Introduction to Audio Content Analysis

Module 4.1: Classification

alexander lerch

Georgia Tech Center for Music Technology
College of Design

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

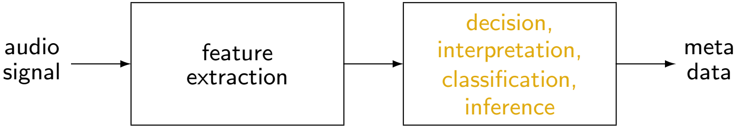
corresponding textbook section

Section 4.1

- **lecture content**
  - intuitive intro to machine learning
  - classifier examples

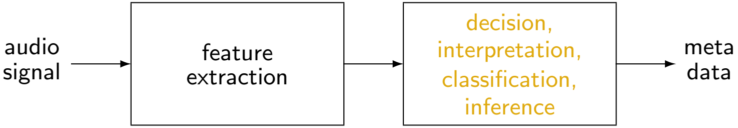
- **learning objectives**
  - describe the basic principles of data-driven machine learning approaches
  - implement a kNN classifier in Python

remember the flow chart of a general ACA system:



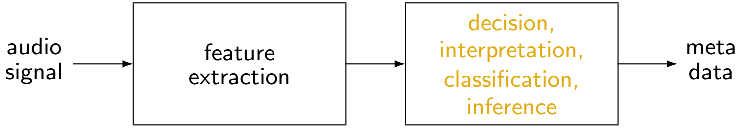
- *classification*:
  - assign class labels to data
- *regression*:
  - estimate numerical labels for data
- *clustering*:
  - find grouping patterns in data

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

remember the flow chart of a general ACA system:



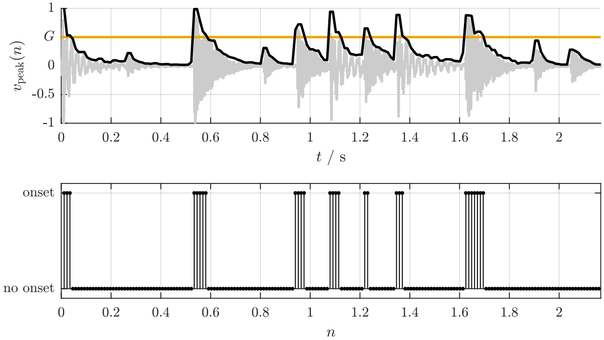
- *classification*:
  - assign class labels to data
- *regression*:
  - estimate numerical labels for data
- *clustering*:
  - find grouping patterns in data

classification

basic example

hypothetical system:

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



matlab source: plotThresholdClassification.m

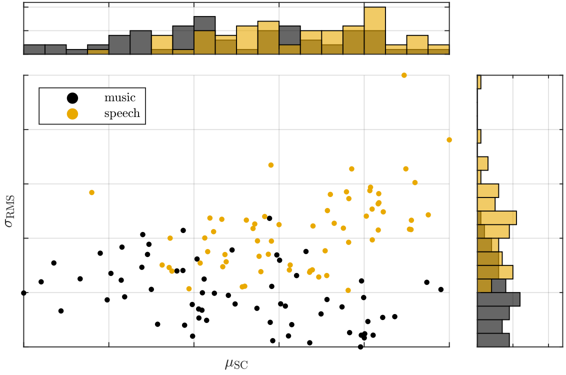
classification

data-driven

- derive classification parameters from data, e.g.,

⇒ learn common feature distributions per class

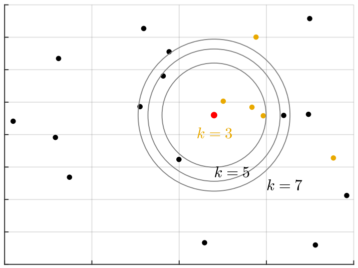
⇒ learn separation metrics per class



matlab source: plotFeatureSpace.m

classifier examples
k-Nearest Neighbor (kNN)

