



UNIVERSITÀ DEGLI STUDI DI MILANO

UNIVERSITÀ DEGLI STUDI DI MILANO STATALE

STEFANO LOCATI

PROGETTO KNN-ALGORITHM 2020-2021

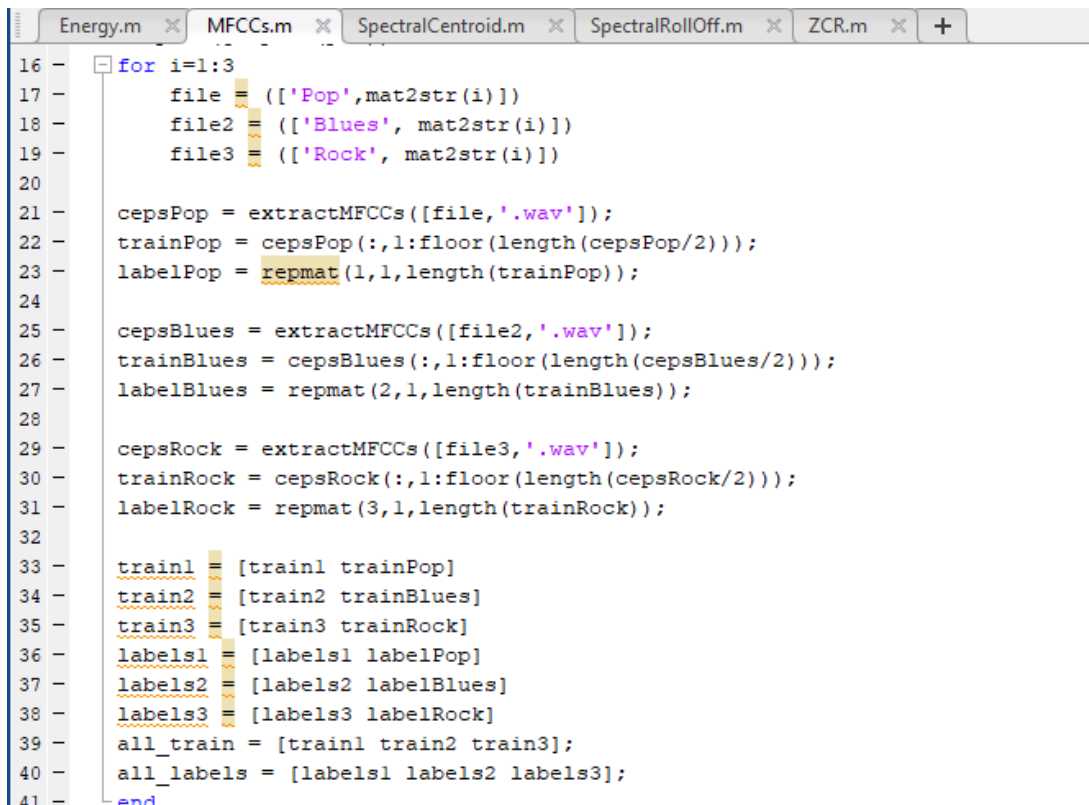
Il progetto che ho sviluppato è strutturato in base alle funzioni delle features fornite nei vari laboratori. Ho preso i file contenenti le funzioni e ne ho creato uno per ogni funzione chiamato “extract + namefunction”. Ho adattato i file extract in modo tale che funzionassero nei file relativi ad ogni feature.

Il progetto prende in analisi tre generi a scelta e per ogni genere analizza sei brani. I generi che ho scelto di analizzare sono: Blues, Pop, Rock

Il file principale prevede che vengano prese le features relative a tre brani di un determinato genere e che vengano utilizzate come Train, mentre le features relative agli altri tre brani di quel genere vengano utilizzate come Test.

L’algoritmo kNN misura la performance, calcolando la percentuale di predizioni corrette e confrontando i risultati con i dati di partenza. Utilizzo questo algoritmo per il riconoscimento di determinati pattern per poi classificare degli oggetti basandosi sulle features degli oggetti vicini a quello considerato.

