

# Approccio Machine Learning

## Definizione finestre temporali:

Le finestre temporali vengono valutate nel seguente modo:

si sceglie una variabile  $k=n+1$ ,  $n$  è  $2^* h$ , dove  $h$  è la dimensione di una finestra temporale.

la feature specifica di un punto (frame) è valutata osservando le relazioni (in termini di energia cinetica repetitiveness, ecc.) fra la finestra  $hk$  prima di quello specifico frame, e la finestra  $hk$  dopo quel frame.

Ad esempio, se utilizziamo  $k=301$ , significa che vengono valutate le relazioni fra la finestra da 0 a 149 e la finestra che comprende frame da 151 a 299 e le feature sono associate al frame 150. (una finestra di 150 frame corrisponde a 3 secondi con 50 frame al secondo, questo è l'eventuale ritardo in un sistema online).

Di seguito verrà chiamata  $w1$  la finestra prima del frame preso in considerazione, e  $w2$  la finestra successiva.

La coppia di finestre temporali successive viene ottenuta "shiftando" di un frame per volta.

## Features utilizzate:

- **Point Density**
- **Kinetic energy globale.**
- **Kinetic energy della testa**
- **Kinetic energy parti distali (KEDP)** (polso e caviglie), 4 features, una per ogni punto distale.
- **Repetitiveness point density** feature per rilevare quando vi è un cambio da un'azione più caotica, ad una più ripetitiva (e viceversa).
- **Repetitiveness Kinetic Energy dei punti distali**, queste features ( in totale 4, una per ogni punto distale) ci servono per rilevare se persiste una periodicità, ma con frequenza fondamentale diversa, oppure più semplicemente, su arto differente (e.g. polso destro invece che sinistro, ecc)

## Definizione feature **Point Density** e **KEDP**

Viene valutata la media della point Density (o delle KEDP) della finestra  $w1$ , quella della finestra  $w2$ , e viene considerata come features del frame in oggetto il valore assoluto della differenza delle 2 medie.

## Definizione feature **Repetitiveness** (su Point Density o KEDP):

Per valutare la periodicità di una finestra temporale (eg. 100,150 frames,  $k=201$  o 301 in questo esempio) considerate 2 misure (che utilizzano tecnologie reperibili dal paper che discute la lightness e la fragility):

1. **Periodicity** (rarity nel paper), questo valore viene calcolato (nel nostro contesto) valutando lo spettro del segnale (il segnale è definito dai frame della finestra temporale ed ha come freq. di campionamento 50 Hz, dunque 25 Hz di banda per evitare Aliasing), dallo spettro viene valutata l'ampiezza più grande e la frequenza sulla quale questa giace. Dopodichè viene computata la seguente somma di Elementi:  
 $\sum E_i$  (per  $i$  diverso da  $j$ , dove  $j$  è l'indice della frequenza con ampiezza massima).  
 $E_i$  è dato dalla differenza fra il valore di ampiezza massima ( $A_j$ ) nello spettro e la ampiezza  $i$ -esima associata alla frequenza  $i$ -esima ( $A_i$ ).
2. **Regularity**, questo valore viene calcolato osservando la distanza fra l'ampiezza massima  $A_j$  e ampiezze molto vicine alla massima (es. tolleranza di 5%), se non ci sono ampiezze che hanno questa caratteristica il valore di Regularity rimane 1, perchè se l'ampiezza più significativa è la sola, allora siamo di fronte a un segnale periodico, altrimenti viene eseguita la seguente computazione:

si valuta la distanza fra la frequenza  $j$ -esima (freq. fondamentale dove giace l'ampiezza massima) e 0 e la si aggiunge a un vettore di distanze come prima componente, dopodichè vengono aggiunte a questo vettore di distanze quella fra la frequenza  $j$ -esima e la successiva frequenza ampiezza dentro la tolleranza, successivamente viene aggiunta la distanza fra la successiva e quella ancora dopo, ecc.

Fatto ciò vengono presi il valore massimo e il valore minimo delle componenti, se sono molto vicini (e.g. circa 0) significa che sono tutte ampiezze a circa  $k*f_0$  dove  $k$  è un numero intero e  $f_0$  è la fondamentale. Se è questo il caso il valore di Regularity rimane 1, altrimenti restituisce 0.1 (attenua la Periodicity).

Dunque la Regularity è una misura **qualitativa** della periodicità (booleana).

La Repetitiveness della specifica finestra temporale è computata come il prodotto di Periodicity e Regularity.

E la features è definita come il rapporto tra la rep. della finestra ( $w1, w2$ ) che ha un valore più piccolo di Repetitiveness, e quella che ha un valore più grande.

Dunque una feature vicina a 0 indica che la Repetitiveness della finestra precedente è molto più grande (o più piccola) della successiva.

