Speech Pattern Classification

A practical approach to feature extraction, machine learning and common tasks

Alberto Abad & Isabel Trancoso

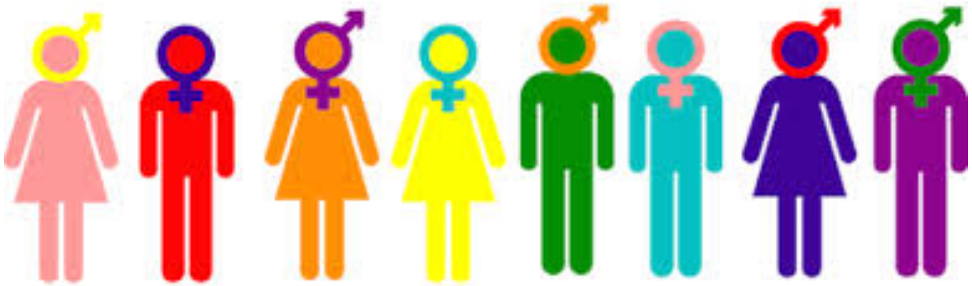
IST/INESC-ID

alberto@l2f.inesc-id.pt

Processamento de Fala - **IST** Lisboa, April, 2016

Introduction

- Speech carries a lot of information:
  - Of course information related to the message (LINGUISTIC?)...
  - ... but also, speaker traits (NON-LINGUISTIC/PARA-LINGUISTIC?):
    - Gender; Age; Language/accent; ID; Personality; Education; Intoxication; Sleepiness; Friendliness; Mood; Physical Stress; Cognitive Load; Emotion; Pathologies?



- If “Speech” is considered in a wider sense (“Audio”) then more information is present:
  - Number of speakers; speakers role; speaker position; audio events; acoustic Scenes;

Introduction

[Fujisaki and Hirose, 1993, Analysis and perception of intonation expressing paralinguistic information in spoken Japanese]

- **Linguistic information** - information that is explicitly in or almost uniquely inferable from the written message
- **Paralinguistic information** - information that is not inferable from the written message, but is added by the speaker to modify or complement the linguistic information (expressing different intentions, attitudes, speaking styles)
- **Nonlinguistic information** - information about other factors such as age, gender, idiosyncrasy, physical and emotional conditions of the speakers which are not related to the linguistic contents of the message and cannot be controlled by the speaker

Introduction

Wouldn't it be dreamy if only we could extract all these information automatically!?!? Then, we could imitate human behavior (IA); or even augment capabilities (data mining); or...

YES, we can!!! (more or less)

→ **SPEECH PATTERN CLASSIFICATION**

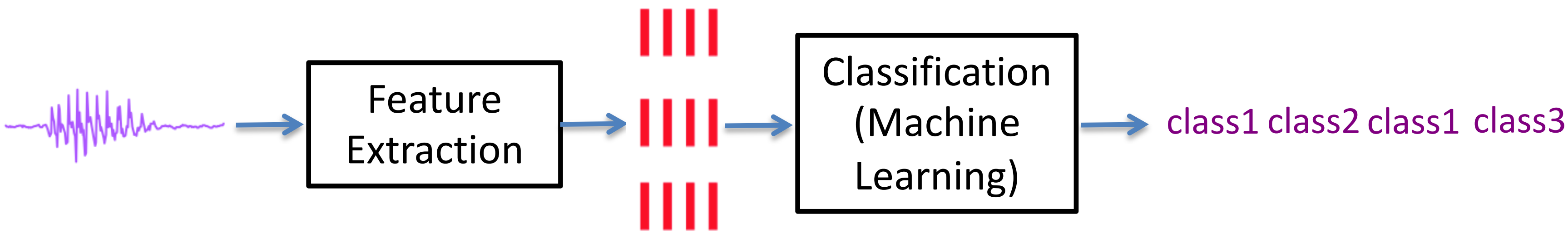
Introduction

- **Speech (audio) pattern classification (**a.k.a. Speech/audio mining) is a multidisciplinary topic, that involves areas such as:
  - Phonetics;
  - Physics/Acoustics;
  - Signal processing;
  - Algebra;
  - Probability theory;
  - Computer science;
  - Machine learning;
  - ...

Introduction

Commonalities in speech pattern classification tasks

- The objective of speech pattern classification is to convert a speech input sequence into a sequence of class labels:



- The common blocks of any speech pattern classification task are the front-end/feature extraction and the back-end/classification:
  - The classifier modules is "learnt" using data during the training phase and used to classify new unseen data during test

Introduction

- It is “more or less” a “dreamy” solution because it presents many challenges:
  - Speech/audio variability → Audio samples belonging to the same “class” can take extremely different forms due to:
    - Source variation: speaker, gender, accent, state, volume, etc.
    - Channel variation: microphone, acoustic environment, noise, reverberation, etc.
    - Other: Intrinsic nature of the classes, etc.

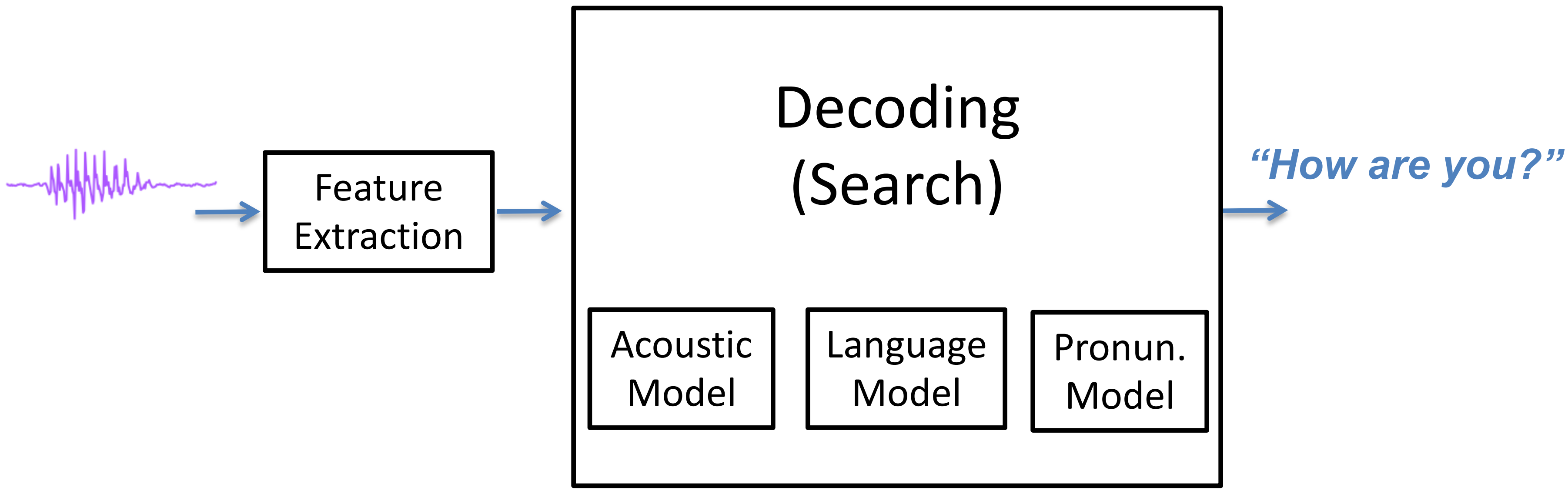
Introduction

- From the machine learning perspective, speech is a quite unique problem due to the nature of the data input and class label outputs:
  - About the input:
    - Time sequence
      - Very different length of the input wrt. output → Segmentation problem
      - Elasticity of the temporal dimension
    - Discriminative cues often distributed over a reasonably long temporal span
  - About the output:
    - Output may consist of a sequence of class labels
      - Too much combinations → Need structure

Introduction

A complex example: Automatic speech recognition (ASR)?

- **Goal** Given a sequence of observations determine which is the most likely sequence of words



- Already decades of research on ASR (and other SLT related topics) → **Very challenging!!!**

Introduction

Focus on "simple" speech classification

- In this part of the course we will focus in "simpler" speech pattern classification tasks:
  - Static output:
    - No sequence of output labels
    - No segmentation problem → An audio segment corresponds to single class
  - No structured knowledge → Models correspond to output labels

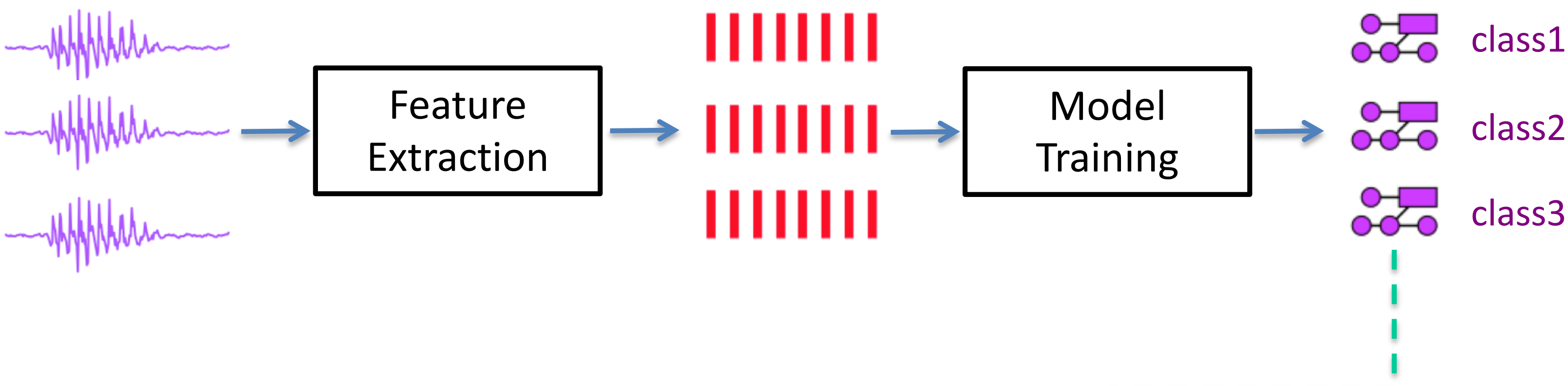
- Focus on non-linguistic (non ASR) tasks

