



Introduction to **Audio Content Analysis**

module 4.1: classification

alexander lerch

introduction

overview

corresponding textbook section

section 4.1

■ lecture content

- intuitive intro to machine learning
- classifier examples

■ learning objectives

- describe the basic principles of data-driven machine learning approaches
- implement a kNN classifier in Python



introduction

overview

corresponding textbook section

section 4.1

■ lecture content

- intuitive intro to machine learning
- classifier examples

■ learning objectives

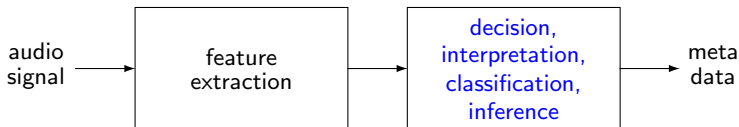
- describe the basic principles of data-driven machine learning approaches
- implement a kNN classifier in Python



classification

introduction

remember the flow chart of a general ACA system:



■ *classification:*

- assign class labels to data

■ *regression:*

- estimate numerical labels for data

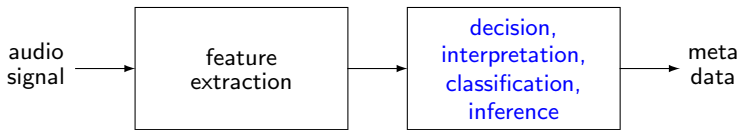
■ *clustering:*

- find grouping patterns in data

classification

introduction

remember the flow chart of a general ACA system:



■ *classification*:

- assign class labels to data

■ *regression*:

- estimate numerical labels for data

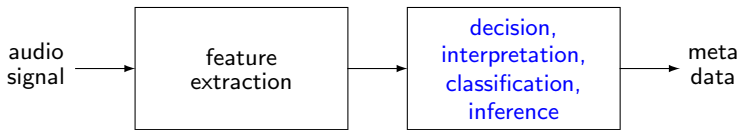
■ *clustering*:

- find grouping patterns in data

classification

introduction

remember the flow chart of a general ACA system:



■ *classification:*

- assign class labels to data

■ *regression:*

- estimate numerical labels for data

■ *clustering:*

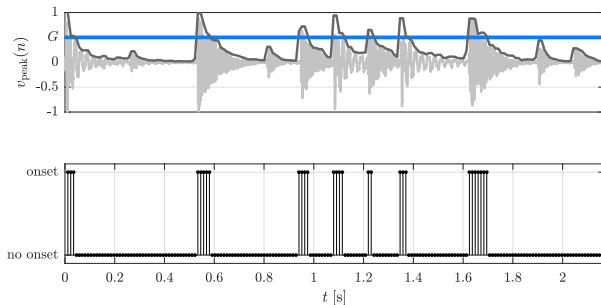
- find grouping patterns in data

classification

basic example

hypothetical system:

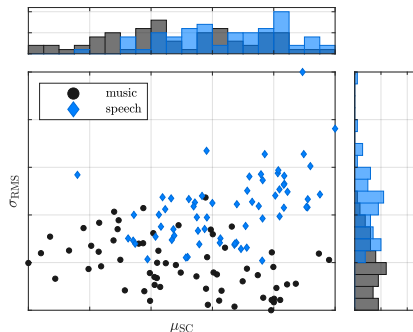
- one feature (envelope)
- predefined threshold
 - higher than threshold
⇒ class 1 (onset)
 - lower than threshold
⇒ class 0 (no onset)



classification

data-driven

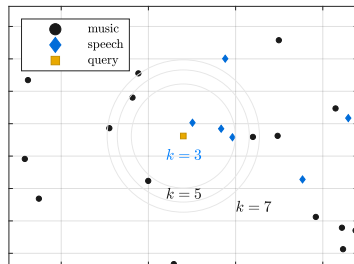
- derive classification parameters from data, e.g.,
 - ⇒ learn common feature distributions per class
 - ⇒ learn separation metrics per class



classifier examples

k-Nearest Neighbor (kNN)

- **training:** extract reference vectors from training set
 - store coordinates and class labels
- **classification:** extract query vector and set class to majority of k nearest reference vectors
 - 1 compute distance between query and all training vectors
 - 2 sort distances to find closest vectors
 - 3 choose majority class out of the k closest vectors
- **classifier model:** all training vectors