- **training**: extract reference vectors from training set (keep class labels)
- **classification**: extract test vector and set class to majority of k nearest reference vectors
- **classifier model**: all training vectors



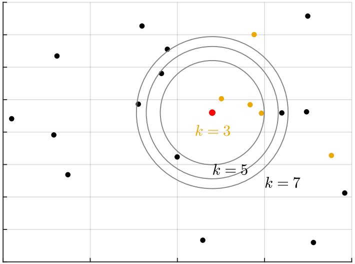
matlab source: plotKnn.m

classifier examples

k-Nearest Neighbor (kNN)

- **training**: extract reference vectors from training set (keep class labels)
- **classification**: extract test vector and set class to majority of k nearest reference vectors

- **classifier model**: all training vectors

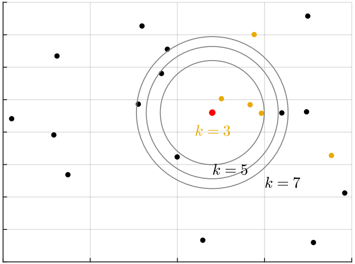


matlab source: plotKnn.m

classifier examples
k-Nearest Neighbor (kNN)

- **training**: extract reference vectors from training set (keep class labels)
- **classification**: extract test vector and set class to majority of k nearest reference vectors

- **classifier model**: all training vectors

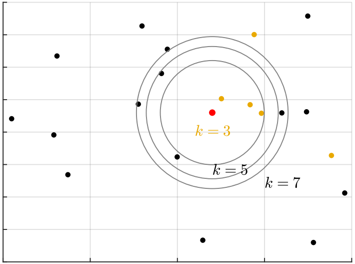


matlab source: plotKnn.m

classifier examples
k-Nearest Neighbor (kNN)

- **training**: extract reference vectors from training set (keep class labels)
- **classification**: extract test vector and set class to majority of k nearest reference vectors

- **classifier model**: all training vectors



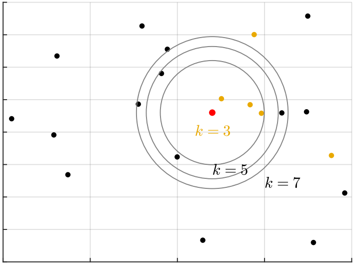
$k = 3 \Rightarrow$ gold majority

matlab source: plotKnn.m

classifier examples
k-Nearest Neighbor (kNN)

- **training**: extract reference vectors from training set (keep class labels)
- **classification**: extract test vector and set class to majority of k nearest reference vectors

- **classifier model**: all training vectors



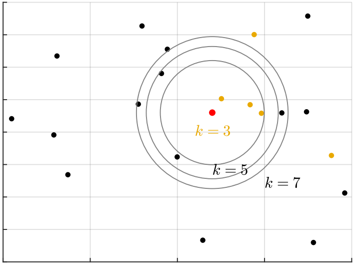
$k = 5 \Rightarrow$ black majority

matlab source: plotKnn.m

classifier examples
k-Nearest Neighbor (kNN)

- **training**: extract reference vectors from training set (keep class labels)
- **classification**: extract test vector and set class to majority of k nearest reference vectors

- **classifier model**: all training vectors



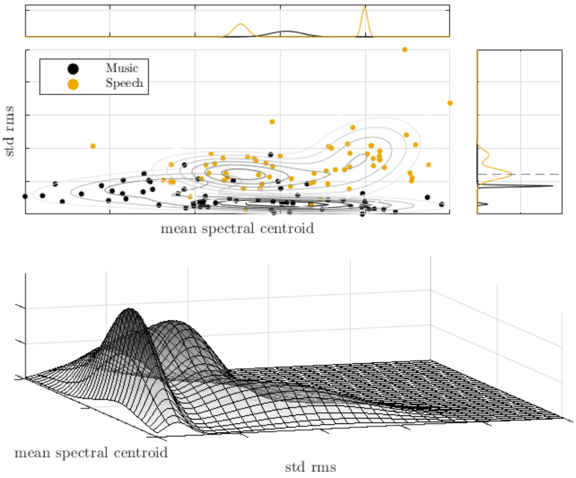
$k = 7 \Rightarrow$ black majority

matlab source: plotKnn.m

classifier examples

Gaussian Mixture Model (GMM)