- Notice that the addressed tasks:
  - Although being "simpler" from the ML perspective, they can be very hard
  - They are based on pre-trained models:
    - Some speech mining problems do not necessarily rely on pre-trained models (ex: speaker diarization)
  - Can be classification/identification, verification or regression problems

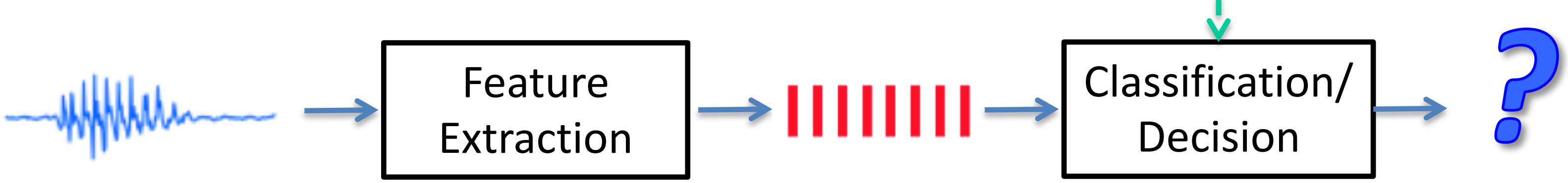
Introduction

Commonalities in the “simple” speech pattern classification

- Learning/Training phase



- Classification phase



Introduction

What about the input time nature?

- In the “simple” speech classification 1 speech/audio segment corresponds to 1 label class, but still the input is time-varying!!!
  - Remember speech analysis and typical semi-stationary properties of speech

- How to deal with this problem... **IDEAS!?**
  - Feature-level?
  - Model-level?
  - Output-level?

Introduction

Relevant Paralinguistic/Non-linguistic challenges

NIST Evaluations

NIST Speaker Recognition Evaluation (SRE)

http://www.nist.gov/itl/iad/mig/sre.cfm

NIST Language Recognition Evaluation (LRE)

http://www.nist.gov/itl/iad/mig/lre.cfm

Introduction

Relevant Paralinguistic/Non-linguistic challenges

COMPARE (Computational Paralinguistic Evaluation) Challenge Series

http://compare.openaudio.eu/

INTERSPEECH 2016 Computational Paralinguistics Challenge (on-going)

Deception Sub-Challenge

Sincerity Sub-Challenge

Native Language Sub-Challenge

Introduction

Relevant Paralinguistic/Non-linguistic challenges

COMPARE Challenge Series

INTERSPEECH 2015 Computational Paralinguistics Challenge:

- *The Degree of Nativeness Sub-Challenge*
- *The Parkinson's Condition Sub-Challenge*
- *The Eating Condition Sub-Challenge*

INTERSPEECH 2014 Computational Paralinguistics Challenge:

- *The Cognitive Load Sub-Challenge*
- *The Physical Load Sub-Challenge*

INTERSPEECH 2013 Computational Paralinguistics Challenge:

- *The Social Signals Sub-Challenge*
- *The Conflict Sub-Challenge*
- *The Emotion Sub-Challenge*
- *The Autism Sub-Challenge*

Introduction

Relevant Paralinguistic/Non-linguistic challenges

COMPARE Challenge Series

INTERSPEECH 2012 Speaker Trait Challenge:

- *The Personality Sub-Challenge*
- *The Likability Sub-Challenge*
- *The Pathology Sub-Challenge*

INTERSPEECH 2011 Speaker State Challenge:

- *The Intoxication Sub-Challenge*
- *The Sleepiness Sub-Challenge*

INTERSPEECH 2010 Paralinguistic Challenge:

- *The Age Sub-Challenge*
- *The Gender Sub-Challenge*
- *The Affect Sub-Challenge*

INTERSPEECH 2009 Emotion Challenge:

- *The Open Performance Sub-Challenge*
- *The Classifier Performance Sub-Challenge*

Outline

- Introduction to speech pattern classification
- Feature Extraction
  – Type of features
  – Additional processing (including time related)
  – Tools
- Machine learning
  – Speech common models
  – Tools
- Some task examples

PART I

FEATURE EXTRACTION

Features for SPC

- Desirable attributes of features for automatic methods:
  - **Informative**
    - Similar(dissimilar) sounds have similar (dissimilar) features
    - Provides discriminative information wrt the target task
      - Discard stuff irrelevant (ie. pitch in Portuguese ASR)
      - Pattern recognition techniques are rarely independent of the problem domain → proper selection of features affects performance
  - **Practical**
    - Occurs naturally and frequently in speech
    - Easy to measure
  - **Robust**
    - Not change over time
    - Not (very) affected by noise and channel

Features for SPC
Features (coarse) classification

- Region of analysis used for feature extraction:
  - Local (frame)
  - Global (functionals)
    - mean, median, std deviation, maximum, minimum, range (max–min), etc.
  - Segmental
    - phoneme, voiced/unvoiced, word, uniform segmentation, etc.

- Type of information represented/extracted:
  - Spectral features:
    - Classical speech (ASR) features, spectral measures, etc.
  - Prosodic features:
    - Pitch, Energy, timing, articulation, etc.
  - Other:
    - Time-domain, model-based, high-level

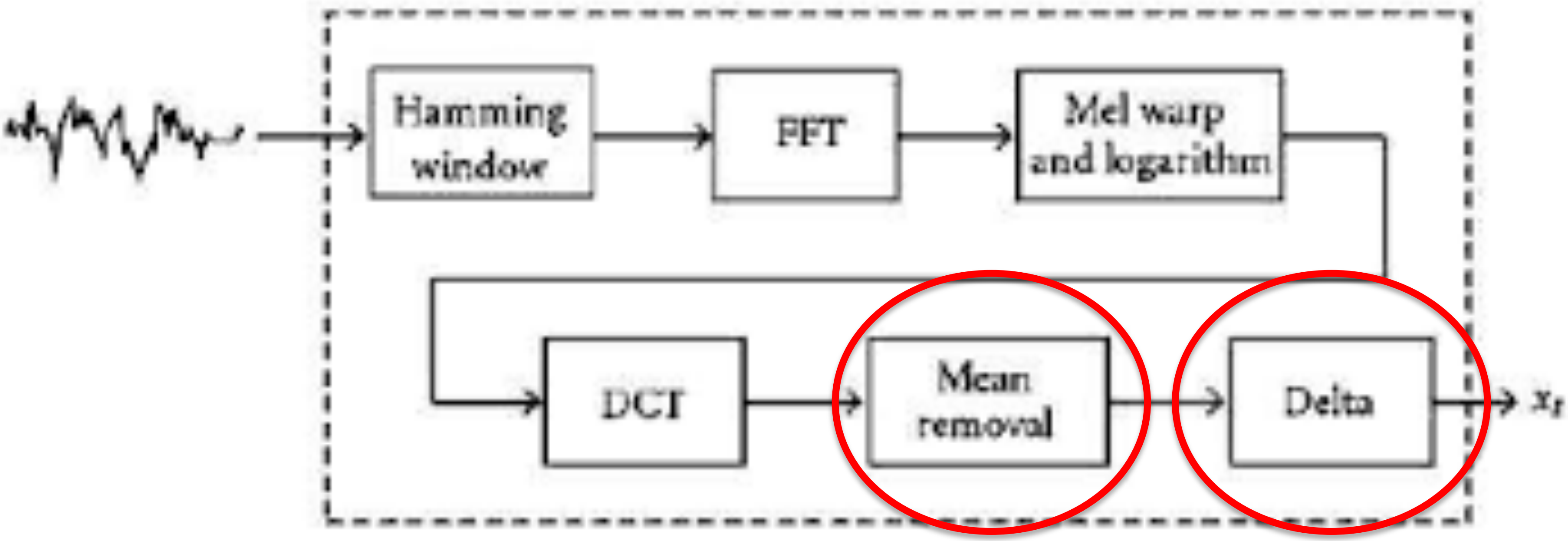
Classical (ASR) speech features

- Classical speech features are acoustic/spectral discussed in the **Speech Analysis** part of the course:
  - Extracted from overlapping windows (**frames**) of ~15-30 msec
  - (Partially) inspired in the human production and **perception** process:
    - Production (source-filter model) → LPC analysis
    - Perception → Cepstral Coefficients

- Notice that “traditional” and “classic” mostly means ASR:
  - **Why?** Because ASR has been the most relevant speech pattern classification task for decades and influences any other task
  - Some of these typical feature representation include:
    - LPC; LPCC; PLP; MFCC

Classical (ASR) speech features

Recalling MFCC



Classical ASR Features

Time information: Deltas and Double-deltas

- Dynamic characteristics of sounds often convey significant information
  - Stop closures and releases
  - Formant transitions