## GROUND TRUTH E ANNOTAZIONI

Abbiamo considerato una salienza per ogni volta che il soggetto cambia movimento invece di considerare un'annotazione più olistica perché dobbiamo orientarla all'algoritmo ML (che deve gestire un database sbilanciato, con queste euristiche abbiamo più salienze rispetto ad un'annotazione olistica).

## DATA CLEANING

Per rimuovere i NAN dai file tsv abbiamo considerato una interpolazione lineare, osservando il valore prima del primo NAN (lower) ed il valore dopo l'ultimo NAN (upper), abbiamo considerato quanti NAN ci sono in mezzo (n) e abbiamo computato il seguente valore:

$\text{incremento} = (\text{upper} - \text{lower}) / (n + 1)$ .

Infine per ogni  $\text{NAN}_i$  abbiamo assegnato il seguente valore ( $\text{lower} + i * \text{incremento}$ ) (i da 1 a n).

## INCONTRO CON PROFESSORE ONETO

1. finestre temporali: Per affrontare il problema dello sbilanciamento il prof propone di considerare Z finestre temporali sopra il frame di salienza (finestre temporali diverse fra loro perché traslate, ma che hanno al loro interno il frame saliente) e quindi etichettarle a 1, questa procedura viene fatta per ogni frame saliente. Per i campioni etichettati a 0, viene fatta la stessa procedura però invece che sul frame saliente, su un frame in mezzo a 2 frame salienti (dunque in quel frame la salienza è minima). Alla fine si dovrebbero ottenere circa lo stesso numero di campioni classificati 1 e classificati 0.

Di seguito i paper per l'estrazione delle features dalle finestre temporali suggeriti dal prof.

<https://github.com/gabrisera/TesiMag/tree/main/papers>

2. Utilizzo Random Forest come algoritmo.
3. Mantenere approccio classification.
4. Per la generalità dell'algoritmo 3 opzioni (in ordine di difficoltà):
  - 4.1. LOSO= Leave one Saliency out, mettere nel test set tutte le finestre associate ad una specifica salienza (prelevare una salienza per ogni video).
  - 4.2. LOVO= Leave one Video out, rimuovere per ogni soggetto (Muriel, Cora, Marianne) 1 take dal training set, ed utilizzarlo solo nel test set.
  - 4.3. LODO= Leave one Dancer out, mettiamo nel test set tutti i take di una specifica ballerina (Muriel, Cora, Marianne).

# Approccio Statistico

Algoritmo utilizzato: CUSUM

Per il cusum, sto seguendo la versione definita su [wikipedia](#) anche se ci sono altre versioni del CUSUM, che sarebbero interessanti da applicare, ma per ora sto ancora alla versione di "base".

Il concetto è che dati un dataset, io considero una finestra temporale di 100 frame = 1 secondo = 100 elementi del mio dataset e per questi 100 elementi io calcolo la mia media e la standard deviation di questa finestra.

Riparto dal primo elemento e ogni elemento calcolo il suo [standard score](#) definito come:

$$Z_t = \frac{X_t - \hat{\mu}_X}{\hat{\sigma}_X}$$

Ovvero  $X_t$  è il singolo elemento del mio dataset meno la media della finestra diviso la standard deviation. Questo per avere dei valori normalizzati.

Perciò ad ogni nuovo  $Z_t$ , lo sommo, per questo si chiama Cumulative SUM, e se la somma supera una certa soglia, è un changepoint. La somma definita così

$$S_{H_{n+1}} = \max(0, S_{H_n} + Z_{n+1} - \omega)$$

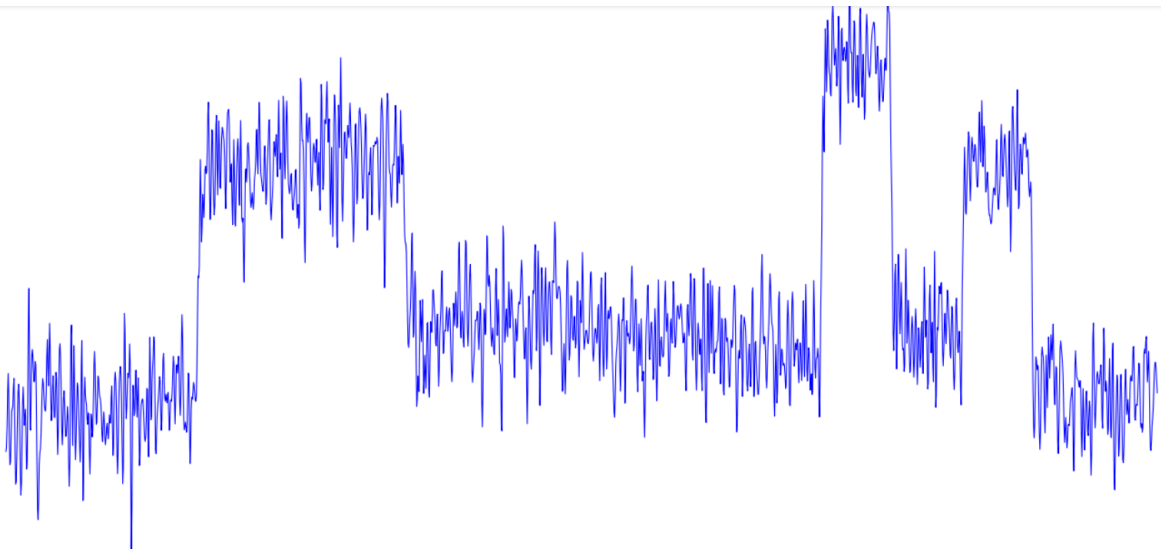
Dove  $\omega$  è un parametro che in base al suo valore rende la sensibilità alla standard deviation più alta o bassa.

