COV(X_i) + \lambda$	SVMR: $COV(X_{ROI}) + \lambda$	SVMR: $\frac{(X_i-\mu)}{\sigma}\frac{(X_i-\mu)}{\sigma}^T + \lambda$	SVMR: $\frac{(X_{ROI}-\mu)}{\sigma}\frac{(X_{ROI}-\mu)}{\sigma}^T + \lambda$	SVM_X	SVM_{ROI}
campione 1	0.27	0.11	0.13	0.27	0.37	0.37
campione 2	0.10	0.25	0.12	0.15	0.19	0.14
campione 3	0.31	0.13	0.36	0.38	0.28	0.46
campione 4	0.20	0.11	0.20	0.20	0.30	0.32
campione 5	0.27	0.20	0.25	0.36	0.34	0.59
campione 6	0.29	0.26	0.39	0.42	0.42	0.37
campione 7	0.13	0.25	0.25	0.24	0.18	0.20
campione 8	0.37	0.29	0.29	0.27	0.34	0.46
campione 9	0.36	0.38	0.36	0.32	0.38	0.38
campione 10	0.20	0.30	0.27	0.37	0.27	0.40
Errore Medio	0.25	0.23	0.30	0.30	0.31	0.37

In generale si registra errore di previsione maggiore applicando *Support Vector Machine* con *kernel* a base radiale, quindi considerando la metrica Riemanniana si ha un miglioramento dei risultati con errore di previsione inferiore.

Si vuole ora valutare se partendo dalla distanza e modellando una *Support Vector Machine* come spiegato nel capitolo 3, si ottengono risultati simili a quelli ottenuti con la riconduzione al *kernel* a base radiale appena mostrata.
I due approcci, seppur teoricamente simili, sono generati attraverso diverse tipologie di ottimizzazione dei parametri e di calcolo della soglia, spiegati in sezione 3.2; quindi a causa di approssimazione e ottimizzazioni differenti conducono a differenti risultati.
In tabella 5.9 si mostrano i minimi errori di previsione dei due approcci nel caso di standardizzazione globale con l'individuazione della regione di interesse; mentre in tabella 5.10 si mostra il caso in cui non venga individuata la regione di interesse. Gli errori di previsioni ottenuti vengono confrontati con gli errori minimi ottenibili tramite l'utilizzo della metrica Riemanniana con il procedimento di riconduzione ad una *Support Vector Machine* con *kernel* a base radiale e tramite una *Support*

Vector Machine con *kernel* a base radiale standard (effettuata tramite la funzione `svm` del pacchetto `e1071` dell'ambiente R e citata in sezione 2.3.1).

Tabella 5.9: Minimo errore di previsione di 10 sotto campioni - Kernel support vector machine Riemanniana implementata a partire dalla distanza, confronto con altri metodi - Individuazione regione di interesse - caso 3 categorie

ROI	SVMR Riconduzione a kernel a base radiale	SVMR A partire dalla distanza	SVM Kernel a base radiale
campione 1	0.27	0.27	0.37
campione 2	0.15	0.22	0.14
campione 3	0.38	0.28	0.46
campione 4	0.20	0.29	0.32
campione 5	0.36	0.27	0.59
campione 6	0.42	0.32	0.37
campione 7	0.24	0.35	0.20
campione 8	0.27	0.37	0.46
campione 9	0.32	0.32	0.38
campione 10	0.37	0.35	0.40
Errore Medio	0.30	0.30	0.37

Tabella 5.10: Minimo errore di previsione di 10 sotto campioni - Kernel support vector machine implementata a partire dalla distanza, confronto con altri metodi - Senza individuazione regione di interesse - caso 3 categorie

NO ROI	SVMR Riconduzione kernel a base radiale	SVMR A partire dalla distanza	SVM Kernel a base radiale
campione 1	0.13	0.26	0.37
campione 2	0.12	0.26	0.19
campione 3	0.36	0.35	0.28
campione 4	0.20	0.36	0.30
campione 5	0.25	0.29	0.34
campione 6	0.39	0.42	0.42
campione 7	0.25	0.25	0.18
campione 8	0.29	0.40	0.34
campione 9	0.36	0.28	0.38
campione 10	0.27	0.23	0.27
Errore Medio	0.30	0.31	0.31

Dall'analisi effettuata, nonostante le differenze nei risultati tra la *Support Vector Machine* Riemanniana ottenuta a partire dalla distanza e quella ottenuta con la riconduzione al *kernel* a base radiale, si confermano le ipotesi fatte precedentemente, infatti:

- una scelta migliore dei parametri di regolarizzazione può portare a migliori risultati
- l'individuazione della regione di interesse può ridurre nettamente gli errori di previsione
- l'utilizzo della metrica Riemanniana riconduce ad errori, la maggior parte delle volte, inferiori.