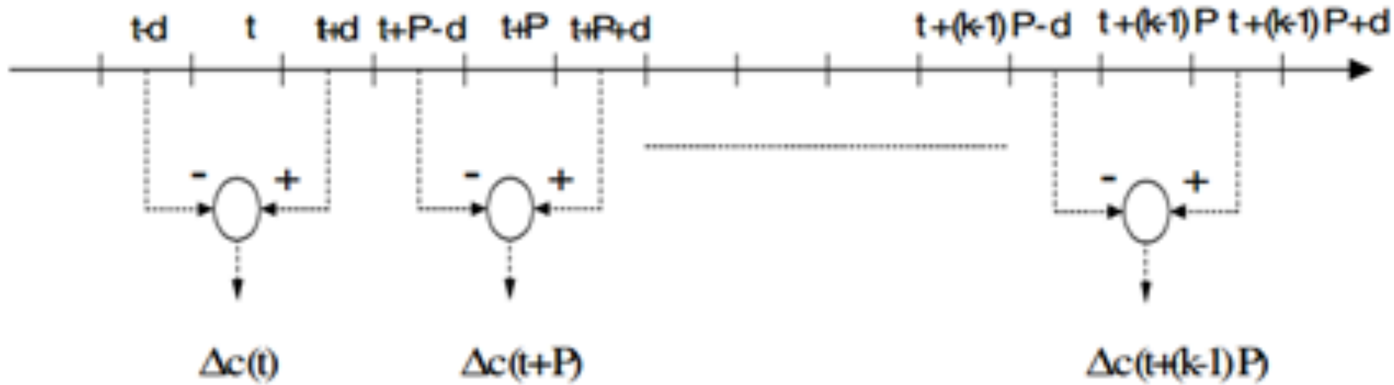
- **Bright idea** augment normal “static” feature vector with dynamic features:
  - Moreover, it can contribute to noise slow time-varying cancellation (homomorphic)

- If y(t) is the feature vector at time t, then compute

$\Delta yt = y(t+D) - y(t-D)$ and create a new feature vector

Classical (ASR) speech features
Time information: Deltas and Double-deltas

- Delta computation block diagram:



- Computation of the deltas of delta coefficients? → Double-deltas
  – So-called first and second derivatives, respectively
- Typical final feature vector:
  – Static + deltas + Double-deltas → Newdim = 3xDim
- Variants: Need more temporal context? → Shifted Delta Cepstrum

CAN BE USED WITH ANY TYPE OF LOCAL FEATURES (NOT ONLY CLASSIC ASR)

Classical (ASR) speech features

Feature Normalization: Cepstral Mean Normalization

- Homomorphic properties of cepstral features?
  - Convolutive effects in time domain are linear in the cepstral domain
- Mean removal for constant channel-effect cancelation
  - Compute mean feature vector:
    - Sliding window, complete segment?
  - Simple removal/subtraction
- Some typical variations: CM**V**N, Feature Warping

CAN BE USED WITH ANY TYPE OF LOCAL FEATURES (NOT ONLY CLASSIC ASR)

More spectral features (not typical of ASR)

- Perceptually motivated spectrum:
  - Direct use of mel-/bark- scaled filter bank
  - Perceptual features computed over mel/bark spectrum:
    - Loudness, sharpness (centroid), spread

- Measurements of the spectrum:
  - Spectral centroid (1st order moment), Spectral Spread (2nd order moment), spectral skewness (3rd order moment), spectral kurtosis (4th order moment)
  - Spectral envelope, flatness, slope, decrease, roll-off (energy below 95%), variation (flux)

- Harmonic features (voice quality, related to prosodic features):
  - Obtained from the peaks (close to multiples of F0) of the spectrum
  - Harmonic/noise ratio, harmonic deviation, noiseness, etc.

- Formants:
  - first to fourth formants, and their bandwidths, etc.

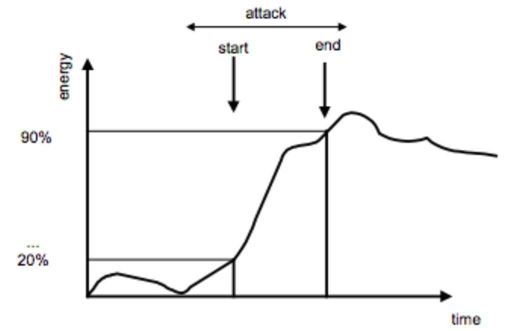
Prosodic features

- **Fundamental frequency (F0):** mean, median, standard deviation, maximum, minimum, range (max–min), jitter, pitch contour (linear regression coefficients, Legendre parameters), etc.

- **Energy:** mean, median, standard deviation, maximum, minimum, range (max–min), shimmer, energy contours (linear regression coefficients, Legendre parameters), voice level, etc.

- **Duration:** speech rate, ratio of duration of voiced and unvoiced regions, duration of the longest voiced speech.

Other features



- Time-domain features:
  - ZCR
  - Autocorrelation
  - Attack (duration, slope (increase, decrease))
  - Temporal energy centroid
- Model-based features:
  - A model is obtained from the audio segments and the parameters used as a feature vector
    - We will comment more on this next days
  - A (pre-trained) model is used to obtain features:
    - Bottle-neck features
    - Posterior-based features
- High-level features
  - Usually depend on text (or ASR or other sophisticated processes)
  - Phonetic (number of phonemes per second, phonotactic), lexical (n-gram words), discourse markers (filler pauses)

Feature Pre-Processing

- It is possible to apply general purpose speech enhancement methods as pre-processing:
  - Wiener filter,
  - Spectral Subtraction,
  - RASTA filtering (kind of auditory motivated filtering process)
  - etc.

- It is (almost) mandatory to apply Voice Activity Detection (**VAD**):

  Why?
    - Presence of silence in the training data can corrupt the model
    - Presence of silence in the test data will degrade the decision

  How?
    - energy thresholds (non-adaptive and adaptive)
    - waveform and spectrum analysis (and some heuristics)
      - pitch and harmonic detection, periodicity measures, zero-crossing rates
    - Based on statistical models

Feature Post-Processing

Feature Normalization: Cepstral Mean Normalization

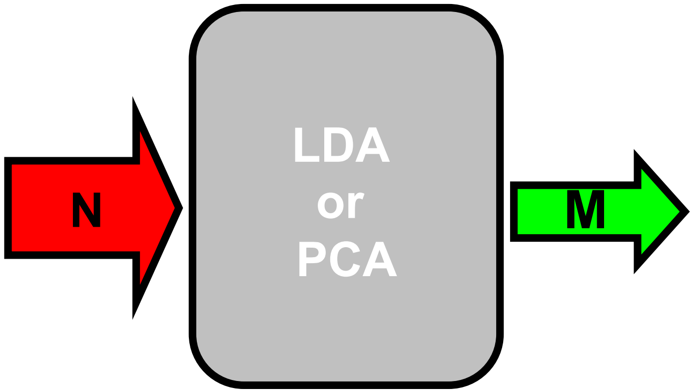
- Remember Homomorphic properties of cepstral features?
  - Convolutive effects in time domain are linear in the cepstral domain
- Mean removal for constant channel-effect cancelation
  - Compute mean feature vector:
    - Sliding window, complete segment?
  - Simple removal/subtraction
- Some typical variations: CM**V**N, Feature Warping

CAN BE USED WITH ANY TYPE OF LOCAL FEATURES (NOT ONLY CLASSIC ASR)

Feature dimensionality reduction

- Feature selection:
  - Smaller feature space (fewer dimensions):
    - Simple models
    - Less training data and faster training
  - How?
    - Knowledge-based
    - Automatic (Based on some selection criterion in a development set)

- Feature transformation:
  - Linear/non-linear transformations
  - Need to be trained
  - Most popular methods:
    - Principal Component Analysis (PCA):
      - Simple; Not need for labelled data
    - Linear (Fisher) Discriminant Analysis (LDA):
      - Theoretically better; Need for labelled data; Can be used as a classifier



Tools for Feature Extraction: HTK

HTK http://htk.eng.cam.ac.uk

- HMM toolkit primarily used for ASR
  - I has been one of the most important publicly available ASR toolkits for many years
  - Provides source code written in C (Linux/Windows)
    - It does not allow re-distribution
  - Well-documented
- Contains several tools, including **HCopy** , the tool that allows for feature extraction
  - **HCopy** permits computation of the most relevant classical ASR features and typical pre-/post- processing:
    - LPC, FBE, MFCC, PLP
    - Energy, Delta, double-delta, CMVN, VTLN
  - It can read several audio input formats

HTK

- Usage example:

```
# Obtain features
# hcopy.conf – configuration file
# lists/raws2code.scp  – 2 columns list with audio and feature file
Hcopy -T 1 –C hcopy.conf -S lists/raws2code.scp


# List contents of feature file
Hlist -o -h file000.mfc
```

Tools for Feature Extraction: openSMILE

openSMILE - Open-Source Audio Feature Extractor

SMILE - Speech & Music Interpretation by Large-space Extraction

http://audeering.com/research/opensmile/

- It is a extremely popular and versatile feature extraction tool in the area of paralinguistics:
  - Baseline in ComParE evaluations
- Open-source multi-platform (written in C++)
  - It permits stand-alone tool usage or library access
- Well-documented http://www.audeering.com/research-and-open-source/files/openSMILE-book-latest.pdf
- Popular I/O file formats are supported:
  - HTK, Comma separated value (CSV) text, WEKA, LibSVM

openSMILE main characteristics

- **General audio signal processing:**
  - Windowing Functions (Hamming, Hann, Gauss, Sine, …)
  - Fast-Fourier Transform
  - Pre-emphasis filter
  - FIR filterbanks
  - Autocorrelation
  - Cepstrum
- **Extraction of speech-related features, e.g.:**
  - Signal energy
  - Loudness
  - Mel-/Bark-/Octave-spectra
  - MFCC
  - PLP-CC
  - Pitch
  - Voice quality (Jitter, Shimmer)
  - Formants
  - LPC
  - Line Spectral Pairs (LSP)
- **Music-related features:**
  - Pitch classes (semitone spectrum)
  - CHROMA and CENS features
  - Weighted differential

openSMILE main characteristics

- **Moving average smoothing of feature contours**
- **Moving average mean subtraction and variance normalisation (e.g. for on-line cepstral mean subtraction)**
- **On-line histogram equalisation**
- **Delta Regression coefficients of arbitrary order**
- **Statistical functionals (feature summaries), e.g.:**
  - Means, Extremes
  - Moments
  - Segments
  - Samples
  - Peaks
  - Linear and quadratic regression
  - Percentiles
  - Durations
  - Onsets
  - DCT coefficients
  - Zero-crossings
  - Modulation-spectrum (new)

Over 6000 features!!!

openSMILE usage

SMILExtract -C $config_file_name -I $input_file_name -O $output_file_name

where:

+ $config_file_name: there are several configuration files in the package with a selection of features

/openSMILE-2.2rc1/config/gemaps/eGeMAPSv01a.conf
/openSMILE-2.2rc1/config/gemaps/GeMAPSv01a.conf

+ $output_file_name (single archive .arff for all input audio files)

[it is possible to add "Class" label information to incorporate to the feature set and use directly with modeling toolkits (WEKA)]

GeMAPS – openSMILE sub-set

GeMAPS - Geneva Minimalistic Acoustic Parameter Set for Voice Research and Affective Computing:

- Kind of standard acoustic parameter recommendation, agreed upon by many leading scientists, including psychologists, linguists, voice researchers, and engineers,
- Extracted with openSMILE
- Basic set contains 62 parameters
  - Extended set includes 26 additional features (62+26=88)
- Let’s have a look...

