



MATLAB

FEATURES

Utilities – Sort array and retrieve indexes

sort

Sort array elements

co

Syntax

```
B = sort(A)
B = sort(A,dim)
B = sort(__,direction)
B = sort(__,Name,Value)
[B,I] = sort(__)
```

Description

B = **sort**(**A**) sorts the elements of **A** in ascending order.

- If **A** is a vector, then **sort**(**A**) sorts the vector elements.
- If **A** is a matrix, then **sort**(**A**) treats the columns of **A** as vectors and sorts each column.
- If **A** is a multidimensional array, then **sort**(**A**) operates along the first array dimension whose size does not equal 1, treating the elements as vectors.

B = **sort**(**A**,**dim**) returns the sorted elements of **A** along dimension **dim**. For example, if **A** is a matrix, then **sort**(**A**,2) sorts the elements of each row.

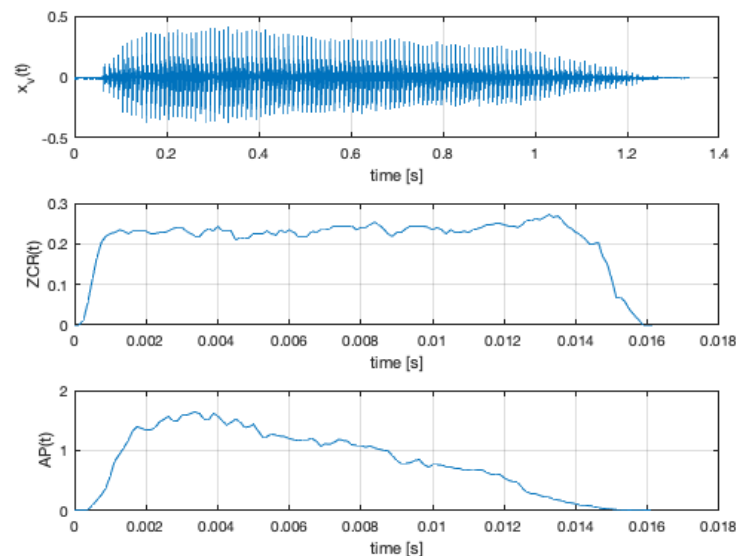
B = **sort**(__ ,**direction**) returns sorted elements of **A** in the order specified by **direction** using any of the previous syntaxes. 'ascend' indicates ascending order (the default) and 'descend' indicates descending order.

B = **sort**(__ ,**Name**,**Value**) specifies additional parameters for sorting. For example, **sort**(**A**, 'ComparisonMethod', 'abs') sorts the elements of **A** by magnitude.

[B,I] = **sort**(__) also returns a collection of index vectors for any of the previous syntaxes. **I** is the same size as **A** and describes the arrangement of the elements of **A** into **B** along the sorted dimension. For example, if **A** is a vector, then **B** = **A**(**I**).

Exercise 1 – Feature Extraction

- Load the file 'voiced.wav'
- Define a Hamming window of length 40 ms
- Extract ZCR and AP from each window of the audio signal, using hop size of 10 ms
 - Extract a 50ms audio excerpt
 - Apply Hamming window
 - Compute ZCR
 - Compute AP
 - Save feature vector [ZCR, AP]
 - Repeat for all windows...
- Plot the following figures
 - The audio waveform
 - The ZCR in time
 - The AP in time



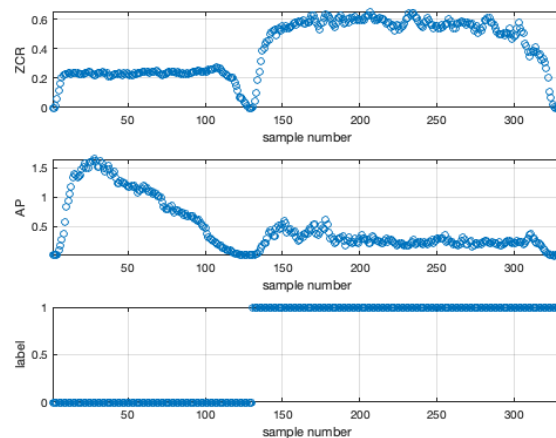
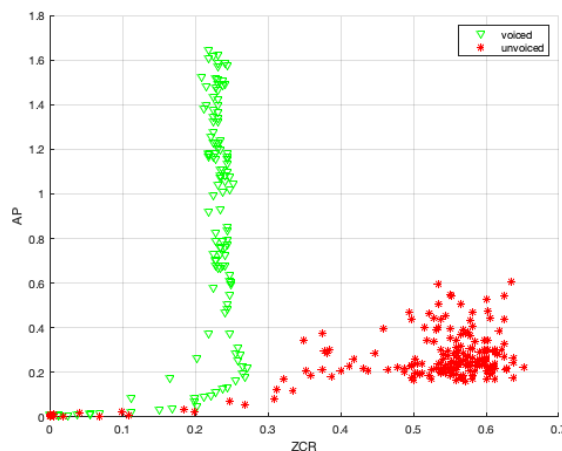
Exercise 2 – KNN Classification

Develop a *voiced* vs. *unvoiced* sound classification system

- Load the three audio signals (voiced, unvoiced, and test one)

Training:

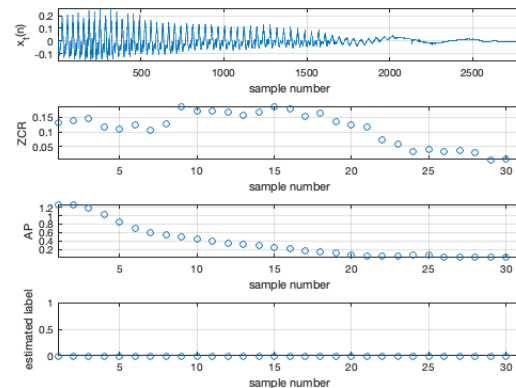
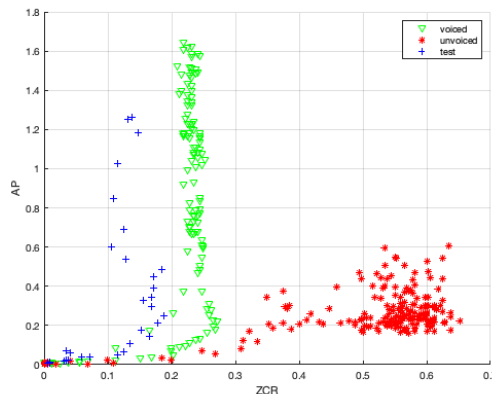
- Using the same parameters of exercise 1, extract ZCR and AP features from voiced and unvoiced signals
- Using a different color for each class, plot training feature vectors in the feature space
- Plot in three separate subplots: all training ZCR, all training AP, the corresponding label (0=voiced, 1=unvoiced)



Exercise 2 – KNN Classification

Deploy / test the system:

- Using the same parameters of exercise 1, extract ZCR and AP features from the test signal
- Plot test feature vectors in the feature space together with training features
- Using KNN with $K=3$, classify each window of the test signal separately
- Plot the test waveform, the ZCR for each window, the AP for each window, and the estimated label for each window



- Is the test sound a voiced or unvoiced one? Listen to it and check

Exercise 2 – KNN Classification

- Repeat with 'test_long.wav'