Risulta importante ricordare che l'implementazione della *Support Vector Machine* a partire dalla distanza ha molte possibili applicazioni nella classificazione delle immagini, poiché concede l'opportunità di utilizzare differenti metriche in differenti contesti; tuttavia le difficoltà di calcolo non hanno permesso un'ottimizzazione completamente soddisfacente.

Per dare completezza all'analisi si è deciso di effettuare un'ultimo confronto tra la *Support Vector Machine* Riemanniana a partire dalla funzione di distanza nel caso in cui si utilizza il pre-processamento tramite standardizzazione globale della matrice e nel caso in cui si utilizza il descrittore di caratteristiche spesso considerato in letteratura.
In tabella 5.11 si mostrano gli errori di previsione ottenuti tramite l'utilizzo del descrittore nella *Support Vector Machine* Riemanniana a partire dalla distanza e si confrontano con quelli relativi al pre-processamento tramite matrice standardizzata globalmente con parametro di regolarizzazione scelto con convalida incrociata, precedentemente riportati.
Per la SVMR a partire dalla distanza con pre-processamento tramite descrittore si è mostrato il confronto solo con il caso in cui non venga individuata la regione di interesse; infatti, utilizzando il descrittore si includono nell'analisi anche tutte le caratteristiche che vengono valutate nell'individuazione dei contorni di una figura. Invece nel caso di SVMR a partire dalla distanza con pre-processamento con la matrice standardizzata globalmente e nel caso di SVM a base radiale si è

considerata l'individuazione della regione di interesse; infatti, devono essere valutate nell'analisi le caratteristiche che vengono inserite implicitamente mediante il descrittore.

Tabella 5.11: Minimo errore di previsione di 10 sotto campioni - Kernel support vector machine implementata a partire dalla distanza - caso 3 categorie

	SVMR pre-processamento con descrittore	SVMR pre-processamento con standardizzazione	SVM Kernel a base radiale
campione 1	0.37	0.27	0.37
campione 2	0.19	0.22	0.14
campione 3	0.57	0.28	0.46
campione 4	0.06	0.29	0.32
campione 5	0.36	0.27	0.59
campione 6	0.42	0.32	0.37
campione 7	0.22	0.35	0.20
campione 8	0.27	0.37	0.46
campione 9	0.43	0.32	0.38
campione 10	0.37	0.35	0.40
Errore Medio	0.33	0.30	0.37

Dal confronto fatto si nota come l'utilizzo del descrittore per alcuni campioni porta ad un errore notevolmente minore, mentre per altri invece mostra picchi elevati con risultati instabili. Per questo motivo si suppone che sia preferibile utilizzare il pre-processamento a partire direttamente dalle immagini iniziali, nonostante la necessità di un metodo di regolarizzazione, in modo tale da non includere nell'analisi il grande onere computazionale dato dalla creazione del descrittore.

Conclusione

Con il cambiamento e l'evoluzione delle risorse *hardware* la classificazione di immagini è diventata negli anni sempre più importante. Questo lavoro di tesi ha avuto come obiettivo principale l'inclusione delle metriche Riemanniane in alcuni degli strumenti di analisi statistica presenti in letteratura.
Innanzitutto si è cercato un procedimento alternativo alla costruzione del descrittore di caratteristiche spesso utilizzato in letteratura e per tale motivo il preprocessamento delle immagini ha rivestito un ruolo cruciale nella costruzione di matrici definite positive necessarie all'applicazione delle metriche Riemanniane.
I metodi di classificazione proposti sono due: la minima distanza dalla media Riemanniana e la *Support Vector Machine* con *kernel* Riemanniano. Per la minima distanza dalla media si è generato un algoritmo che associa ogni nuova osservazione alla classe con cui presenta minore distanza Riemanniana; per la *Support Vector Machine* con *kernel* Riemanniano si sono considerati due approcci, nel primo si implementa una SVM a partire dalla funzione di distanza Riemanniana mentre nel secondo si riconduce il *kernel* Riemanniano ad un *kernel* a base radiale attraverso il logaritmo matriciale.
La novità apportata, non ancora presente in letteratura, è proprio l'implementazione di una *Support Vector Machine* a partire da una qualsiasi funzione di distanza con la quale si costituisce il nucleo riproducente; così facendo si costruisce un metodo versatile ed adattabile a diversi contesti. Infine sono stati effettuati i confronti tra i metodi proposti e quelli presenti in letteratura sulla base dell'errore di classificazione valutato nell'insieme di verifica effettuando anche, dove possibile, analisi di sensibilità.
Il principale risultato emerso è che l'utilizzo della geometria Riemmaniana all'interno degli algoritmi di classificazione, nonostante le difficoltà di calcolo, migliora,

