

Progetto: Cross-correlazione applicata a file audio

Fabio Chiarani - VR408541

Sommario—Il seguente documento espone come è stato realizzato un algoritmo di riconoscimento delle canzoni attraverso Matlab e la funzione di cross-correlazione, la quale rappresenta la misura di similitudine di due segnali, eseguita su un numero di 250 casi generati dal prodotto cartesiano di 5 sorgenti audio e 5 rumori. L'algoritmo è stato testato con i 250 casi che variano come lunghezza audio da 1 a 10 secondi.

I. INTRODUZIONE

Nella teoria dei segnali, la cross-correlazione è una misura di similitudine di due segnali in funzione dello spostamento di uno rispetto all'altro. Questo è noto come uno scorrevole prodotto scalare. E' comune utilizzare la cross-correlazione per la ricerca di un segnale per una caratteristica. In matlab, la funzione di cross-correlazione è definita attraverso la funzione `[r, lags] = xcorr(x,y)`, dove la variabile `r` assume il risultato dell'operazione di cross-correlazione (se `x` è una matrice $N \times M$, allora `r` sarà anch'essa una matrice di valore $(2M - 1) \times N^2$). L'obiettivo del progetto è di sviluppare il codice visto a lezione sul tema in oggetto, ed in particolare si articola nei seguenti sotto-obiettivi:

- Individuare un numero di casi di studio elevato (20 almeno) dove applicare, variando i parametri del codice visto a lezione
- Grazie all'analisi di cui al punto precedente, l'idea è di capire quali condizioni di acquisizione non permettano una buona accuratezza (= numero casi giusti/numero casi totali)
- Capire come vari l'andamento dell'accuratezza al variare della lunghezza del segmento di test

II. DESCRIZIONE PROGETTO

Il progetto Matlab è suddiviso in 3 file principali: *LoadFile.m*, il cui scopo è quello di caricare i file audio in un array di celle. *SalvaFile.m*, il cui scopo è salvare in formato .m4a i casi d'uso creati. *UnioneSegnali.m*, il cui scopo è quello di eseguire il prodotto cartesiano tra 5 sorgenti audio e 5 rumori, e infine *main.m* dove vengono chiamate le funzioni definite nei precedenti file, e viene eseguita la cross-correlazione.

Nella cartella *musica* sono presenti i 5 file audio, mentre in *rumore* sono situati i 5 file rumore. La scelta dei file audio e di rumore è stata presa considerando la loro similitudine in alcuni casi (per verificare quanto potesse essere preciso l'algoritmo) e, la loro dissimilarità. Dunque, ci saranno alcuni casi in cui la musica e il rumore sono molto simili, e alcuni casi in cui il rumore discorde di tanto il sorgente audio musicale. Dentro la cartella *casi* sono presenti i casi generati dal prodotto cartesiano dei sorgenti musica e rumore, descritti nel seguente formato: *Caso_Mx_Ry.m4a*, dove `x` corrisponde al

numero del file musicale originale e, `y` corrisponde al rumore che è stato sommato. L'algoritmo verrà in seguito eseguito per 10 volte, creando dei casi d'uso di lunghezza differente, creando i casi di lunghezza da 1 secondo a 10 secondi.

Durante la fase di somma dei segnali, si controlla che il rumore e il sorgente audio abbiano la stessa frequenza, altrimenti, il caso viene scartato.

Dopo aver caricato i file audio, sommati ai rumori per creare i casi d'uso, e salvati, viene eseguito l'algoritmo di cross-correlazione. L'algoritmo confronta il caso i -esimo, (C_i , ad esempio *Caso_M1_R1.m4a*) con tutte e 5 le sorgenti musicali. Il risultato della cross-correlazione viene salvata in una struttura dati d'appoggio, dove viene salvato il valore massimo ritornato dalla funzione `xcorr`. Dopo aver confrontato il caso C_i con tutte le sorgenti audio musicali, viene mostrato in output, secondo l'algoritmo, quale match tra caso e sorgente audio originale è stato scelto.

L'algoritmo è il seguente:

```
for i_e = 1 : length(esempi)
    risultati = [];
    for i_m = 1 : length(sorgentiMusica)
        [r, lags] = xcorr(esempi{i_e, 1},
                        sorgentiMusica{i_m, 1}(:,1));
        maxValue = max(r);
        risultati = [risultati; {i_e, i_m,
                                maxValue}];
    end
end

findMax = 1;
index = 1;
for k = 1 : length(sorgentiMusica)
    if(risultati{k, 3} > findMax)
        index = k;
        findMax = risultati{k, 3};
    end
end
```

Un esempio di output è il seguente:

```
Processato XCORR Caso_M5_R1
<--> Musica originale: 1
<--> Max: 8.464793e+02
Processato XCORR Caso_M5_R1
<--> Musica originale: 2
....
Processato XCORR Caso_M5_R1
<--> Musica originale: 5
<--> Max: 1.248917e+04
MATCH: Caso_M5_R1
<--> Musica originale: 5,
max: 1.248917e+04
```

III. RISULTATI OTTENUTI

1) Con 1s di taglio:

- Casi: 25
- Corretti: 15
- Errati: 10
- **Rapporto: 60% corretto**
 - Caso_M1_X: 5/5
 - Caso_M2_X: 0/5
 - Caso_M3_X: 0/5
 - Caso_M4_X: 5/5
 - Caso_M5_X: 5/5

2) Con 2s di taglio:

- Casi: 25
- Corretti: 15
- Errati: 10
- **Rapporto: 60% corretto**
 - Caso_M1_X: 5/5
 - Caso_M2_X: 0/5
 - Caso_M3_X: 0/5
 - Caso_M4_X: 5/5
 - Caso_M5_X: 5/5