Se la somma  $S_H$ , continua a rimanere sotto una soglia (definita da me che bisogna settare bene), allora le varie  $Z_t$  continuano a essere calcolate in base alla media e std della finestra iniziale.

Se invece si supera la soglia, si aggiorna la media e standard deviation con la finestra che parte dal changepoint in avanti di 100 elementi. E così via.

## Test sui dati: dati d'esempio facile

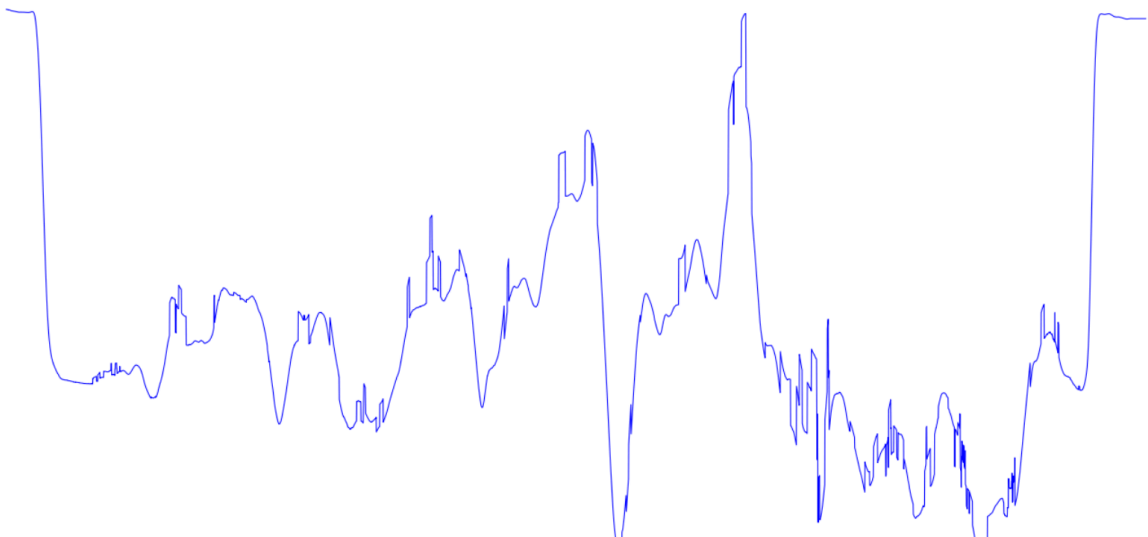
---



Il dataset in questione non è preso dai video delle danzatrici ma è un dataset di test, molto facile, perchè si vede ad occhio nudo dove sono i changepoint.

Ho iniziato a lavorare su questo per vedere se riuscissi a trovare i changepoint su questo dataset

## Test sui dati: energia cinetica del chest



Questo invece è la energia cinetica del chest riguardo il video di marianne t\_014. Il motivo per cui ho preso la energia cinetica, è dato dal fatto che ho pensato che vedere ad occhio nudo, un punto che cambia la sua velocità, può essere un change point, come per esempio una persona che si muove ad una **velocità** costante e poi aumenta la sua velocità, poi la diminuisce etc.

## Piano di lavoro

Ora sto “settando” i vari parametri per il changepoint e una volta ottenuto dei valori soddisfacenti, devo applicare dei filtri passa basso, per pulire i dati della energia cinetica per avere un **grafico** più “pulito” e poi generare i miei output e poi utilizzare ELAN Annotation e confrontare ad occhio se i changepoint che ho individuato io, sono vicini ai changepoint definiti da Eleonora.

Il confronto non lo faccio a codice, ma preferisco farlo ad occhio per riuscire a vedere meglio se i miei changepoint sono in un range vicino al changepoint definito nel groundtruth.

Una volta ottenuto valori soddisfacenti, il passo successivo è quello di utilizzare Elan Annotation per generare il groundtruth per le nuove features che sono state definite col prof. Camurri. Una volta generato il groundtruth, utilizzare il CUSUM sulle nuove features e poter valutare l'efficienza del CUSUM sulle altre features. Comparando l'output del CUSUM al nuovo groundtruth