seppur di poco, la precisione nella previsione.
Sia nell'algoritmo proposto di *Minima Distanza dalla Media Riemanniana* sia nel modello di classificazione considerato di *Support Vector Machine* con *Kernel* Riemanniano, si ottengono migliori risultati in relazione ai metodi standard che non considerano la metrica Riemanniana. Inoltre l'utilizzo del nucleo riproducente Riemanniano porta a migliori risultati rispetto all'uso del primo algoritmo di classificazione.
Nella minima distanza dalla media Riemanniana l'individuazione della regione di interesse sembra portare meno vantaggi con errori di previsione più alti, in contrasto con la *Support Vector Machine* con *kernel* Riemanniano.
Di rilievo è anche la necessità di considerare matrici definite positive, tuttavia grandi difficoltà emergono nella regolarizzazione del parametro che completa il rango delle matrici: nel caso di tre categorie e pre-processamento delle immagini tramite coviaranza si ha meno stabilità, data dall'inserimento di penalizzazione differente nelle diverse matrici, ma risultati migliori; nel caso di standardizzazione globale, invece, si ottengono risultati leggermente peggiori ma più stabili in fase di regolarizzazione. Nel caso di 102 categorie invece la situazione si inverte.
Infine di molta importanza risulta essere l'applicazione diretta sulle matrici di intensità senza il calcolo del descrittore di caratteristiche; dal confronto effettuato, si conclude che l'uso del descrittore porta a un peggioramento dei risultati con errori di previsione instabili e generalmente maggiori vista la possibile perdita di informazione dettata dall'uso di una misura di sintesi del descrittore.
La strada intrapresa che collega l'algebra Riemanniana ai metodi di classificazione porta grandi benefici, vale quindi la pena proseguire in tale verso raffinando i metodi e le tecniche utilizzate focalizzando l'attenzione sulle procedure di implementazione degli algoritmi.
In particolare si suppone che scegliendo con più precisione i possibili valori del parametro di regolazione λ, necessario al completamento del rango della matrice, e del parametro di regolarizzazione σ, che costituisce il *kernel* Riemanniano, si possano migliorare i risultati finali con una diminuzione dell'errore di classificazione.
Inoltre un miglioramento della stima dei parametri nella *Support Vector Machine* Riemanniana a partire dalla funzione di distanza con una conseguente riduzione dell'errore di stima può portare ad una stima migliore del confine decisionale ed a un errore di previsione minore.

Possibili sviluppi futuri sono anche il miglioramento dell'individuazione della regione di interesse attraverso più specifici algoritmi (ad esempio l'algoritmo di Canny, citato nel capito 4), e la riduzione del carico computazionale attraverso la diminuzione del numero delle osservazioni considerate nella stima; a questo proposito, una possibile strada è quella di considerare nell'analisi solo l'insieme dei vicini più prossimi di ogni punto soprattutto nel caso di classificazione con un numero elevato di categorie altrimenti difficile da valutare.

Appendice A

Materiale aggiuntivo

A.1 Analisi Procustiana

L'analisi Procustiana è la più famosa analisi della forma di una figura. Per definire una forma si considerano i *landmark* di un oggetto, ovvero i marcatori della forma utilizzati per disegnare degli pseudo-contorni (esempio riportato in figura A.1).

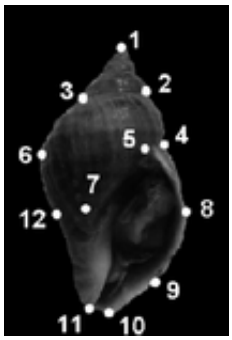


Figura A.1: *Landmark* di un immagine

Ad ogni *landmark* viene associata poi un'etichetta che lo identifica e che viene utilizzata per la corrispondenza nel confronto degli oggetti. Il metodo tradizionale propone di selezionare il rapporto di distanze tra *landmark* ed effettuare un'analisi multivariata. Il metodo geometrico va invece a considerare lo spazio ottenuto direttamente tramite le coordinate date dal calcolo del *landmark*, e tale procedimento

mantiene la configurazione geometrica dei punti [Dryden e Mardia 1998].

