

BDM – approfondimento sperimentale – fase 2

Obiettivo: prendere confidenza con

- riduzione della dimensionalità,
- apprendimento non supervisionato, e
- serie temporali.

Costruiremo sui risultati conseguiti nella fase 1 della sperimentazione.

Considerando i dati messi a vostra disposizione:

- Task 0a: Scrivere un programma che crei un dizionario di “gesture words”:

Data una cartella di gesti contenente le sottocartelle X, Y, Z, e W, una lunghezza di finestra w , una lunghezza di shift s , e una risoluzione r ,

1. per ciascun file di dati, f_i , il programma crea un file $f_i.wrd$ contenente quanto segue:

(a) per ciascuna componente $c_i \in \{X, Y, Z, W\}$

i. scrive in output l’identificatore della componente, c_i ;

ii. per ciascun sensore s_j di f_i nella componente c_i ,

A. scrive nel file di output l’identificatore del sensore s_j ,

B. calcola e scrive nel file di output, l’ampiezza media, $avg_{i,j}$, dei valori,

C. calcola e scrive nel file di output, le deviazioni standard, $std_{i,j}$, of the values,

D. normalizza i valori riconducendoli all’intervallo tra -1.0 e 1.0 (se i valori sono tutti uguali, allora vengono normalizzati a 0.0)

E. quantizza i valori in $2r$ livelli, come descritto della Fase 1 del progetto,

F. sposta una finestra di lunghezza w sulla corrispondente serie temporale (facendo scorrere in avanti l’inizio della finestra di s istanti temporali per volta), e alla posizione h

G. calcola e scrive nel file di output, l’ampiezza quantizzata media, $avgQ_{i,j,h}$, per la finestra h del sensore s_j

H. calcola e scrive nel file di output, il descrittore simbolico della finestra quantizzata, $winQ_{i,j,h}$ per la finestra h del sensore s_j

IMPORTANTE: Per i tasks successivi, il dizionario delle parole consiste di h triple $\langle Component_Name, sensor ID, winQ \rangle$

- Task 0b: Scrivere un programma che, data una directory, dir , associ a ciascun sensore s_j in ciascun file di gesti f_i due vettori di gesti, sulla base dei

– valori TF (i risultati per tutti i sensori per il dato gesto sono scritti in un unico file di output, $tf_vectors_fi.txt$)

– valori TF-IDF (i risultati per tutti i sensori per il dato gesto sono scritti in un unico file di output, $tfidf_vectors_fi.txt$)

- Task 1 Scrivere un programma che, dato un insieme di file di gesti, un modello vettoriale scelto dall’utente, e un valore k , identifichi e restituisca le top- k semantiche latenti/topics, utilizzando

– opzione #1: PCA,

– opzione #2: SVD,

– opzione #3: LDA.

Potete usare i packages Matlab per calcolare queste semantiche. I topic latenti vanno presentati nella forma di coppie $\langle parola, score \rangle$ ordinate secondo l’ordine non crescente degli score.

- Task 2: Scrivere un programma che, dato un file di gesti, individui i 10 gesti piu’ simili nel database di gesti, utilizzando:

– opzione #1: dot product di vettori di gesti,

– opzione #2, 3, 4: top- k semantiche latenti dei sensori (PCA, SVD, LDA),

– opzione #5: edit distance sui descrittori simbolici delle finestre quantizzati (proporre una funzione di costo di edit tra i simboli),

– user option #6: distanza DTW sulle ampiezze quantizzate medie

I risultati vanno presentati nella forma di coppie $\langle gesture, score \rangle$ ordinate secondo l’ordine non crescente degli score di somiglianza.

- Task 3: scoperta di Gesture Latenti: scrivere un programma che, dato un valore p ,

1. crei una matrice di similarita’ gesture-gesture, utilizzando

* opzione #1: dot product di vettori dei gesti,

- * opzioni #2, 3, 4: top-k semantiche dei sensori latenti (PCA, SVD, LDA),
- * opzione #5: edit distance sui descrittori simbolici delle finestre quantizzati (proporre una funzione di costo di edit tra i simboli),
- * opzione #6: distanza DTW sulle ampiezze quantizzate medie

2. calcoli SVD su questa matrice di similarita' gesture-gesture matrix, e

3. restituisca le top-p componenti principali (in termini di appartenenza dei gesti) sottostanti questa matrice di similarita' gesture-gesture

I risultati vanno presentati nella forma di coppie <gesture, score> ordinate secondo l'ordine non crescente degli score di appartenenza.

• Task 4: Analisi e Clustering di Gesti latenti

– Task 4a: Scrivere un programma che consideri le top-p semantiche latent dei gesti ottenuti al task 3 e partizioni i gesti in p gruppi in base ai loro gradi di appartenenza alle p semantiche.

– Task 4b:

scrivere un programma che consideri la matrice di similarita' gesture-gesture ottenuta utilizzando

- * opzione #1: dot product di vettori dei gesti,
- * opzioni #2, 3, 4: top-k semantiche dei sensori latenti (PCA, SVD, LDA),
- * opzione #5: edit distance sui descrittori simbolici delle finestre quantizzati (proporre una funzione di costo di edit tra i simboli),
- * opzione #6: distanza DTW sulle ampiezze quantizzate medie

e clusterizzi i gesti in p gruppi utilizzando un algoritmo basato su k-means (preferibilmente un algoritmo implementato da voi, non da libreria).

– Task 4c: scrivere un programma che consideri la matrice di similarita' gesture-gesture ottenuta utilizzando

1. crei una matrice di similarita' gesture-gesture, utilizzando

- * opzione #1: dot product di vettori dei gesti,
- * opzioni #2, 3, 4: top-k semantiche dei sensori latenti (PCA, SVD, LDA),
- * opzione #5: edit distance sui descrittori simbolici delle finestre quantizzati (proporre una funzione di costo di edit tra i simboli),
- * opzione #6: distanza DTW sulle ampiezze quantizzate medie

e clusterizzi i gesti in p groups utilizzando una tecnica di clustering spettrale basato sulla matrice Laplaciana.

• Task 5: classificazione di gesti. Implementare:

- un algoritmo di classificazione basato su k-nearest neighbor, e
- un algoritmo di classificazione basato su decision tree

che ricevano in input un insieme di gesti etichettati ed associno un'etichetta ai rimanenti gesti del database.