Other publicly available toolboxes

- PRAAT
  - http://www.fon.hum.uva.nl/praat/
  - Phonetics & linguistic oriented
- MIR toolbox
  https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox
  - MATLAB code
  - Music oriented, but it also contains speech features
- YAAFE – Yet another audio feature extraction
  http://yaafe.sourceforge.net/
  - Python and MATLAB bindings
  - Collection of audio features

PART II
PATTERN CLASSIFICATION FOR SPEECH

Introduction to ML

- Assume we have a training set D={(x(i),y(i))} drawn from the distribution p(x,y), x€X y€Y

- The goal of learning is to find a decision function f: X → Y that correctly predicts the output of future input from the same distribution:

$$f(x) = \mathit{argmax}_y \, d_y(x)$$

- Two fundamental elements in ML methods:
  - Type of "discriminant function" (the model)
  - Type of "loss function" (the training objective)

Classification (coarse) of ML methods

- Nature of the model and loss function:
  - Generative learning (descriptive)
    - Models the probability distribution of data $p(x|y)$, ex: GMM
    - Loss function: Joint likelihood distribution → Maximum Likelihood estimation (MLE) training criteria

    Note: Bayes' rule makes them useful for classification $p(y|x) = p(x|y)p(y)$
  - Discriminative learning
    - Discriminative models maps directly x to y, ex: MLPs, SVM, CRFs
    - Discriminative loss function, ex. MCE, MPE, MMI

    Note: Discriminative learning criteria can be used with Generative models

- How training data is used:
  - Supervised – all training samples are labeled
  - Semi-supervised – both labeled and unlabeled
  - Unsupervised – all training samples are unlabeled

Statistical models is speech pattern classification problems

- The most common model in speech pattern recognition problems is the Gaussian Mixture Model (GMM):
  - A GMM is a particular case of Hidden Markov models (HMM) → HMMs also model time

- Many other models have been also used in different speech classification tasks:
  - K-NN – K nearest neighbor
  - MLP – Multi-layer perceptron
  - SVM – Support Vector Machines
  - DNN – Deep neural networks
  - etc.

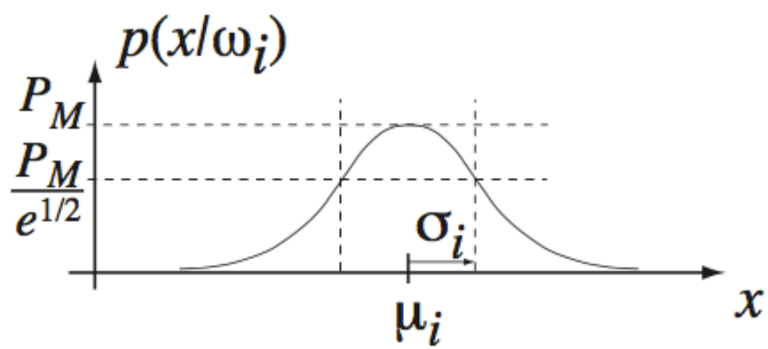
Gaussian mixture models (GMM)

Gaussian models

- Easiest way to model distributions is via parametric model
  - assume known form, estimate a few parameters
- Gaussian model is simple and useful. In 1D

$$p(x \mid \theta_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu_i}{\sigma_i}\right)^2\right]$$

- Parameters mean μ_i and variance σ_i $\rightarrow$ fit



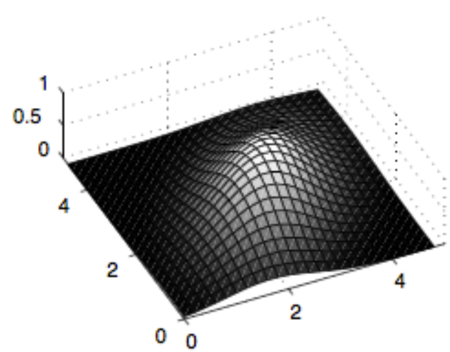
Slide after [1]

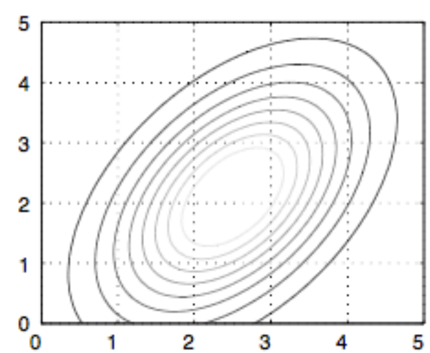
Gaussian mixture models (GMM)

Gaussians in d dimensions

$$p(\mathbf{x} \mid \theta_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mu_i)^T\Sigma_i^{-1}(\mathbf{x}-\mu_i)\right]$$

Described by a d-dimensional mean μ_i
and a $d \times d$ covariance matrix Σ_i



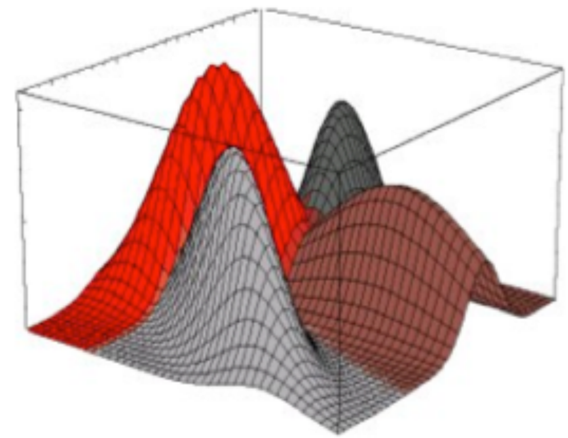


Slide after [1]

Gaussian mixture models (GMM)

Gaussian mixture models

- Single Gaussians cannot model
  - distributions with multiple modes
  - distributions with nonlinear correlations
- What about a weighted sum?



$$p(x) \approx \sum_k c_k p(x \mid \theta_k)$$

  - where $\{c_k\}$ is a set of weights and $\{p(x \mid \theta_k)\}$ is a set of Gaussian components
  - can fit anything given enough components
- Interpretation: each observation is generated by one of the Gaussians, chosen with probability $c_k = p(\theta_k)$

Slide after [1]

Gaussian mixture models (GMM)

In order to use GMMs we need:

1. A method to estimate GMM parameters
   - We use the **Expectation-maximization** (EM) algorithm:
     - General procedure for estimating model parameters
       - Similar for instance to **k-means** used in VQ
     - Iteratively updated model parameters leads to MLE:
       - Can lead to local optimum – depend on initialization
2. Compute the (log-)**likelihood** of a sequence of features given a GMM

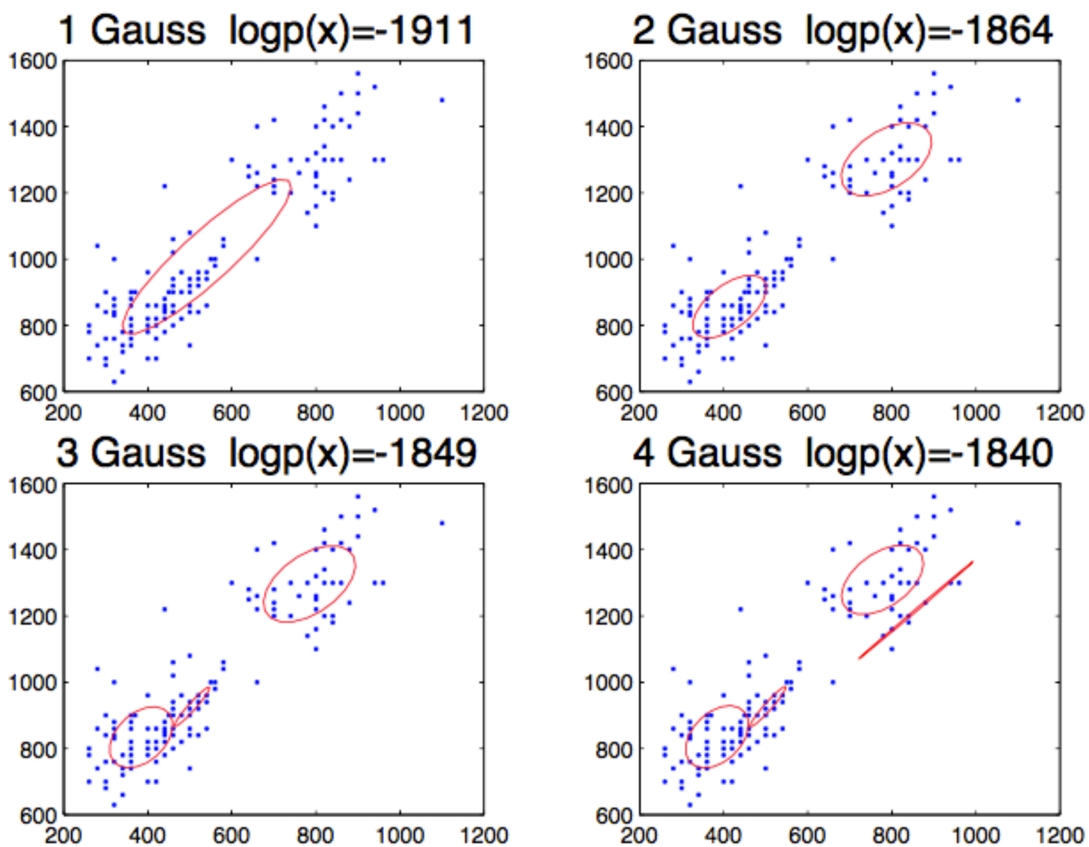
$$\log p(\vec{x}_1,\ldots,\vec{x}_N \mid \lambda) = \sum_{n=1}^{N} \log p(\vec{x}_n \mid \lambda)$$

$$= \sum_{n=1}^{N} \log\left(\sum_{i=1}^{\mathrm{M}} p_i b_i(\vec{x}_n) \right)$$