Per rappresentare la forma occorrono diverse misure. Interessano in modo particolare la configurazione, ovvero l'insieme dei marcatori di confine di un oggetto, la matrice di configurazione, cioè la matrice delle coordinate cartesiane dei k configuratori in m dimensioni e lo spazio di configurazione, ovvero lo spazio di tutte le possibili coordinate.
La taglia è la funzione positiva definita su valori reali della matrice di configurazione; essa va scorporata dall'oggetto per ottenere una misura della forma.
Data la matrice di configurazione X si può calcolare la taglia del centroide come:

$$S(X) = \| CX \| = \sqrt{\sum_{i=1,\dots,k} \sum_{j=1,\dots,m} (X_{ij} - \overline{X}_j)^2}, \qquad X \in \mathbb{R}^{km} \tag{A.1}$$

con k numero dei marcatori, m numero delle dimensioni reali, $\overline{X}_j$ media aritmetica e C matrice di centramento data da:

$$C = I_k - \frac{1}{k} 1_k 1_k^T \tag{A.2}$$

Dato che $\| X \| = \sqrt{\mathrm{Tr}(X^T X)}$ corrisponde alla norma Euclidea e $S(aX) = aS(X)$ è una misura di dimensione allora la misura del centroide corrisponde alla radice quadrata del quadrato della distanza euclidea di ogni singolo marcatore dal centroide:

$$S(X) = \sqrt{\sum_{j=1,\dots,k} \| X_j - \overline{X} \|^2} \tag{A.3}$$

dove $(X)_j$ è la j-esima riga di X e $\overline{X} = (\overline{X}_1, \dots, \overline{X}_m)$ è il centroide. Tale misura è una delle più utilizzate nell'analisi delle forme, solitamente normalizzata, nel caso in cui ci siano configurazioni con diverso numero di marcatori.

A.1.1 Coordinate di *Bookstein*

Le coordinate di *Bookstein* vengono utilizzate nel caso planare, quando la dimensione m vale 2. L'idea è quella di traslare, ruotare e riscalare i marcatori per calcolare la similarità su una posizione fissata. Le coordinate di *Bookstein* sono appunto le coordinate di un oggetto al netto di traslazione, rotazione e trasformazione di scala a partire da alcune coordinate iniziali e standardizzate.

Date (u_j^B,v_j^B) coordinate di *Bookstein* con $j = 3, \ldots, k$ e $k \geq 3$ numero di marcatori esse vengono calcolate dati due punti di base (x_1,y_1) e (x_2,y_2) attraverso le equazioni:

$$u_j^B = (x_2 - x_1)(x_j - x_1) + (y_2 - y_1)(y_j - y_1)/D_{12}^2 - \frac{1}{2} \tag{A.4}$$

$$v_j^B = (x_2 - x_1)(y_j - y_1) + (y_2 - y_1)(x_j - x_1)/D_{12} \tag{A.5}$$

dove $D_{12}^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2 > 0$ e $-\infty < u_j^B, v_j^B < \infty$.
La scelta dei punti di base è arbitraria; tuttavia si da particolare importanza a tale decisione, visto che da essa dipende la perdita di simmetria.
Un primo approccio è di effettuare un'analisi multivariata sulle coordinate di *Bookstein* ottenute ignorando la natura non Euclidea dello spazio considerato. Questo tipo di analisi può essere utilizzata per la stima della media o per test di ipotesi visto che la variazione nei dati è piccola. Per quanto riguarda il calcolo della variabilità si ha una situazione meno lineare infatti la trasformazione degli oggetti induce correlazione spuria.
Esistono altri vari tipi di coordinate possibili da calcolare ed associabili al caso planare come ad esempio le coordinate di *Kendall* che sono simili a quelle di *Bookstein* ma rimuovono diversamente la posizione o le coordinate triangolari di *Watson* [Dryden e Mardia 1998].

A.1.2 Analisi Procustiana sul piano

Uno degli aspetti importanti dell'analisi della forma è il calcolo della distanza tra due figure per stimare media e varianza in un campione casuale di immagini. Date due differenti configurazioni $y = (y_1, \ldots, y_k)^T$ e $w = (w_1, \ldots, w_k)^T$ appartenenti all'insieme dei numeri complessi $k-$dimensionale, con $y^* 1_k = w^* 1_k = 0$ e con y^* trasposta del complesso coniugato di y, per confrontare due forme si deve calcolare la distanza tra esse.
Una procedura interessante è quella di accoppiare w e y utilizzando la similarità euclidea e la differenza tra valori previsti e osservati di y per indicare la differenza tra le immagini.
Si considera l'equazione di regressione complessa:

$$y = (a + ib)1_k + \beta e^{i\theta} w + \epsilon = [1_k, w]A + \epsilon = X_D A + \epsilon \tag{A.6}$$