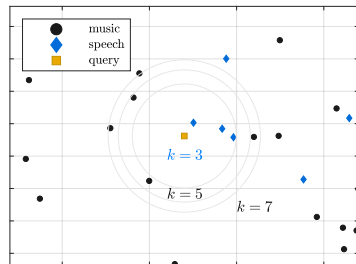


classifier examples

k-Nearest Neighbor (kNN)

- **training:** extract reference vectors from training set
 - store coordinates and class labels
- **classification:** extract query vector and set class to majority of k nearest reference vectors
 - 1 compute distance between query and all training vectors
 - 2 sort distances to find closest vectors
 - 3 choose majority class out of the k closest vectors

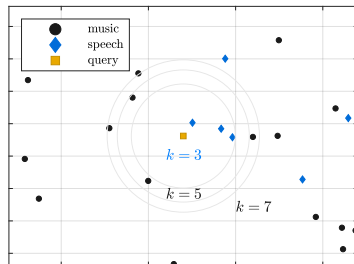
■ **classifier model:** all training vectors



classifier examples

k-Nearest Neighbor (kNN)

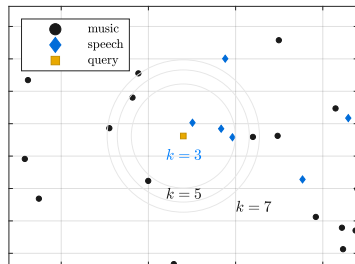
- **training:** extract reference vectors from training set
 - store coordinates and class labels
- **classification:** extract query vector and set class to majority of k nearest reference vectors
 - 1 compute distance between query and all training vectors
 - 2 sort distances to find closest vectors
 - 3 choose majority class out of the k closest vectors
- **classifier model:** all training vectors



classifier examples

k-Nearest Neighbor (kNN)

- **training:** extract reference vectors from training set
 - store coordinates and class labels
- **classification:** extract query vector and set class to majority of k nearest reference vectors
 - 1 compute distance between query and all training vectors
 - 2 sort distances to find closest vectors
 - 3 choose majority class out of the k closest vectors
- **classifier model:** all training vectors

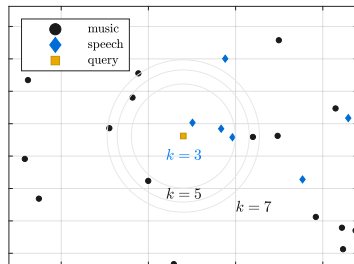


$k = 3 \Rightarrow$ blue majority

classifier examples

k-Nearest Neighbor (kNN)

- **training:** extract reference vectors from training set
 - store coordinates and class labels
- **classification:** extract query vector and set class to majority of k nearest reference vectors
 - 1 compute distance between query and all training vectors
 - 2 sort distances to find closest vectors
 - 3 choose majority class out of the k closest vectors
- **classifier model:** all training vectors

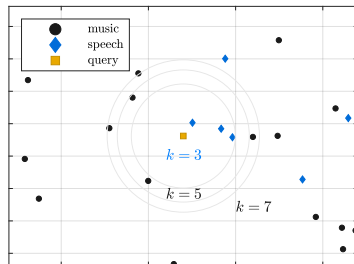


$k = 5 \Rightarrow$ black majority

classifier examples

k-Nearest Neighbor (kNN)

- **training:** extract reference vectors from training set
 - store coordinates and class labels
- **classification:** extract query vector and set class to majority of k nearest reference vectors
 - 1 compute distance between query and all training vectors
 - 2 sort distances to find closest vectors
 - 3 choose majority class out of the k closest vectors
- **classifier model:** all training vectors



$k = 7 \Rightarrow$ black majority

classifier examples

Gaussian Mixture Model (GMM)

■ training:

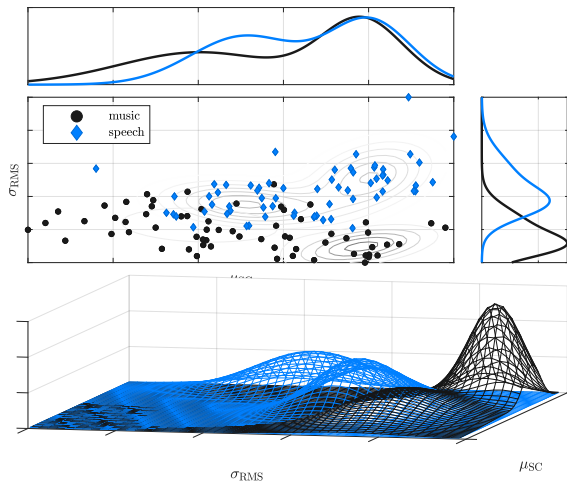
model each class distribution as superposition of Gaussian distributions