Gaussian mixture models (GMM)

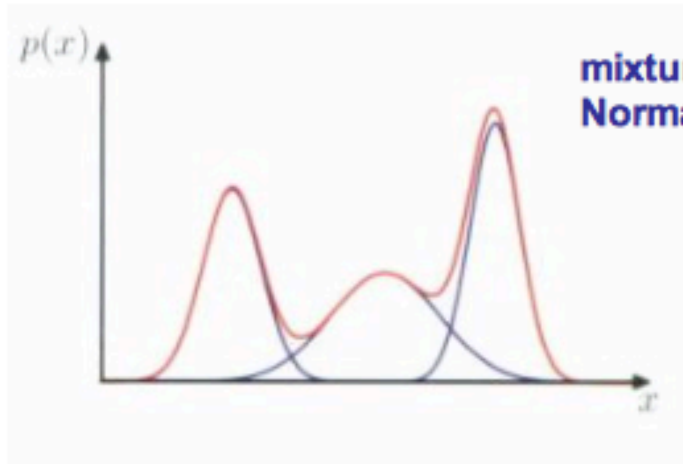
GMM examples

Vowel data fit with different mixture counts



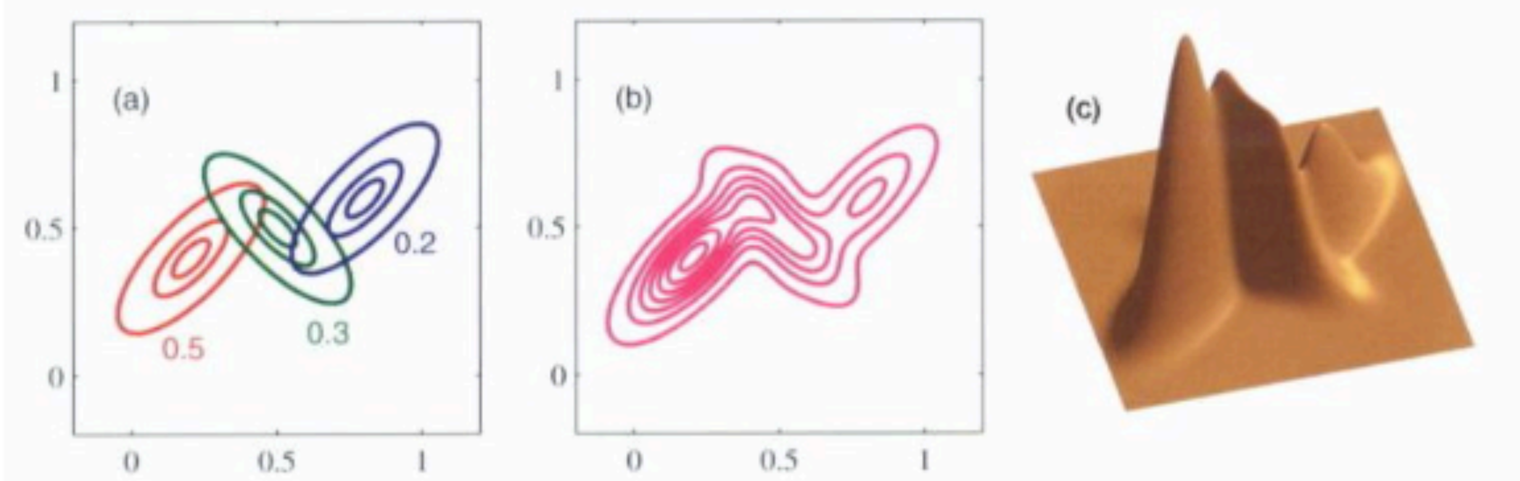
Slide after [1]

Gaussian mixture models (GMM)



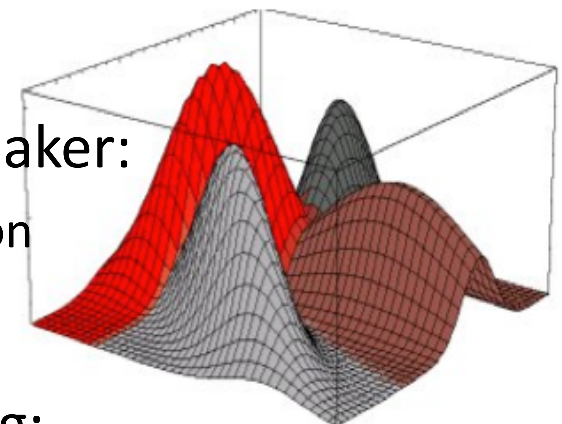
mixture of 3 one-dimensional Normal distributions

mixture of 3 two-dimensional Gaussians



Gaussian mixture models (GMM)
GMM-ML & Speaker Recognition

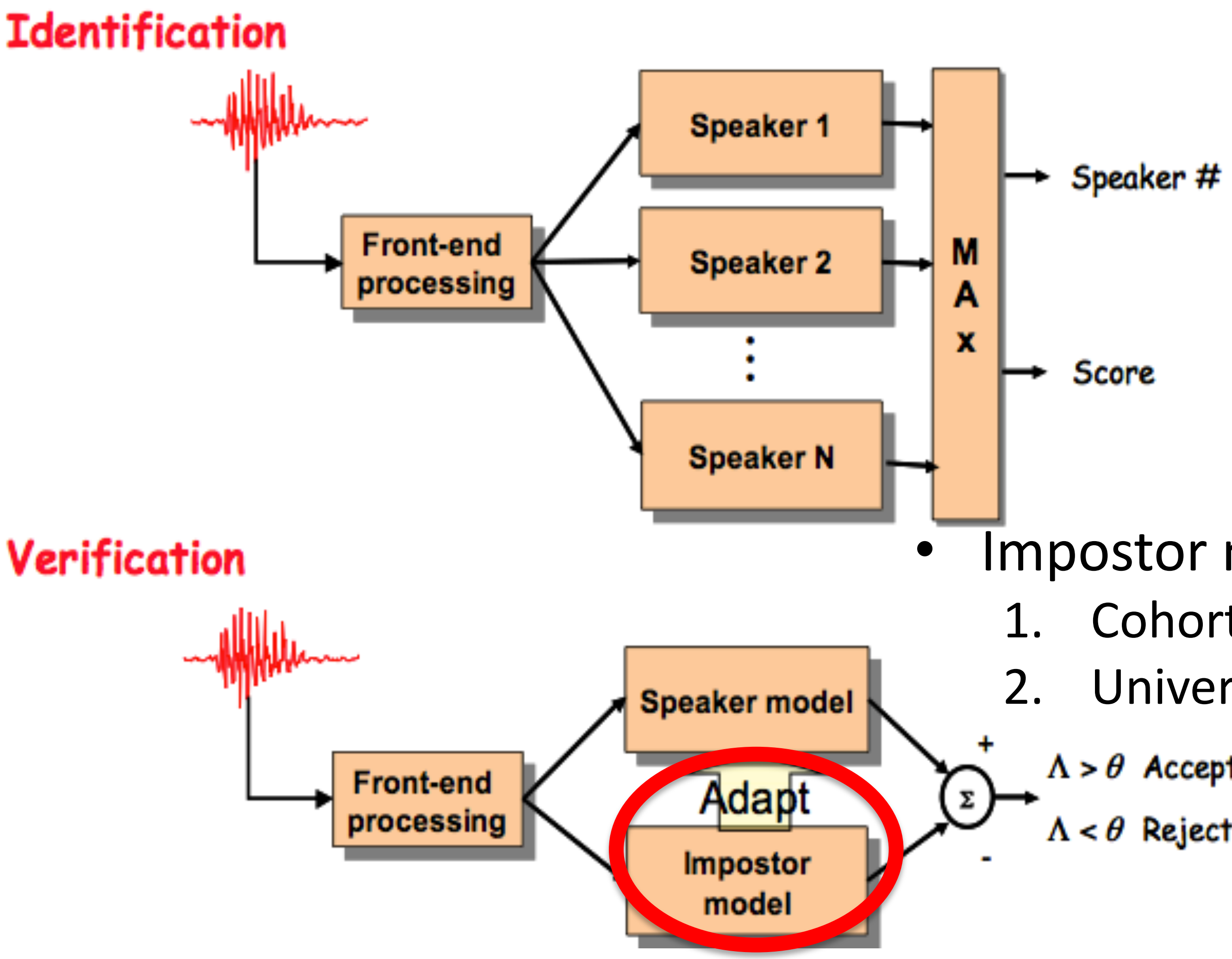
- Conventional **GMM-ML** approach:
  - In **train** phase:
    - Train a GMM model per target speaker:
      - Apply EM algorithm for ML estimation



  - In **test** phase:
    - Compute log-likelihoods for scoring:
      - Speaker ID → MAX(LL)
      - Speaker Verification → log-likelihood compared to a threshold or impostor model

Gaussian mixture models (GMM)