con $A = (a+ib, \beta e^{i\theta})^T$ vettore di parametri complessi con traslazione pari ad $a+ib$, con scala pari a $\beta > 0$, con rotazione $0 \leq \theta < 2\pi$, con ϵ vettore di errori complessi e con $X_D = [1_k, w]$ matrice di disegno di dimensione $k \times 2$. Tale equazione viene utilizzata per ottenere la super-imposizione di w su y:

$$w^P = X_D \hat{A} = (\hat{a} + i\hat{b})1_k + \hat{\beta} e^{i\hat{\theta}} w \tag{A.7}$$

stimando A ai minimi quadrati e (β, θ, a, b) minimizzando la funzione:

$$D^2(y, w) = \| \, y - w\beta e^{i\theta} - (a + ib)1_k \, \|^2 \tag{A.8}$$

Imponendo la standardizzazione tale per cui:

$$\sqrt{y^* y} = \sqrt{w^* w} = 1 \tag{A.9}$$

si riesce a dare una misura della distanza tra le forme ovvero la *Distanza Procustiana Completa*:

$$d_F(w, y) = \inf_{\beta,\theta,a,b} \left\| \frac{y}{\| \, y \, \|} - \frac{w}{\| \, w \, \|} \beta e^{i\theta} - a - ib \right\| = \left\{ 1 - \frac{y^* w w^* y}{w^* w y^* y} \right\}^{\frac{1}{2}} \tag{A.10}$$

Il termine *Completa* viene usato perché l'insieme delle similarità euclidee viene stimato al seguito delle traslazioni, rotazioni e trasformazioni di scala ma con y e w precedentemente scalate a dimensione unitaria [Dryden e Mardia 1998].
La stima Procustiana di w su y con relazione suriettiva viene calcolata tramite regressione lineare complessa di y su w. Nell'analisi Procustiana la trasformazione ottimale dei parametri viene stimata tramite il criterio dei minimi quadrati. Si ottiene quindi la stima dei parametri pari a:

$$\hat{a} + i\hat{b} = 0 \tag{A.11}$$

$$\hat{\theta} = -arg(y^* w) \tag{A.12}$$

$$\hat{\beta} = \frac{(w^* y y^* w)^{\frac{1}{2}}}{w^* w} \tag{A.13}$$

Considerando ora un campione casuale di configurazioni $w_1, \ldots, w_n$ disponibile a partire dal modello precedentemente enunciato:

$$w_i = \gamma_i 1_k + \beta_i e^{i\theta_i}(\mu + \epsilon_i), \qquad i = 1, \ldots, n \tag{A.14}$$

con γ_i il vettore che impone la traslazione, $\beta_i \in \mathbb{R}$ parametro di scala, $0 \leq \theta_i < 2\pi$ parametro di rotazione, ϵ_i vettore di errore complesso a media nulla e μ media delle configurazioni.
La media della forma della popolazione può essere stimata attraverso la minimizzazione della somma del quadrato della distanza di tutte le w_i dalla media della configurazione non nota:

$$\hat{\mu} = arg \inf_{\mu} \sum_{i=1}^{n} d_F^2(w_i, \mu) = arg \sup_{\|\mu\|=1} \mu^* S \mu \qquad dove \qquad S = \sum_{i=1}^{n} \frac{w_i w_i^*}{w_i^* w_i} \tag{A.15}$$

Per ottenere una misura della variabilità della somma si può considerare la radice media quadratica della distanza *Procustiana Completa* data da:

$$RMS(d_F) = \sqrt{n^{-1} \sum_{i=1}^{n} d_F^2(w_i, \hat{\mu})} \tag{A.16}$$

Una volta ottenuta la configurazione media si cerca di esaminare la variabilità della forma nel campione. Un metodo conveniente è quello di analizzare la variabilità nello spazio tangente alla media calcolata.
Considerando i vettori delle coordinate tangenti v_i con $i = 1, \ldots, n$ e i residui dell'analisi Procustiana:

$$r_i = w_i^P - \left(\frac{1}{n} \sum_{i=1}^{n} w_i^P \right) \qquad con \qquad w_i^P = w_i^* \hat{\mu} w_i / (w_i^* w_i) \tag{A.17}$$

si ottiene la matrice di covarianza delle coordinate tangenti pari a:

$$S_v = \frac{1}{n} \sum_{i=1}^{n} (v_i - \bar{v})(v_i - \bar{v})^T \qquad con \qquad \bar{v} = \frac{1}{n} \sum v_i \tag{A.18}$$

Attraverso l'analisi delle componenti principali di S_v con scomposizione in componenti principali e standardizzazione è possibile successivamente ottenere una misura di variabilità e la percentuale di essa catturata da ogni componente [Dryden e Mardia 1998].

