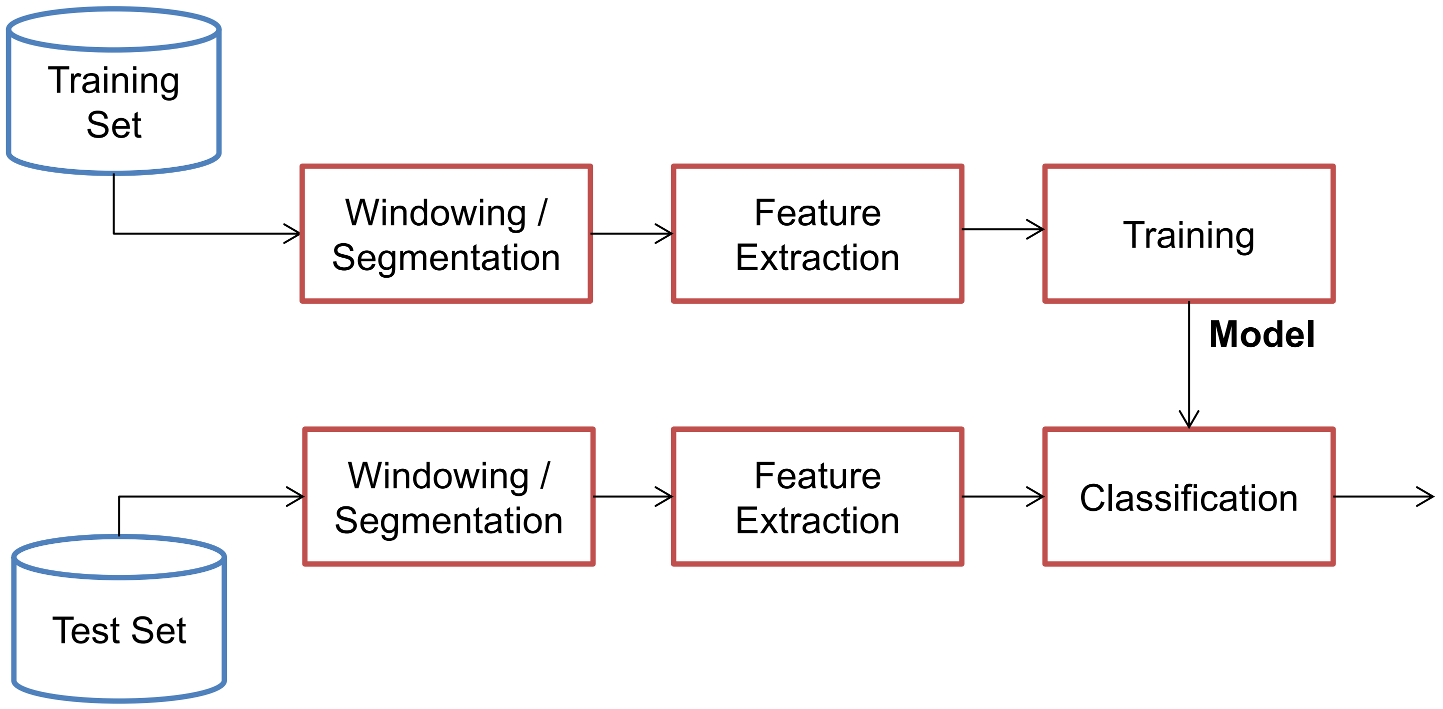
POLITECNICO
MILANO 1863

ISPL
Image and Sound Processing Lab

Classification

First Part

Classification Scheme



Windowing and segmentation

Windowing refers to the fragmentation of the signal into regularly spaced windows. Multiple consecutive windows may contain content from the same event.

Segmentation refers to the fragmentation of the signal into non-regularly spaced windows so that each segment contains a signal that is uniform. A segmentation algorithm performs a rough subdivision of the stream into events.

Feature Extraction

Extract enough representative characteristics (features) which best discriminate classes.