GMM-ML & Speaker Recognition



- Impostor model approaches:
  1. Cohort of impostors
  2. Universal model

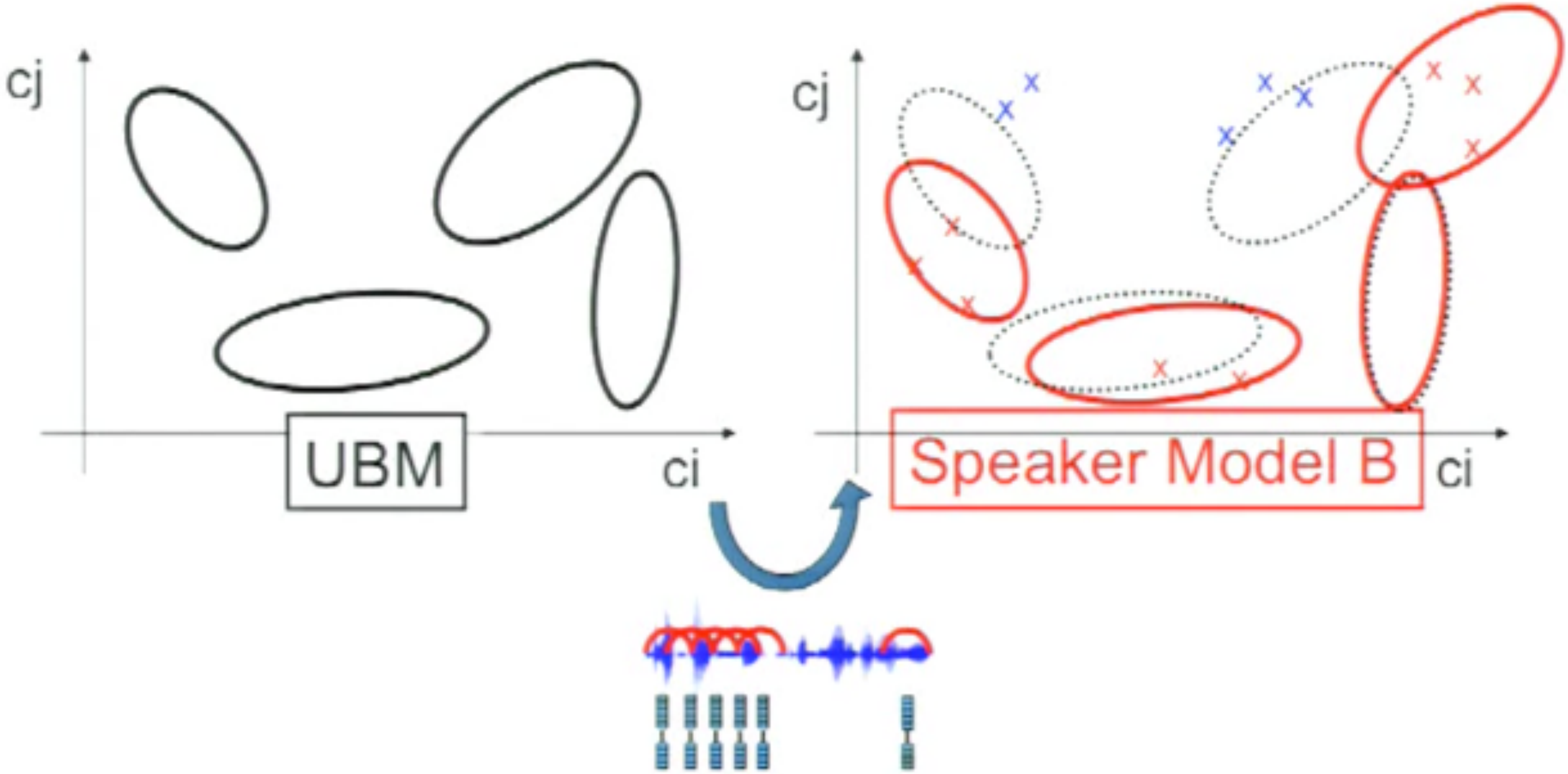
Slide after [2]

Gaussian mixture models (GMM)
GMM-UBM & Speaker Recognition

- **GMM-UBM** approach:
  - In **train** phase:
    - Estimate the parameters of an UBM (Universal Background Model) with data from different speakers, channels, noise conditions, etc…
    - Adapt the UBM to each one of the target speakers:
      - Use MAP adaptation (usually only-means)
      - MAP "updates" the parameters of the prior model with new "information" obtained from the adaptation data (instead of computing from-the-scratch new model parameters)
  - In **test** phase is like in previous GMM-ML approach.
  - **Advantages**
    - Needs less data,
    - permits updating only seen events,
    - keeps correspondence between means, allows fast scoring (top-M)

Gaussian mixture models (GMM)

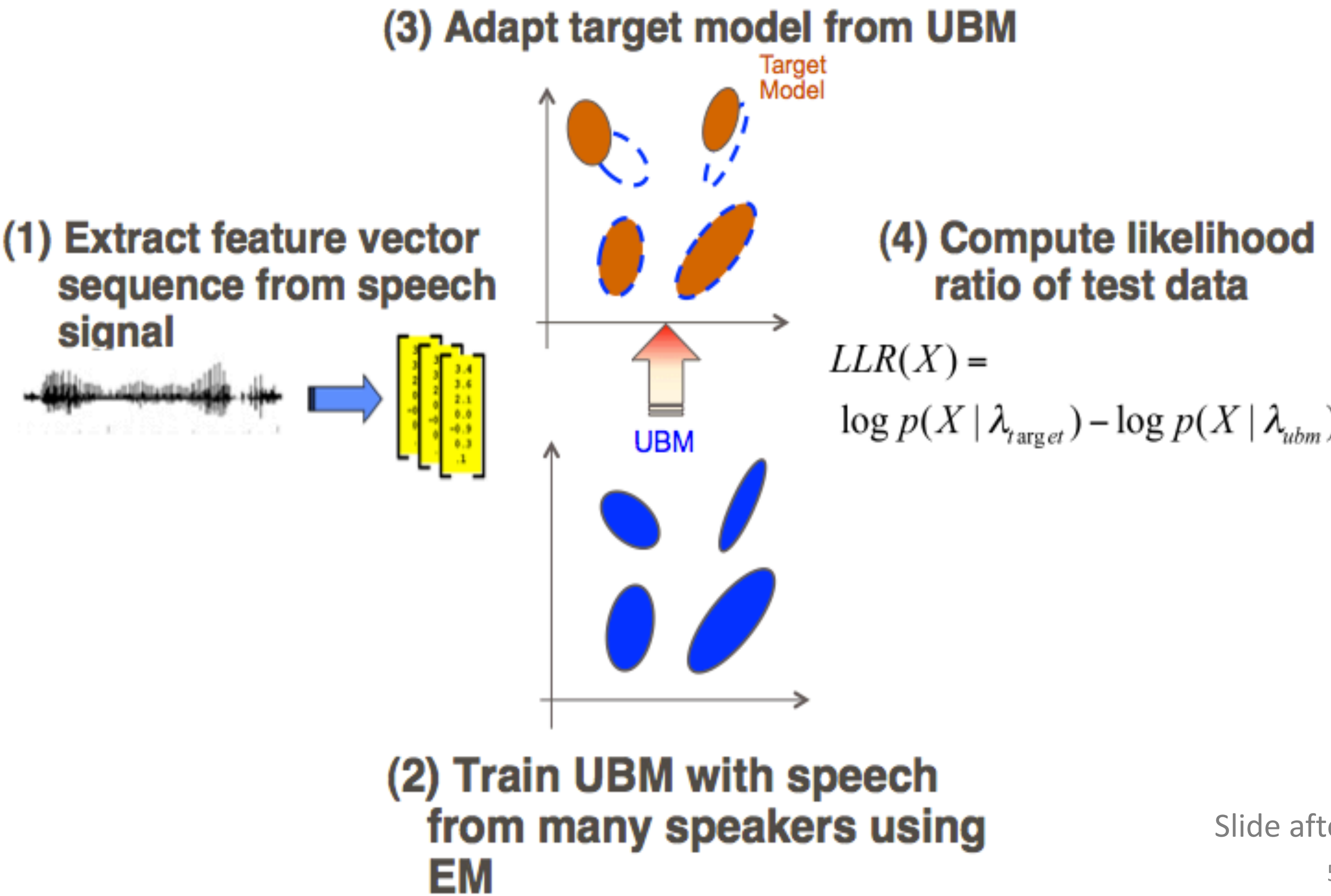
GMM-UBM & Speaker Recognition



Slide after [3]

Gaussian mixture models (GMM)

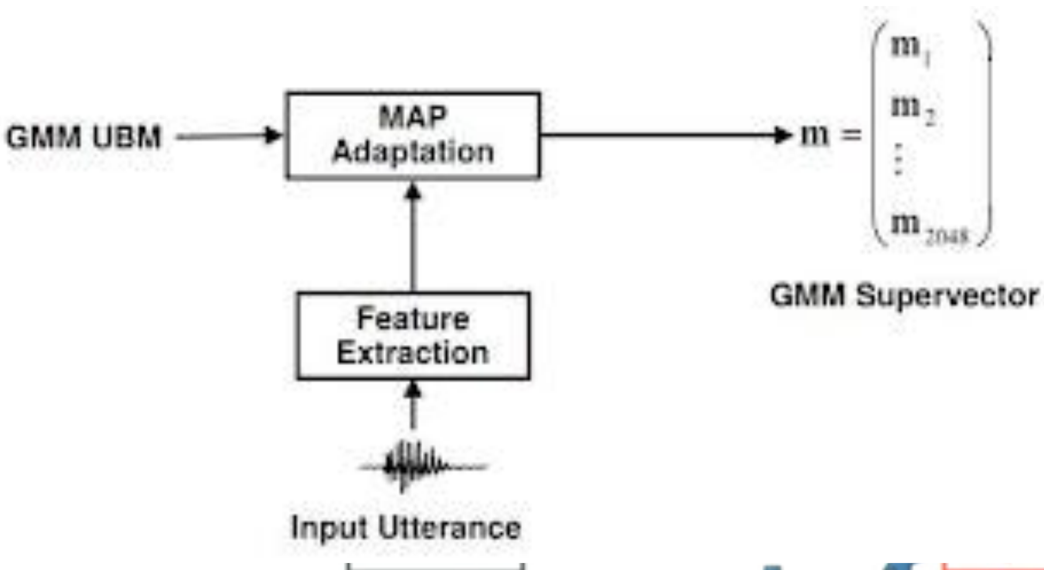
GMM-UBM & Speaker Recognition



Slide after [2]