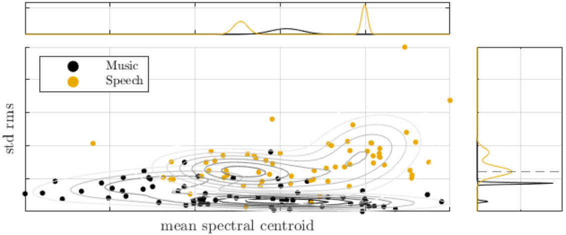
- **training**:
  model each class distribution as superposition of Gaussian distributions
- **classification**:
  compute output of each Gaussian and select class with highest probability
- **classifier data**:
  per class per Gaussian: μ and covariance, mixture weight

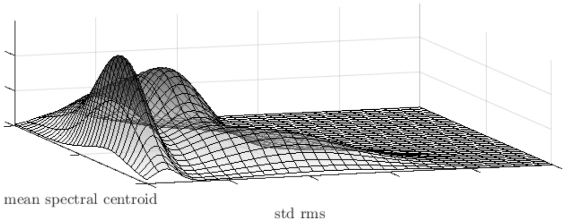


matlab source: plotGmm.m

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

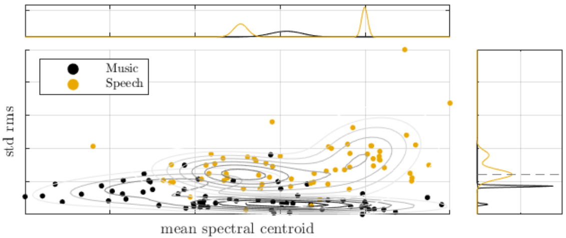


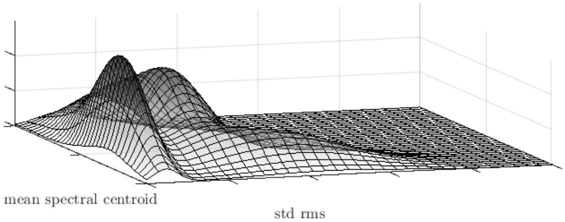


matlab source: plotGmm.m

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

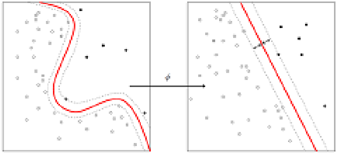




matlab source: plotGmm.m

classifier examples
Support Vector Machine (SVM)

- **training**:
  - map features to high dimensional space



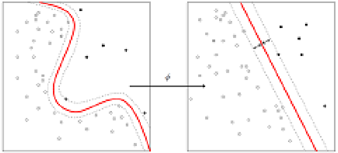
  - find separating hyperplane through maximum distance of support vectors (data points)
- **classification**: apply feature transform and proceed with 'linear' classification
- **classifier data**: support vectors, kernel, kernel parameters

https://en.wikipedia.org/wiki/Support_vector_machine

classifier examples
Support Vector Machine (SVM)

Georgia Tech | Center for Music Technology
College of Design

- **training**:
    - map features to high dimensional space

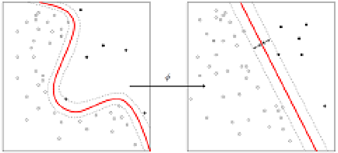


    - find separating hyperplane through maximum distance of support vectors (data points)
- **classification**: apply feature transform and proceed with 'linear' classification
- **classifier data**: support vectors, kernel, kernel parameters

https://en.wikipedia.org/wiki/Support_vector_machine

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

https://en.wikipedia.org/wiki/Support_vector_machine

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. ...