3) Con 3s di taglio:

- Casi: 25
- Corretti: 16
- Errati: 9
- **Rapporto: 64% corretto**
 - Caso_M1_X: 5/5
 - Caso_M2_X: 0/5
 - Caso_M3_X: 1/5
 - Caso_M4_X: 5/5
 - Caso_M5_X: 5/5

4) Con 4s di taglio:

- Casi: 25
- Corretti: 16
- Errati: 9
- **Rapporto: 64% corretto**
 - Caso_M1_X: 5/5
 - Caso_M2_X: 0/5
 - Caso_M3_X: 1/5
 - Caso_M4_X: 5/5
 - Caso_M5_X: 5/5

5) Con 5s di taglio:

- Casi: 25
- Corretti: 16
- Errati: 9
- **Rapporto: 64% corretto**
 - Caso_M1_X: 5/5
 - Caso_M2_X: 0/5
 - Caso_M3_X: 1/5
 - Caso_M4_X: 5/5
 - Caso_M5_X: 5/5

6) Con 6s di taglio:

- Casi: 25
- Corretti: 17
- Errati: 8

• Rapporto: 68% corretto

- Caso_M1_X: 5/5
- Caso_M2_X: 0/5
- Caso_M3_X: 2/5
- Caso_M4_X: 5/5
- Caso_M5_X: 5/5

7) Con 7s di taglio:

- Casi: 25
- Corretti: 19
- Errati: 6
- **Rapporto: 75% corretto**
 - Caso_M1_X: 5/5
 - Caso_M2_X: 1/5
 - Caso_M3_X: 3/5
 - Caso_M4_X: 5/5
 - Caso_M5_X: 5/5

8) Con 8s di taglio:

- Casi: 25
- Corretti: 21
- Errati: 4
- **Rapporto: 84% corretto**
 - Caso_M1_X: 5/5
 - Caso_M2_X: 1/5
 - Caso_M3_X: 5/5
 - Caso_M4_X: 5/5
 - Caso_M5_X: 5/5

9) Con 9s di taglio:

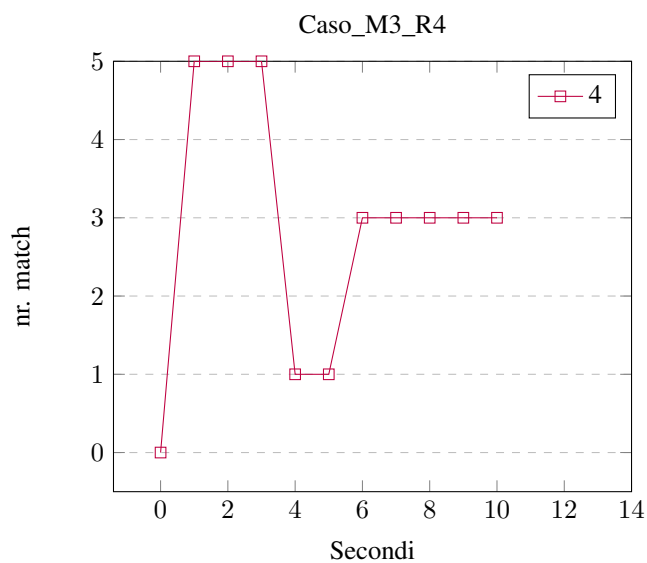
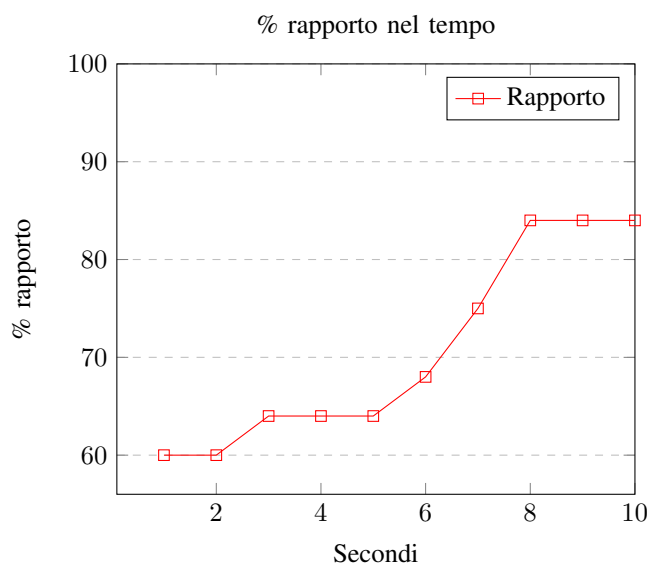
- Casi: 25
- Corretti: 21
- Errati: 4
- **Rapporto: 84% corretto**
 - Caso_M1_X: 5/5
 - Caso_M2_X: 1/5
 - Caso_M3_X: 5/5
 - Caso_M4_X: 5/5
 - Caso_M5_X: 5/5

10) Con 10s di taglio:

- Casi: 25
- Corretti: 21
- Errati: 4
- **Rapporto: 84% corretto**
 - Caso_M1_X: 5/5
 - Caso_M2_X: 1/5
 - Caso_M3_X: 5/5
 - Caso_M4_X: 5/5
 - Caso_M5_X: 5/5

IV. CONCLUSIONI

Si può notare che i test dimostrano che il variare della lunghezza del caso, aumenta la precisione del match. Difatti, il rapporto con 1 secondo di taglio è del 60% rispetto al 84% ottenuto con 10 secondi di taglio, in linea con ciò che ci si aspettava dalla corss-correlazione.



Variazione match caso 3 con rumori, rispetto alla lunghezza