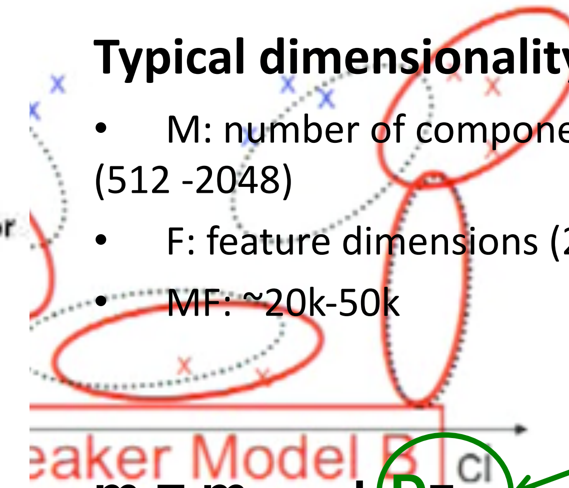
Gaussian mixture models (GMM)

GMM-UBM: The supervector concept



Typical dimensionality:

- M: number of components (512 -2048)
- F: feature dimensions (20-60)
- MF: ~20k-50k



MAP

$$\mathbf{m} = \mathbf{m}_{\mathbf{UBM}} + \mathbf{D}\mathbf{z}_{\mathbf{sh}}$$

D = Full rank diagonal matrix (relevance MAP)

$\mathbf{z}_{\mathbf{sh}}$ = Full rank vector

- The supervector concept and its derivations has had a **huge impact** in in the last decade:

1. As a kind of feature extraction for discriminative machine learning methods → REMEMBER features based on models!?
2. As a tool for Factor Analysis derivation

Gaussian mixture models (GMM)

Factor Analysis approaches: The i-vector

Factor Analysis (FA) is a statistical method for investigating if a number of variables are linearly related to a small number of unobservable factors.

GMM-UBM (MAP) → $\mathbf{m} = \mathbf{m}_{UBM} + \mathbf{D}\mathbf{z}_{sh}$

- **D** diagonal full-rank
- $\mathbf{z}_{sh}$: speaker (and more) component

i-vectors → $\mathbf{m} = \mathbf{m}_{UBM} + \mathbf{T}\mathbf{w}$

- **T** total variability subspace (low-rank)
- **w** variability (loading) factors, a.k.a i-vectors
  - ~400-600 dimensions
  - They contain all speaker and channel variability
  - It is used as a low-dimensional representation (on top of them other models can be trained)

Example of discriminative model

Support Vector Machines (SVMs)

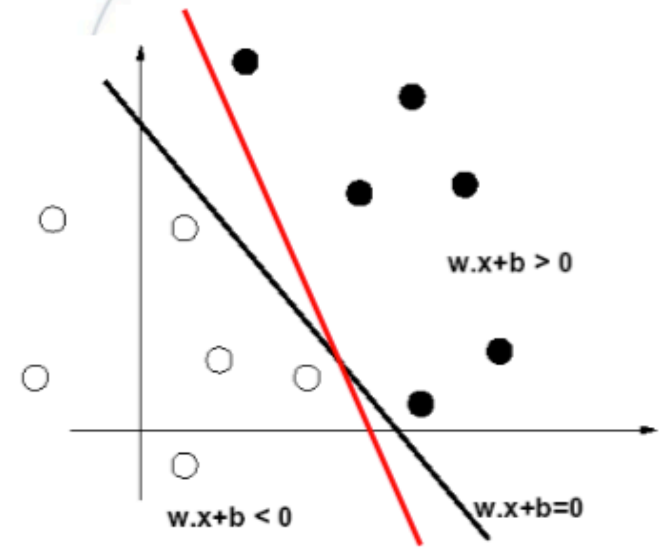
SVM – Basic formulation

- Linear classifier (linear combination of features)
- Hyperplane equation

$$\vec{w}.\vec{x}+b=0$$

- Class is determined by the sign of

$$\vec{w}.\vec{x}+b$$



- Non-probabilistic binary linear classifier

Slide after [4]

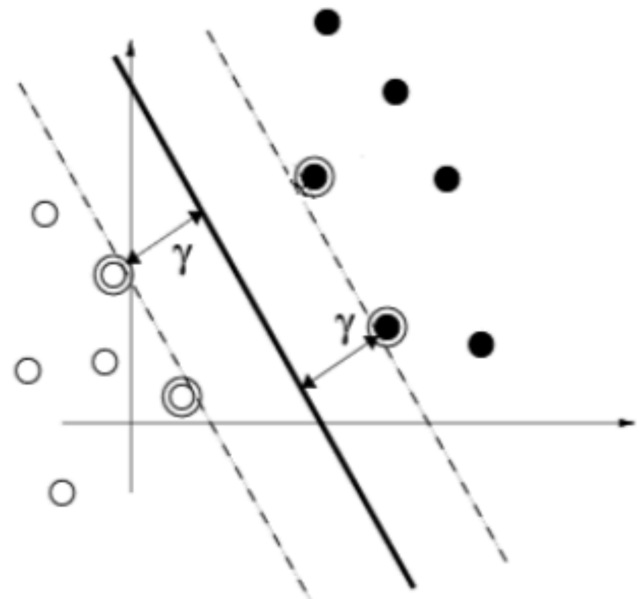
Example of discriminative model
Support Vector Machines (SVMs)

SVM - maximum-margin hyperplane

- Margin between both hyperplanes

$$\left.\begin{array}{l}\vec{w}.\vec{x}_i+b=1\\[1em]\vec{w}.\vec{x}_i+b=-1\end{array}\right\}y_i(\vec{w}.\vec{x}_i+b)\geq 1$$

- Support Vectors: Elements that are at contained by the hyperplanes



Slide after [4]

Example of discriminative model

Support Vector Machines (SVMs)

SVM – Support Vectors

- The hyperplane can be calculated using only a linear combination of the support vectors

$$\vec{w}^* = \sum_{x_i \in VS} \lambda_i^* y_i \vec{x}_i$$

- The parameter λ_i^* has to be estimated by the minimization procedure
- The parameter b also needs to be estimated

Slide after [4]

Example of discriminative model
Support Vector Machines (SVMs)

SVM - Classifying

- A new observation can be classified using the dot product of the support vectors and the new example:

$$\vec{w}^{*}.\vec{x} + b = \sum_{x_i \in VS} \lambda_i^{*} y_i \vec{x_i}.\vec{x} + b^{*}$$

- The dot product can be replaced by kernels
- Kernels allow to transform the initial space to a new space where the examples are linearly separable

Slide after [4]

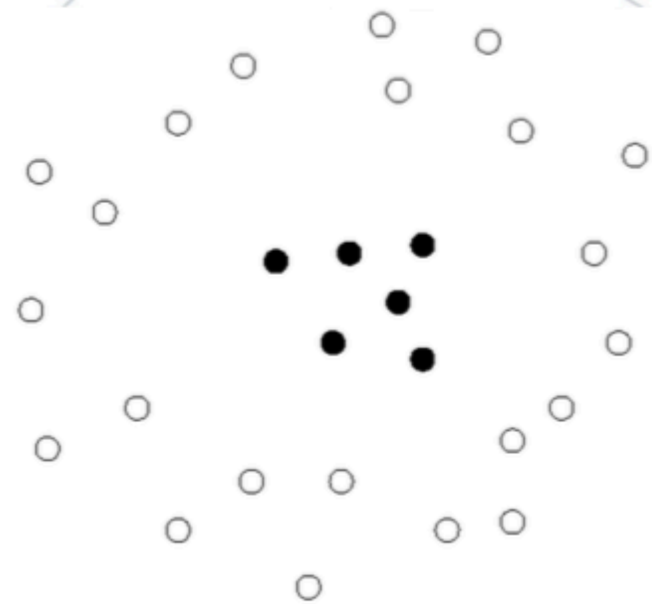
Example of discriminative model

Support Vector Machines (SVMs)

SVM – Non Linear Space

- When the examples are not linearly separable, a kernel may be used transform the initial space

$$K(\vec{x}, \vec{x}') = \phi(\vec{x}).\phi(\vec{x}')$$



Slide after [4]

Example of discriminative model

Support Vector Machines (SVMs)