■ classification:

compute output of each Gaussian and select class with highest probability

■ classifier data:

per class per Gaussian: μ and covariance, mixture weight



classifier examples

Gaussian Mixture Model (GMM)

■ training:

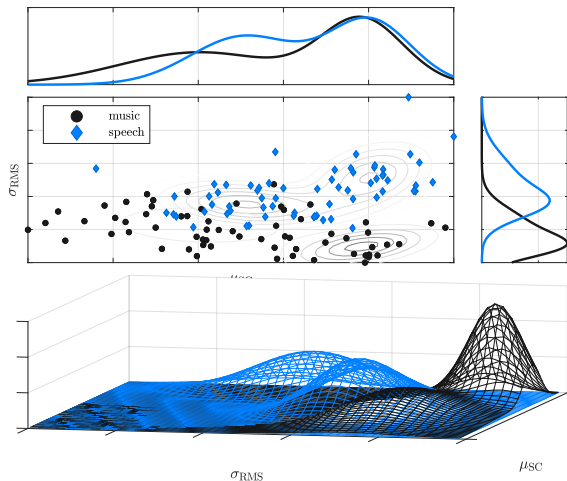
model each class distribution as superposition of Gaussian distributions

■ classification:

compute output of each Gaussian and select class with highest probability

■ classifier data:

per class per Gaussian: μ and covariance, mixture weight



classifier examples

Gaussian Mixture Model (GMM)

■ training:

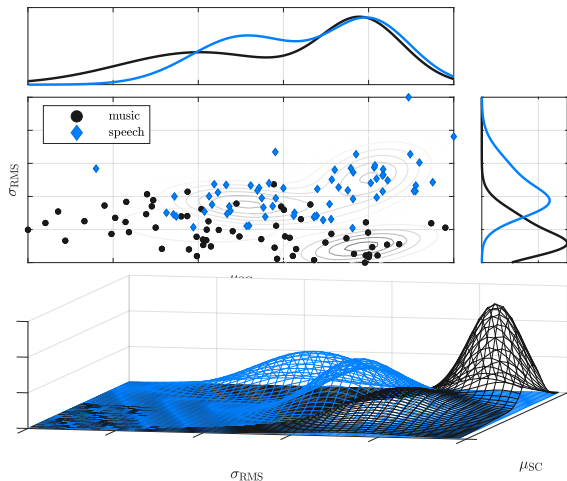
model each class distribution as superposition of Gaussian distributions

■ classification:

compute output of each Gaussian and select class with highest probability

■ classifier data:

per class per Gaussian: μ and covariance, mixture weight



classifier examples

Support Vector Machine (SVM)

■ training:

- map features to high dimensional space



- find separating hyperplane through maximum distance of support vectors (data points)

- **classification:** apply feature transform and proceed with 'linear' classification
- **classifier data:** support vectors, kernel, kernel parameters

classifier examples

Support Vector Machine (SVM)

■ training:

- map features to high dimensional space



- find separating hyperplane through maximum distance of support vectors (data points)

■ **classification:** apply feature transform and proceed with 'linear' classification

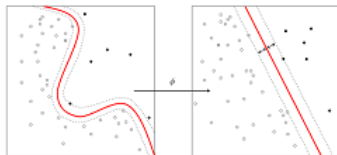
■ **classifier data:** support vectors, kernel, kernel parameters

classifier examples

Support Vector Machine (SVM)

■ training:

- map features to high dimensional space



- find separating hyperplane through maximum distance of support vectors (data points)

- **classification:** apply feature transform and proceed with 'linear' classification
- **classifier data:** support vectors, kernel, kernel parameters

summary

lecture content

■ data-driven approach

- 'general' system learns parameters/behavior from data
- human interaction through
 - ▶ parametrization and procedures
 - ▶ data selection

■ many classifiers with different levels of complexity

- 1 kNN
- 2 GMM
- 3 SVM
- 4 RandomForest
- 5 DNN
- 6 ...