Le canzoni vengono analizzate per circa quaranta secondi, così da non occupare troppo tempo nell’analisi dell’intero brano. Per procedere all’analisi delle features dei pezzi, ho deciso di denominare ogni file musicale nel seguente modo: “NomeGenere NumeroBrano .wav”. All’interno del file principale di ogni caratteristica viene fatto un ciclo che prende ogni brano solamente cambiando il NumeroBrano. In questo modo, il codice si posiziona nella cartella “Canzoni” e attraverso il ciclo, prende di volta in volta la canzone di cui ha bisogno.



```
16 - for i=1:3
17 -     file = ([ 'Pop', mat2str(i) ])
18 -     file2 = ([ 'Blues', mat2str(i) ])
19 -     file3 = ([ 'Rock', mat2str(i) ])
20 -
21 -     cepsPop = extractMFCCs([file, '.wav']);
22 -     trainPop = cepsPop(:, 1:floor(length(cepsPop)/2));
23 -     labelPop = repmat(1, 1, length(trainPop));
24 -
25 -     cepsBlues = extractMFCCs([file2, '.wav']);
26 -     trainBlues = cepsBlues(:, 1:floor(length(cepsBlues)/2));
27 -     labelBlues = repmat(2, 1, length(trainBlues));
28 -
29 -     cepsRock = extractMFCCs([file3, '.wav']);
30 -     trainRock = cepsRock(:, 1:floor(length(cepsRock)/2));
31 -     labelRock = repmat(3, 1, length(trainRock));
32 -
33 -     train1 = [train1 trainPop]
34 -     train2 = [train2 trainBlues]
35 -     train3 = [train3 trainRock]
36 -     labels1 = [labels1 labelPop]
37 -     labels2 = [labels2 labelBlues]
38 -     labels3 = [labels3 labelRock]
39 -     all_train = [train1 train2 train3];
40 -     all_labels = [labels1 labels2 labels3];
41 - end
```

Nella foto qui sopra viene mostrato il ciclo (In questo caso per l'estrazione dell'MFCCs). Ho creato dei vettori inizialmente vuoti: train1, train2, train3, labels1, labels2, labels3. Successivamente, li ho uniti tutti in all_train e all_labels. Lo stesso procedimento è stato fatto nella fase di test. Il codice poi procede come nell'esempio svolto durante le lezioni, in particolare nel Laboratorio 8. Il codice relativo al kNN algorithm, infatti, non è stato modificato.

```

69
70
71 - k=[1 5 10 15 20];
72 - rate = [];
73 - for kk=1:length(k)
74 -     disp(['set-up the kNN... number of neighbors: ',mat2str(k(kk))])
75 -     Mdl = fitcknn(all_train',all_labels','NumNeighbors',k(kk));
76
77     % test the kNN
78 -     predicted_label = predict(Mdl,all_test');
79
80     % measure the performance
81 -     correct = 0;
82 -     for i=1:length(predicted_label)
83 -         if predicted_label(i)==correct_label(i)
84 -             correct=correct+1;
85 -         end
86 -     end
87 -     disp('recognition rate:')
88 -     rate(kk) = (correct/length(predicted_label))*100
89 - end
90
91 - [a,b]=max(rate);
92 - disp('-----results-----')
93 - disp(['the maximum recognition rate is ',mat2str(a)])
94 - disp(['and it is achieved with ',mat2str(k(b)), ' nearest neighbors'])

```

Infine, viene mostrato attraverso tre “disp” il risultato ottenuto dall'algoritmo e in particolare viene mostrato il tasso di massimo riconoscimento con il numero di nearest neighbors con il quale esso è stato ottenuto.

```

Command Window

rate =

    49.4815    50.5715    50.2445    51.7331

set-up the kNN... number of neighbors: 20
recognition rate:

rate =

    49.4815    50.5715    50.2445    51.7331    51.4061

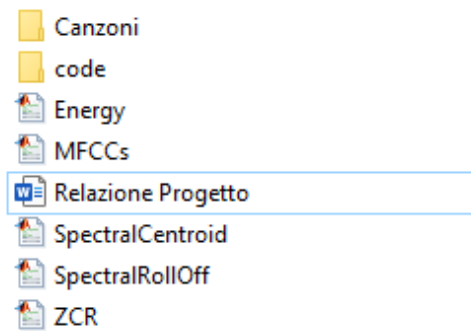
-----results-----
the maximum recognition rate is 51.7330509794152
and it is achieved with 15 nearest neighbors
fx >>

```

Nella foto viene mostrato il risultato relativo a MFCCs

Il procedimento sopra elencato è lo stesso per ogni feature. La differenza sta nei file relativi alle funzioni di ogni feature i quali prendono spunto dai codici forniti dal professore durante i vari laboratori.

Sono state implementate le seguenti features: Energy, MFCCs, Spectral Centroid, Spectral Roll off, Zero-crossing rate.



Nella cartella “Canzoni” sono presenti i file in formato .wav di tutti i 18 brani scelti. Nella cartella “code” sono presenti i file delle funzioni utili per estrarre le varie features da un file audio. Infine, nella cartella principale sono presenti i cinque file principali per l'estrazione e la visualizzazione del risultato finale.