In our example we extract

- Zero-Crossing
- Spectral Centroid
- Spectral Decrease

Notation

t – index of time samples

$s(t)$ – input digital audio signal

F_s– sampling rate

l – index of the time frame

$s_l(t)$ – l-th frame from s(t)

$hopSize$ - time interval between successive frames

N_{hop} - number of samples between successive frames

L_w - length of a time-frame (in seconds)

N_w - number of samples in each time frame

L - number of windows in $s(t)$

k – index of the frequency bin

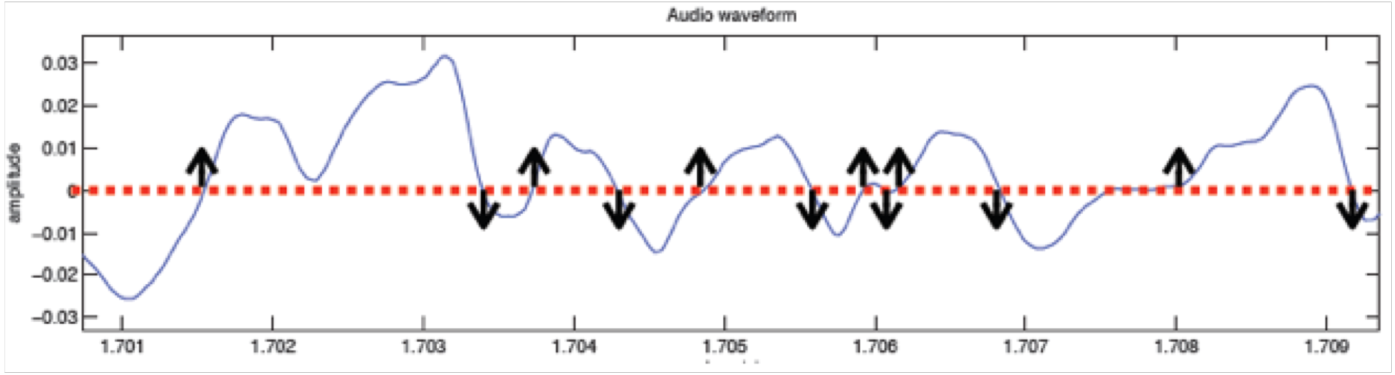
$S_l(k)$ – k-th coefficient extracted from the DFT of $s_l(t)$

$P_l(k)$ – k-th coefficient extracted from the power spectrum of $s_l(t)$

Zero Crossing Rate

The Zero Crossing Rate is computed counting the number of times that the audio waveform crosses the zero axis.

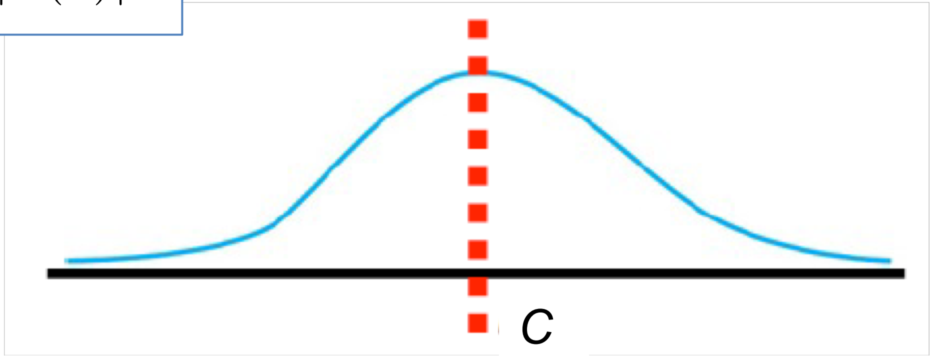
$$ZCR = \frac{1}{2}\sum_{t=1}^{N-1} |sign(t) - sign(t-1)| \frac{F_s}{N}$$



Spectral Centroid

The Spectral Centroid is the center of the mass **(**centroid**)** of the magnitude spectrum and is a measure of central tendency of the frequency distribution.

$$C = \frac{\sum_{k=0}^{K/2-1} f_k |S(k)|}{\sum_{k=0}^{K/2-1} |S(k)|}$$



Spectral Decrease

The spectral decrease aims at quantifying the amount of decrease of the amplitude spectrum. Coming from perceptive studies, it is supposed to be more correlated with human perception.

$$DEC = \frac{1}{\sum_{k=1}^{K/2-1} |S(k)|} \sum_{k=1}^{K/2-1} \frac{|S(k)| - |S(0)|}{k-1}$$

A low DEC indicates a concentration of the amplitude spectrum around the bin 0.

