Introduction to **Audio Content Analysis**

module 3.6: learned features

alexander lerch

introduction
overview

corresponding textbook section

section 3.6

- **lecture content**
  - introduction to the concept of feature learning
  - quick overview of important methods
- **learning objectives**
  - describe the advantages and disadvantages of feature learning
  - give examples of approaches to feature learning

introduction
overview

corresponding textbook section

section 3.6

- **lecture content**
  - introduction to the concept of feature learning
  - quick overview of important methods

- **learning objectives**
  - describe the advantages and disadvantages of feature learning
  - give examples of approaches to feature learning

introduction
feature learning vs. traditional features

- **hand-crafted features**:
  - arbitrary definitions
  - simple to compute
  - mostly focus on one technical property
  - provide limited information
- **feature learning**:
  - *automatically* learn features from data-set
  - meaning not obvious, can combine multiple properties

introduction

feature learning vs. traditional features

- **hand-crafted features**:
  - arbitrary definitions
  - simple to compute
  - mostly focus on one technical property
  - provide limited information

- **feature learning**:
  - *automatically* learn features from data-set
  - meaning not obvious, can combine multiple properties

overview
general properties

- **principle**
  1. put (a lot of) raw data at input
  2. learn a way of reducing dimensionality while keeping as much information as possible

- **advantages**
  - features might contain more useful information than provided by hand-crafted features
  - no expert knowledge required

- **disadvantages**
  - usually time consuming
  - limited ways of controlling the type of information learned

overview
general properties

- **principle**
  1. put (a lot of) raw data at input
  2. learn a way of reducing dimensionality while keeping as much information as possible

- **advantages**
  - features might contain more useful information than provided by hand-crafted features
  - no expert knowledge required

- **disadvantages**
  - usually time consuming
  - limited ways of controlling the type of information learned

overview
general properties

- **principle**
  1. put (a lot of) raw data at input
  2. learn a way of reducing dimensionality while keeping as much information as possible

- **advantages**
  - features might contain more useful information than provided by hand-crafted features
  - no expert knowledge required

- **disadvantages**
  - usually time consuming
  - limited ways of controlling the type of information learned

approaches

dictionary learning

- **dictionary learning** (sparse coding, non-negative matrix factorization)

$$X = B \cdot A$$

  X: input signal to be modeled (often spectrogram)
  B: dictionary/template matrix (often set of single spectra that comprise the basic building blocks of X)
  A: activation matrix indicating the weight and superposition of templates

  - derive B,A, by minimizing a cost function, e.g. $||X - BA||_2$

$\rightarrow$ templates are trained, activations are used as feature vector (length: number of templates)

approaches
dictionary learning

- **dictionary learning** (sparse coding, non-negative matrix factorization)

$$X = B \cdot A$$

  X: input signal to be modeled (often spectrogram)
  B: dictionary/template matrix (often set of single spectra that comprise the basic building blocks of X)
  A: activation matrix indicating the weight and superposition of templates

  - derive B,A, by minimizing a cost function, e.g. $||X - BA||_2$

$\rightarrow$ templates are trained, activations are used as feature vector (length: number of templates)

approaches
deep learning

- **clustering**
  - find clusters in data set (e.g., from magnitude spectra or simple features)
  - store median of clusters (compare: template matrix)

  → features:
    - binary vector (length: number of clusters, zero except for closest cluster)
    - distance vector (distance to each cluster)

- neural networks and **deep architectures**
  - stack multiple layers of simple learning blocks
  - each layer uses the output of the previous layer as input

  → feature: output of the highest layer
  - task-dependency:
    - independent: use auto-encoder structure to maximize encoded information
    - dependent: train for specific task(s) (cf. multi-task learning) and use representation for other tasks (cf. transfer learning)

- **clustering**
  - find clusters in data set (e.g., from magnitude spectra or simple features)
  - store median of clusters (compare: template matrix)

  $\rightarrow$ features:
    - binary vector (length: number of clusters, zero except for closest cluster)
    - distance vector (distance to each cluster)

- neural networks and **deep architectures**
  - stack multiple layers of simple learning blocks
  - each layer uses the output of the previous layer as input

  $\rightarrow$ feature: output of the highest layer
  - task-dependency:
    - independent: use auto-encoder structure to maximize encoded information
    - dependent: train for specific task(s) (cf. multi-task learning) and use representation for other tasks (cf. transfer learning)

approaches
deep learning

- **clustering**
  - find clusters in data set (e.g., from magnitude spectra or simple features)
  - store median of clusters (compare: template matrix)
  - $\rightarrow$ features:
    - binary vector (length: number of clusters, zero except for closest cluster)
    - distance vector (distance to each cluster)

- neural networks and **deep architectures**
  - stack multiple layers of simple learning blocks
  - each layer uses the output of the previous layer as input
  - $\rightarrow$ feature: output of the highest layer
  - task-dependency:
    - independent: use auto-encoder structure to maximize encoded information
    - dependent: train for specific task(s) (cf. multi-task learning) and use representation for other tasks (cf. transfer learning)

approaches
deep learning

- **clustering**
  - find clusters in data set (e.g., from magnitude spectra or simple features)
  - store median of clusters (compare: template matrix)
  - → features:
    - binary vector (length: number of clusters, zero except for closest cluster)
    - distance vector (distance to each cluster)

- neural networks and **deep architectures**
  - stack multiple layers of simple learning blocks
  - each layer uses the output of the previous layer as input
  - → feature: output of the highest layer
  - task-dependency:
    - independent: use auto-encoder structure to maximize encoded information
    - dependent: train for specific task(s) (cf. multi-task learning) and use representation for other tasks (cf. transfer learning)

- **clustering**
  - find clusters in data set (e.g., from magnitude spectra or simple features)
  - store median of clusters (compare: template matrix)

  → features:
    - binary vector (length: number of clusters, zero except for closest cluster)
    - distance vector (distance to each cluster)

- neural networks and **deep architectures**
  - stack multiple layers of simple learning blocks
  - each layer uses the output of the previous layer as input

  → feature: output of the highest layer
  - task-dependency:
    - independent: use auto-encoder structure to maximize encoded information
    - dependent: train for specific task(s) (cf. multi-task learning) and use representation for other tasks (cf. transfer learning)

summary
lecture content

- **learned features**
  - not designed but learn from data
    - no hand-designed features $\Rightarrow$ no expert knowledge needed
    - requires large amounts of data and computational resources
  - not interpretable
  - potentially more powerful as it contains all relevant information