A.1.3 Spazio delle forme e distanza Procustiana

Come già spiegato la trasformazione similare Euclidea di una matrice X è l'insieme delle operazioni di traslazione, rotazione e trasformazione scalare isotropica di X tale che:

$$\{\beta X \Gamma + 1_k \gamma^T : \beta \in \mathbb{R}^+, \Gamma \in SO(m), \gamma \in \mathbb{R}^m\} \tag{A.19}$$

dove β è il parametro di scala, γ vettore m-dimensionale di traslazione e Γ la matrice di rotazione per la quale vale che $\Gamma^T\Gamma = \Gamma\Gamma^T = I_m$ e $|\Gamma| = 1$.

Si può considerare la forma di X come la classe equivalente di tutti gli insiemi delle trasformazioni di similarità di una configurazione o alternativamente filtrare la similarità dalla configurazione attraverso una procedura automatica.

La traslazione è la più agevole da scorporare da X, infatti considerando il contrasto dei dati, basta pre-moltiplicare per una matrice adeguata. Più specificatamente si effettua un contrasto pre-moltiplicando X alla sotto matrice di Helmert:

$$X_H = HX \in \mathbb{R}^{(k-1)m} - \{0\} \tag{A.20}$$

La matrice di centramento idempotente, per togliere le informazioni relative alla locazione, sarà data da $C = H^T H$; per tale relazione si ha che $H^T X_H = CX$.
Si considera la pre-forma della matrice X data da:

$$Z = \frac{X_H}{\parallel X \parallel} \tag{A.21}$$

in modo tale da considerare una configurazione di X invariante sotto traslazione e trasformazione di scala [Dryden e Mardia 1998]. Per ottenere infine la forma dalla pre-forma è necessario considerare tutte le possibili rotazioni post-moltiplicando la pre-forma ad una matrice Γ appartenente all'insieme ortogonale di tutte le possibili rotazioni.
Lo spazio delle forme è l'insieme di tutte le possibili forme e la sua dimensione è pari a $M = km - m - 1 - \frac{1}{2}m(m-1)$ con m dimensione della locazione e k numero di marcatori della forma. Per considerare un insieme di forme è spesso conveniente utilizzare un'*icona* ovvero una forma dell'insieme di tutte le possibili forme sotto trasformazione di similarità euclidea atta a rappresentare una specifica classe. La distanza Procustiana Completa tra due matrici di k punti e m dimensioni X_1 e X_2 con pre-forma rispettivamente Z_1 e Z_2 è pari a:

$$d_F(X_1, X_2) = \inf_{\Gamma \in SO(m), \beta \in \mathbb{R}} \parallel Z_2 - \beta Z_1 \Gamma \parallel = \left\{ 1 - \Big(\sum_{i=1}^{m} \lambda_i \Big)^2 \right\}^{\frac{1}{2}} \tag{A.22}$$

dove SO(m) è l'insieme ortogonale di tutte le possibili rotazioni, $Z_r = \frac{HX_r}{\|HX_r\|}$ con $r = (1, 2)$, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{m-1} \geq |\lambda_m|$ sono le radici quadrate degli autovalori di $Z_1^T Z_2 Z_2^T Z_1$ e il più piccolo valore λ_m è la radice quadrata negativa se e solo se

$det(Z_1^T Z_2) < 0$.
La stima della rotazione ottenuta tramite minimizzazione della funzione obiettivo e, considerando la decomposizione a valori singolari di $Z_2^T Z_1$, vale:

$$\hat{\Gamma} = UV^T \tag{A.23}$$

dove $U, V \in SO(m)$ e $Z_2^T Z_1 = V\Lambda U^T$ con $\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_m)$. La stima del parametro di scala è dato da:

$$\hat{\beta} = \sum_{i=1}^{m} \lambda_i \tag{A.24}$$

Sapendo che gli autovalori λ_i sono compresi tra 0 e 1, per naturale conseguenza si ha che anche la distanza Procustiana è valida in tale intervallo.
Lo spazio tangente è una versione lineare dello spazio delle forme nelle vicinanze di un punto. Il punto generalmente considerato è chiamato *Polo* e corrisponde alla media della forma dei dati a disposizione.
Una buona approssimazione della distanza Procustiana si ottiene attraverso la distanza Euclidea nello spazio tangente nelle vicinanze del Polo.
Considerando il caso planare, dati i marcatori complessi con pre-forma:

$$z = (z_1, \ldots, z_{k-1}) = \frac{Hz'}{\parallel Hz' \parallel} \tag{A.25}$$

si possono ottenere le coordinate parziali Procusti di *Kent* tramite un Polo γ calcolato attraverso la media Procustiana Completa. Tali coordinate si estrapolano ruotando la configurazione attraverso un angolo θ il più vicino possibile al *polo* e proiettandolo sul piano tangente in γ, $T(\gamma)$. La stima di θ è data da $\hat{\theta} = arg(-\gamma^* z)$ e minimizza l'equazione $\parallel \gamma - ze^{i\theta} \parallel^2$.
Per considerare il piano procustiano tangente completo bisogna inserire nell'equazione anche il parametro di scala $\beta > 0$ e calcolare le coordinate minimizzando la funzione obiettivo $\parallel \gamma - \beta ze^{i\theta} \parallel^2$.
La soluzione comporta che $\hat{\beta}e^{i\hat{\theta}} = z^*\gamma$ e le coordinate risultano essere:

$$v_F = \hat{\beta}e^{i\hat{\theta}}[I_{k-1} - \gamma\gamma^*]z = zz^*\gamma - \gamma \parallel z^*\gamma \parallel^2, \quad v \in \quad T(\gamma) \tag{A.26}$$

Inoltre è importante sapere che le coordinate tangenti vengono approssimate correttamente dai residui dell'analisi procustiana [Dryden e Mardia 1998].

A.1.4 Analisi Procustiana ordinaria

L'analisi Procustiana ordinaria associa configurazioni con trasformazioni similari il più vicino possibile. Date due matrici X_1 e X_2, centrate senza perdita di generalità, attraverso la minimizzazione della distanza Euclidea quadratica:

$$D^2(X_1, X_2) = \| X_2 - \beta X_1 \Gamma - 1_k \gamma^T \|^2 \tag{A.27}$$

con $\| X \| = \{\mathrm{Tr}(X^T X)\}^{\frac{1}{2}}$ norma euclidea, e tramite l'utilizzo dei minimi quadrati si riescono a calcolare le stime dei parametri di similarità γ, Γ e β. La soluzione al problema di minimizzazione è data dalla somma dei quadrati ordinari Procusti:

$$OSS(X_1, X_2) = \| X_2 \|^2 \sin^2 \arccos\Big(\sum_{i=1}^{m} \lambda_i\Big) \tag{A.28}$$

$$\hat{\gamma} = 0 \qquad \hat{\Gamma} = U\hat{V}^T \qquad \hat{\beta} = \frac{\mathrm{Tr}(X_2^T X_1 \hat{\Gamma})}{\mathrm{Tr}(X_1^T X_1)} \tag{A.29}$$

La previsione Procustiana Completa di X_1 su X_2 con relazione suriettiva risulta essere:

$$X_1^P = \hat{\beta} X_1 \hat{\gamma} + 1_k \hat{\gamma}^T \tag{A.30}$$

e la matrice dei residui calcolata dopo l'associamento Procustiano è $R = X_2 - X_1^P$. Esaminando la matrice dei residui è facile osservare e fare diagnosi riguardo le differenze tra le forme.
Occorre specificare che, vista l'imposizione di parametri di similarità, invertendo l'ordine delle matrici si ottengono risultati differenti.

A.1.5 Analisi Procustiana generalizzata

Nel caso in cui il numero delle matrici di configurazione è maggiore di 2 e quindi si ha a disposizione un campione casuale, si vuole stimare una media della forma del campione.
Si considera il modello di perturbazione

$$X_i = \beta_i(\mu + E_i)\Gamma_i + 1_k \gamma_i^T \tag{A.31}$$

con E_i matrice di errori casuali indipendenti con media nulla, μ matrice di configurazione della media, β_i, Γ_i e γ_i parametri di scala, rotazione e traslazione rispettivamente.

L'analisi Procustiana completa generalizzata va a minimizzare una quantità proporzionale a:

$$G(X_1, \ldots, X_n) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \parallel (\beta_i X_i \Gamma_i + 1_k \gamma_i^T) - (\beta_j X_j \Gamma_j + 1_k \gamma_j^T) \parallel^2 \quad \text{(A.32)}$$

con $\Gamma_i \in SO(m)$, $\beta_i > 0$, $\parallel X \parallel = \sqrt{\mathrm{Tr}(X^T X)}$ e soggetta a vincolo sulla dimensione della media $S(\overline{X}) = 1$ dove $S(X)$ è la dimensione del centroide e la media della configurazione vale:

$$\overline{X} = \frac{1}{n} \sum_{i=1}^{n} (\beta_i X_i \Gamma_i + 1_k \gamma_i^T) \quad \text{(A.33)}$$