Classification

- Once the audio features have been extracted from the signals, they are fed to the classifiers.
- If the chosen features are relevant for the problem at hand, they will cluster in the feature space in different classes.
- Each cluster refers to a single acoustic event (class)

Goal: given a point in the feature space, estimate the class which it belongs to

Classification

Training: given a “labelled” dataset (each element in the dataset is assigned in advance to an acoustic event), estimate the parameters of one or more functions that partitions the feature space into portions, each belonging to a specific audio event.

Testing: given an “un-labelled” data and the parameters of the above function, assign it a class.

Gaussian Mixture Models

- The function is a mixture of Gaussian probability density functions (pdf’s)
- The GMM approach assumes that the density of an observed process can be modelled as a weighted sum of component densities given by:

$$p(\mathbf{x}|\lambda) = \sum_{m=1}^{M} c_m b_m(\mathbf{x}) \qquad b_m(\mathbf{x}) = \frac{1}{\sqrt[N]{2\pi}\sqrt{\det \boldsymbol{\Sigma}_m}} e^{-(\mathbf{x}-\boldsymbol{\mu}_m)^T \boldsymbol{\Sigma}_m^{-1}(\mathbf{x}-\boldsymbol{\mu}_m)}$$

$\boldsymbol{x}$: N-dimensional random vector
$b_m(\boldsymbol{x})$: multivariate Gaussian density, parameterized by its mean μ_m and the covariance matrix $\boldsymbol{\Sigma}_m$
M : number of components
c_m : weightfor them-th gaussian component

Training of GMMs

- The training of GMM’s is found as the parameter that maximizes the likelihood of the dataset.
- A useful algorithm in this sense is the Expectation Maximization (EM).

Expectation Maximization Algorithm

Expectation step: computes an expectation of the log likelihood with respect to the current estimate of the distribution for the latent variables

$$\boldsymbol{\mu}_m^{\text{new}} = \frac{\sum_i p(m|\mathbf{x}_i, \lambda)\mathbf{x}_i}{\sum_i p(m|\mathbf{x}_i, \lambda)}$$

$$\boldsymbol{\Sigma}_m^{\text{new}} = \frac{\sum_i p(m|\mathbf{x}_i - \boldsymbol{\mu}_m, \lambda)(\mathbf{x}_i - \boldsymbol{\mu}_m)^T(\mathbf{x}_i - \boldsymbol{\mu}_m)}{\sum_i p(m|\mathbf{x}_i, \lambda)}$$

$$c_m^{\text{new}} = \frac{1}{n}\sum_i p(m|\mathbf{x}_i, \lambda)$$

given the distribution we update the estimate of the parameters from it for each point of the dataset $\boldsymbol{x}_i$

Expectation Maximization Algorithm

Maximization step → computes the parameters that maximize the expected log likelihood found on the expectation step

$$p(m|\mathbf{x}_i, \lambda) = \frac{c_m b_m(\mathbf{x}_i)}{\sum_{j=1}^{M} c_j b_j(\mathbf{x}_i)}$$

given the parameters, we update the estimation of the distribution for each point in the dataset $\boldsymbol{x_i}$

Evaluation

	Relevant	Not Relevant
Retrieved	TP	FP
Not Retrieved	FN	TN

Precision = relevant/retrieved = TP/(TP+FP) is a measure of the correctness of the system

Recall = retrieved/relevant = TP/(TP+FN) is a measure of the completness of the system

F-score combines these two metrics

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2(P+R)} \qquad F_1 = 2\frac{PR}{(P+R)}$$

Accuracy = correctly classified / all = (TP + TN) / (TP + TN + FP + FN)

Hands on: packages

Create the environment: `conda create --name CMLS python=3.7`

Activate the environment: `conda activate CMLS`

Install the following packages:

```
conda install jupyter notebook

conda install scipy

conda install matplotlib

conda install scikit-learn

conda install -c conda-forge librosa

conda install seaborn
```