SVM – Basic Kernels

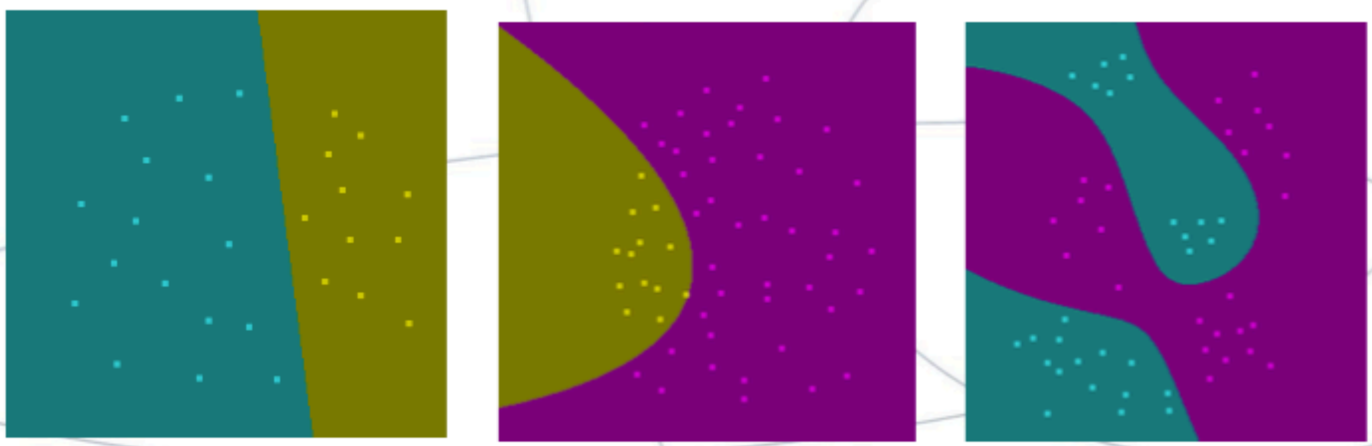
- Linear Kernel – Corresponds to the dot product in the previously presented expression
- Polynomial Kernel $$K_{Poli3}(\vec{x}, \vec{x}') = (\gamma \vec{x}.\vec{x}' + c)^d$$
  - Where d is the degree of the polynomial. c and γ are constants
- Radial Basis Kernel $$K(\vec{x}, \vec{x}') = \exp(-\gamma \left\| \vec{x} - \vec{x}' \right\|^2)$$
  - Where γ defines the "size" of the radial basis function

Slide after [4]

Example of discriminative model
Support Vector Machines (SVMs)

SVM – Kernel Examples

- http://www.csie.ntu.edu.tw/~cjlin/libsvm/



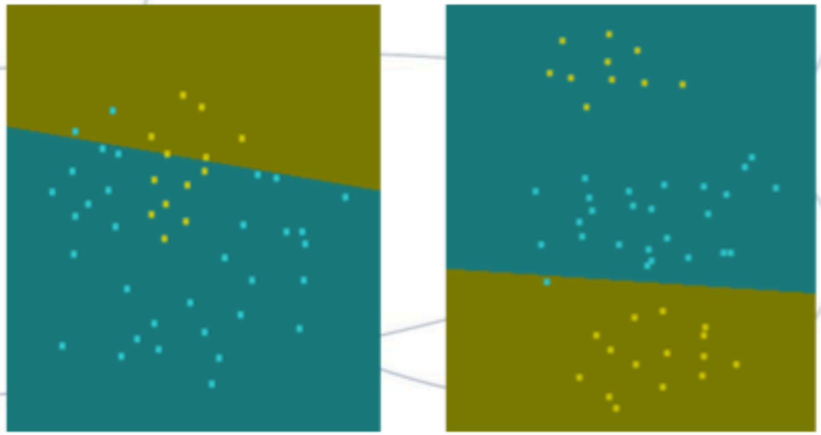
Linear | Polinomial Degree=2 | Radial

Example of discriminative model

Support Vector Machines (SVMs)

SVM – Kernel Advantages / Disadvantages (1/3)

- Linear Kernel
- Advantage
  - is faster to calculate and less prune to overfitting
- Disadvantage
  - If the data is not linearly separable (can't learn)
  - High dimension data is easier to separate
  - Complex data is harder

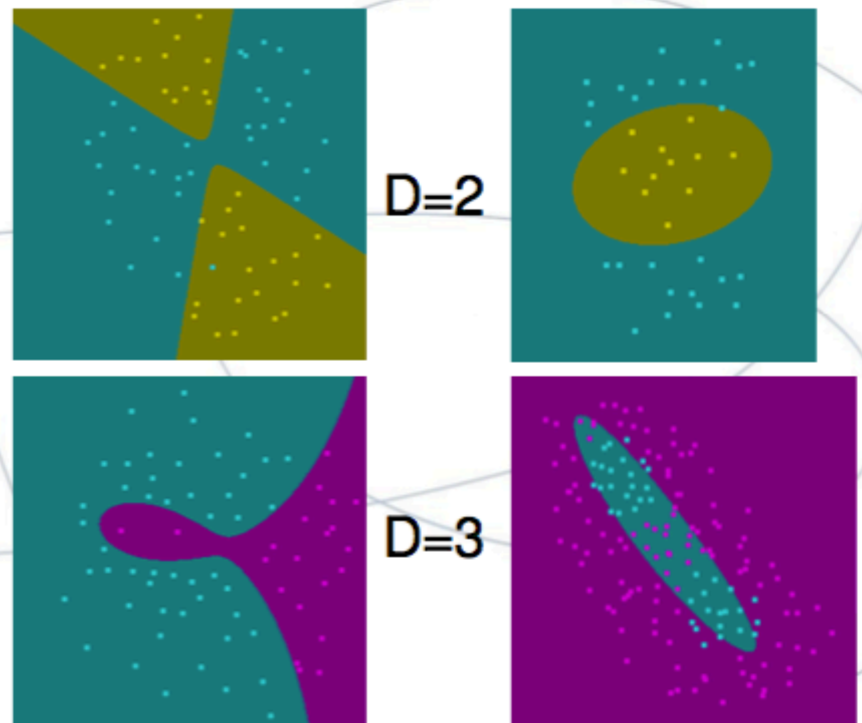


Slide after [4]

Example of discriminative model
Support Vector Machines (SVMs)

SVM – Kernel Advantages / Disadvantages (2/3)

- Polinomial Kernel
- Advantage
  – Higher power to separate data
- Disadvantage
  – Can have overfitting problems, specially with high degree polynomials
  – Still some data that can't be separated



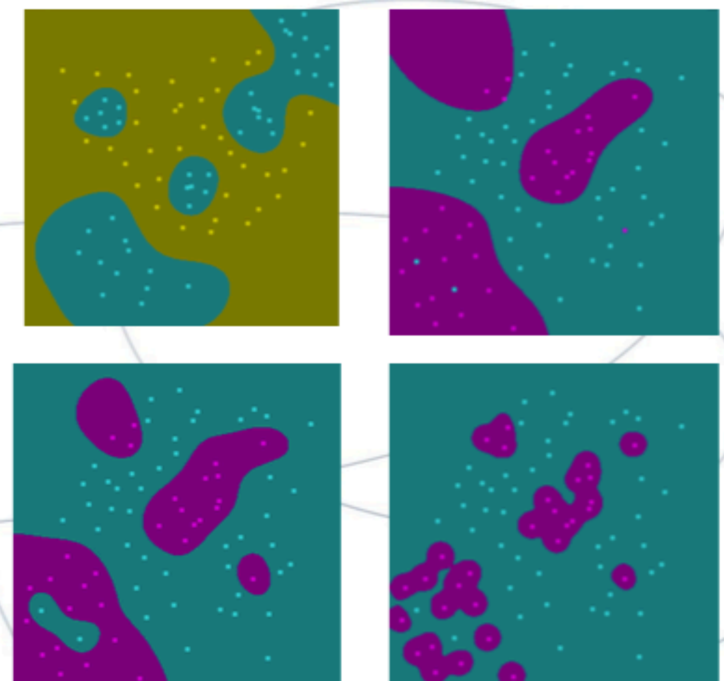
Slide after [4]

Example of discriminative model

Support Vector Machines (SVMs)

SVM – Kernel Advantages / Disadvantages (3/3)

- Radial Kernel
- Advantage
  - In the limit it can separate any data
- Disadvantage
  - Used without caution causes many overfitting problems



Slide after [4]

Example of discriminative model

Support Vector Machines (SVMs)

SVM - Advantages

- Easy to use
  - Few parameters to test.
    - The default parameters work for most problems, though testing some parameters with a simple cross validation can give extra precision
- Works with limited data
  - SVMs are used in applications with few data (ex: medical data)
    - Calculating the maximum margin is usually a good extrapolation
- It can separate any type of data
  - In the limit radial kernels separate any data (watch for overfitting)
- Is robust to overfitting if some precautions are taken
  - Optimize the parameters with a different data set or cross validation

Slide after [4]

Brief HOW-TO: Building a classifier

- Define task and classes
  - Need a labeled training data set
- Define feature space
  - Use meaningful features, disregard useless info
  - Prepare data (some ML methods are very sensible to scale, range, etc.)
- Define decision algorithm
  - Choose the right tool for the right job
    - The literature is full of examples
  - Avoid over-fitting (too complex model for few data):
    - Need a development data set
    - If no possible, cross-validation
- Measure performance
  - Use a separate evaluation data set

Tools for speech modeling

GMM

- SPEAR: A Speaker Recognition Toolkit based on Bob (Python)

https://pythonhosted.org/bob.bio.spear/

- MATLAB - Statistics and Machine Learning Toolbox

http://www.mathworks.com/help/stats/fitgmdist.html

SVM

- LIBSVM -- A Library for Support Vector Machines

https://www.csie.ntu.edu.tw/~cjlin/libsvm/

- Weka 3: Data Mining Software in Java (Collection of ML tools)

http://www.cs.waikato.ac.nz/ml/weka/

NEURAL NETWORKS

- Neural Network Toolbox

http://www.mathworks.com/help/nnet/index.html

- QuickNet

http://www1.icsi.berkeley.edu/Speech/qn.html

References

- These are some presentations that were used for this course:

[1] Michael Mandel, “Lecture 3: Machine learning, classification, and generative models”

http://www.ee.columbia.edu/~dpwe/e6820/lectures/L03-ml.pdf

[2] Douglas A. Reynolds, “Overview of Automatic Speaker Recognition” http://www.fit.vutbr.cz/study/courses/SRE/public/prednasky/2009-10/07_spkid_doug/sid_tutorial.pdf

[3] Javier González-Domínguez, “Session Variability Compensation in Speaker Recognition” http://tv.uvigo.es/matterhorn/20022

[4] Miguel Bugalho, “Support Vector Machines (SVMs) Classifiers: Introduction and Application. Case Study: VidiVideo Audio Event Detection”