

# Foot classifier

## Progetto IC

Luigi Lomasto, Marco Mecchia

29 Febbraio 2016

Prof. Roberto Tagliaferri

Dott. Michele Fratello

Dott. Paola Galdi

Dott. Angela Serra

# Outline

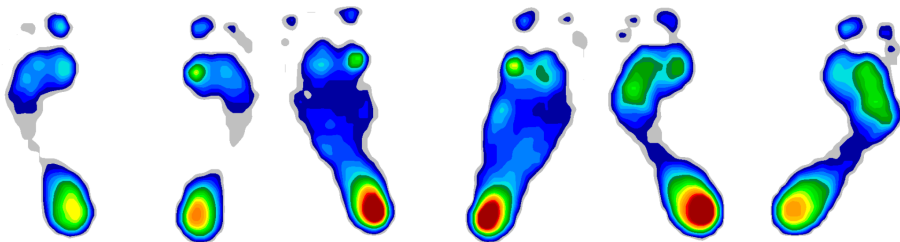
- 1 Introduzione al problema
- 2 Dataset
- 3 Preprocessing
  - Trasformazione delle immagini
  - Divisione, rotazione e cropping delle immagini
- 4 Features extraction
  - Features primo classificatore
  - Features secondo classificatore
- 5 Features Selection
- 6 Scelta dei classificatori
- 7 Infrastruttura
- 8 Risultati
  - Primo classificatore

# Il problema

Dato un insieme di immagini piedebarometriche, classificare automaticamente le seguenti patologie:

- Cavo
- Piatto
- Normale
- Valgo

# Patologie



# Struttura del dataset

Il dataset è costituito da 190 piedi (95 coppie) di cui:

- 121 Cavi
- 13 Piatti
- 56 Normali

per la prima classe di patologie, mentre:

- 88 Valghi
- 102 Normali

per la seconda classe di patologie.

# Panoramica

Nella fase di preprocessing, abbiamo eseguito le seguenti operazioni.

- Conversione delle immagini .bmp in .png.
- Pulizia delle immagini (rimozione del baricentro).
- Trasformazione delle immagini in scala di grigio.
- Divisione, rotazione e cropping dei piedi.

# Trasformazione delle immagini

- Trasformazione da uno spazio 3D (R,G,B) ad uno spazio 1D.
- Necessaria per valutare la pressione in un pixel dell'immagine.
- Problema: Come associare ad ogni tripla (R,G,B) il valore di intensità corretto?

# Prima soluzione

- Clustering sulle immagini.
- Trovati i centroidi, sostituire ogni pixel dell'immagine con il rappresentante del proprio cluster.
- Ordinare i centroidi per intensità crescente e associare valori compresi tra 0 e 1.
- Problemi: Trovare il  $k$  adatto, trovare un ordinamento per i cluster.



# Soluzione definitiva

- Variante della prima soluzione.
- Possibile grazie alla scala colorata delle pressioni fornita dal medico.



# Rotazione e cropping delle immagini

- La divisione é stata necessaria per separare piede destro da piede sinistro.
- La rotazione non é strettamente necessaria.
- Il cropping rimuove la parte di sfondo superflua.

# Rotazione del piede

Per ruotare il piede abbiamo utilizzato un algoritmo molto semplice:

- 1 Abbiamo trovato il centro

# Features extraction

Terminata la fase di preprocessing siamo passati alla fase di features extraction. Nel nostro caso, le regioni d'interesse di ogni piede differiscono significativamente tra loro. Per questo motivo, è stato necessario implementare algoritmi ad-hoc per l'estrazione delle features.

Dovendo lavorare con due classi di patologie:

- Cavo, piatto e normale.
- Valgo e normale.

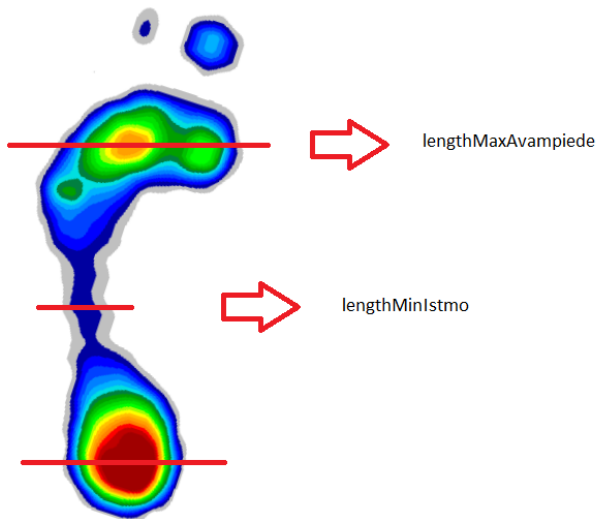
sono stati implementati due algoritmi per l'estrazione delle features.

## Features primo classificatore (1/2)

Le features estratte per le patologie appartenenti alla prima classe sono le seguenti:

- **lengthMinIstmo**: esprime la lunghezza minima che assume l'istmo.
- **lengthMedialstmo**: esprime la lunghezza media dell'istmo.
- **lengthMaxAvampiede**: esprime la massima lunghezza che assume l'avampiede.
- **indexPathology**: Si ottiene dal rapporto  $\frac{\text{lengthMaxAvampiede}}{\text{lengthMinIstmo}}$
- **mediumPressure**: Indica la pressione media esercitata dal piede.

## Features primo classificatore (2/2)



# Features secondo classificatore (1/2)

Le features estratte per le patologie appartenenti alla seconda classe sono le seguenti:

- juve
- juve

# Features secondo classificatore (2/2)

immagine



# Features Selection

La features selection è stata fatta in modo esaustivo.