Tale operazione effettua *matching* dopo aver traslato, ruotato e riscalato ogni figura con lo scopo di minimizzare la somma quadratica delle distanze Euclidee.
La previsione Procustiana completa di ogni X_i è data da:

$$X_i^P = \hat{\beta}_i X_i \hat{\Gamma}_i + 1_k \overline{\gamma}_i^T, \quad i = 1, \ldots, n \quad \text{(A.34)}$$

dove $\hat{\Gamma}_i \in SO(m)$, $\hat{\beta}_i > 0$ e $\hat{\gamma}_i^T$ sono i parametri di similarità che minimizzano la funzione obiettivo e che non sono di primaria importanza nell'analisi della forma. Calcolando poi la media aritmetica delle previsioni Procusti:

$$\overline{X} = \frac{1}{n} \sum_{i=1}^{n} X_i^P \quad \text{(A.35)}$$

si ottiene la stessa forma della media Procustiana completa calcolata tramite:

$$\hat{\mu} = arg \inf_{\mu : S(\mu)=1} \sum_{i=1}^{n} d_F^2(X_i, \mu) \quad \text{(A.36)}$$

tenendo presente che $G(X_1, \ldots, X_n) = \inf_{\mu : S(\mu)=1} \sum_{i=1}^{n} d_F^2(X_i, \mu)$ [Dryden e Mardia 1998].

Bibliografia

Anzai, Yuichiro (2012). *Pattern recognition and machine learning*. Elsevier.

Barachant, Alexandre et al. (2012). «Multiclass brain–computer interface classification by Riemannian geometry». In: *IEEE Transactions on Biomedical Engineering* 59.4, pp. 920–928.

Bosch, Anna, Andrew Zisserman e Xavier Munoz (2007). «Image classification using random forests and ferns». In: *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, pp. 1–8.

Bucak, Serhat S, Rong Jin e Anil K Jain (2014). «Multiple kernel learning for visual object recognition: A review». In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.7, pp. 1354–1369.

Dimitriadou, Evgenia et al. (2005). «Misc Functions of the Department of Statistics (e1071), TU Wien». In: *R package version*, pp. 1–5.

Dryden, Ian L e Kanti V Mardia (1998). *Statistical shape analysis*. Vol. 4. J. Wiley Chichester.

Dryden, Ian L, Alexey Koloydenko e Diwei Zhou (2009). «Non-Euclidean statistics for covariance matrices, with applications to diffusion tensor imaging». In: *The Annals of Applied Statistics*, pp. 1102–1123.

Friedman, Jerome, Trevor Hastie e Robert Tibshirani (2001). *The elements of statistical learning*. Vol. 1. Springer series in statistics Springer, Berlin.

Green, Bill (2002). «Canny edge detection tutorial». In: *Retrieved: March* 6, p. 2005.

Jayasumana, Sadeep et al. (2013). «Kernel Methods on the Riemannian Manifold of Symmetric Positive Definite Matrices». In:

Kanopoulos, N., N. Vasanthavada e R. L. Baker (1988). «Design of an image edge detection filter using the Sobel operator». In: *IEEE Journal of Solid-State Circuits* 23.2, pp. 358–367. ISSN: 0018-9200. DOI: 10.1109/4.996.

Ledoit, Olivier e Michael Wolf (2004). «A well-conditioned estimator for large-dimensional covariance matrices». In: *Journal of multivariate analysis* 88.2, pp. 365–411.

Mendicelli, Amerigo (2007). «Riconoscimento automatico delle forme del paesaggio». Tesi di laurea mag.

R Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria.

Ricerche, ILCNDDTS e Monte Porzio Catone (2009). «Individuazione automatica di difetti presenti in lastre radiografiche digitalizzate». In:

Sanderson, Conrad e Ryan Curtin (2016). «Armadillo: a template-based C++ library for linear algebra». In: *Journal of Open Source Software* 1.2, pp. 26–32.

Tuzel, Oncel, Fatih Porikli e Peter Meer (2007). «Human Detection via Classification of Riemannian Manifolds». In:

Wahba, Grace et al. (1999). «Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV». In: *Advances in Kernel Methods-Support Vector Learning* 6, pp. 69–87.

Xie, Shuisheng (2013). «A Riemannian Framework for Shape Analysis of Subcortical Brain Structures». Tesi di dott. Ohio University.

Zhang, Hao et al. (2006). «SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition». In:

Zhang, Lijun et al. (2015). «Analysis of Nuclear Norm Regularization for Full-rank Matrix Completion». In: *arXiv preprint arXiv:1504.06817*.