Motivi:

- Basso numero di features usate.
- Si valutano tutti i possibili sottoinsiemi di features.

Sottoinsiemi valutati:

- 31 per il primo classificatore.
- 15 per il secondo classificatore.

# Classificatori

## SVM Gaussiano (Radial-Basis-Function).

Scelta dovuta ai seguenti motivi :

- RBF ben di adatta a problemi di classificazione dove il dataset è significativamente più grande rispetto al numero di dimensioni.
- Reti Neurali troppo complesse per il problema affrontato.
- SVM lineare generalmente usato quando si ha alta dimensionalità.

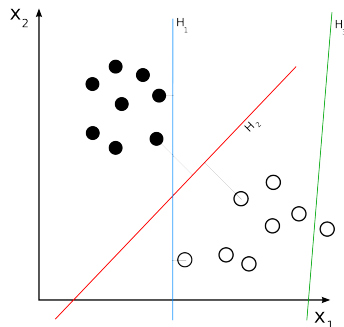
## SVM Lineare.

Motivo:

- Per la prima classe di problemi abbiamo individuato delle features con le quali è possibile generare un iperpiano di separazione.

# SVM Lineare

- Un classificatore SVM costruisce un iperpiano (o insieme di iperpiani) che può essere usato per fare classificazione.
- Una buona separazione è data dall'iperpiano che massimizza la distanza dal punto dei dati più vicino

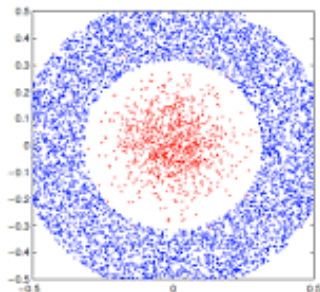


# Kernel

Spesso si ha a che fare con classi non separabili linearmente.

**Kernel:** Rendono utilizzabili modelli lineari con dati non linearmente separabili

- Mapping dei dati in dimensioni più alte
- Applicano un modello lineare nel nuovo spazio



# Kernel RBF (Radial-Basis-Function)

**Kernel RBF:**  $k(x, z) = \exp(-\gamma \|x_i - x\|^2)$

Per scegliere la configurazione ottimale abbiamo testato tutte le possibili combinazioni con:

- $10^{-3} < C < 10^3$
- $10^{-2} < \gamma < 10^0$

# Infrastruttura

L'infrastruttura software è la seguente:

- $\forall$  Sottoinsieme di features
  - Per N volte
    - Dividiamo il dataset in train e test
    - Cross Validation sul train per stimare la configurazione migliore.
    - Calcolo delle accuratèzze sul test.
  - Calcolo delle accuratèzze medie.

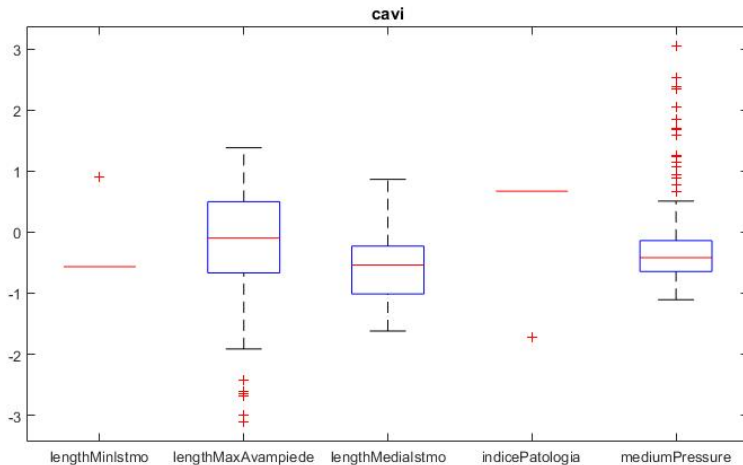
Il sottoinsieme scelto è quello con accuratezza totale media più alta.

# Scelta della migliore configurazione: T-test

Per valutare la migliore configurazione ci siamo serviti del T-test.

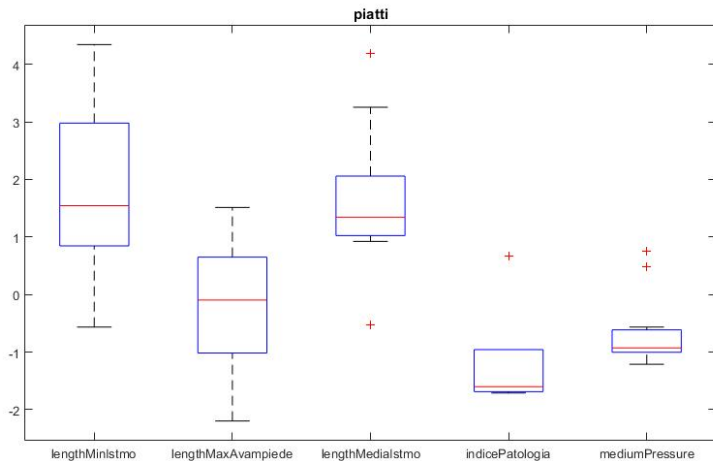
- Per ogni configurazione abbiamo ottenuto una distribuzione di accuratezze.
- T-test tra ogni configurazione e la successiva per studiarne la significatività.
- Se c'è significatività, (differenza non casuale) si prende la configurazione con accuratezza media più alta.

# BoxPlot delle features per i piedi Cavi





# BoxPlot delle features per i piedi Piatti



# BoxPlot delle features per i piedi Normali

