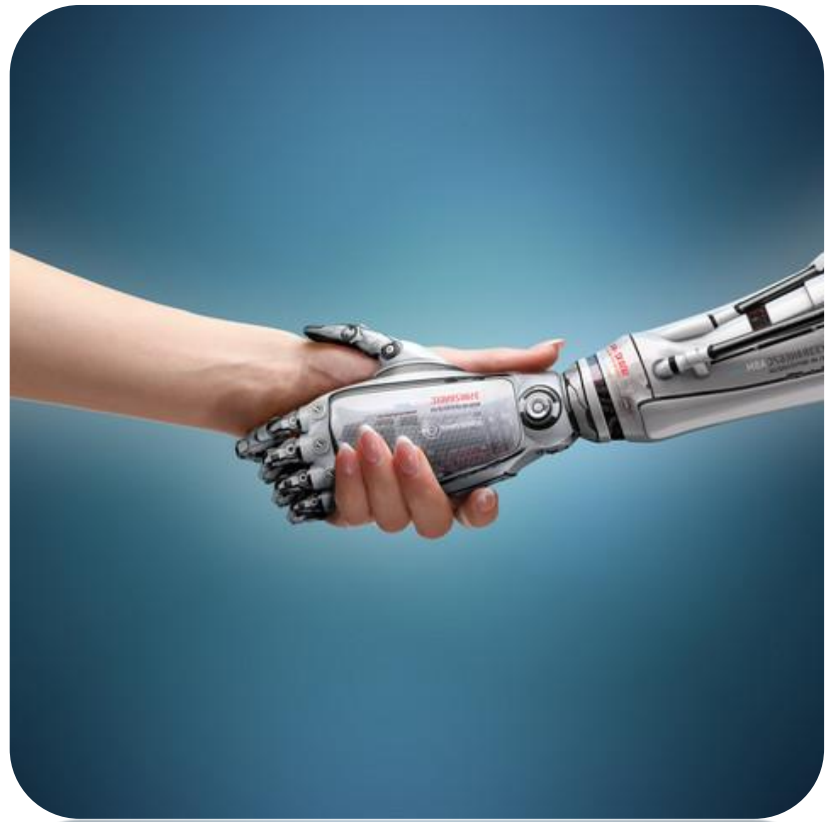
Zurich University
of Applied Sciences

Artificial Intelligence
V10: Probabilistic Learning

Probabilistic modeling
Example domain: speech processing
Gaussian Mixture Models



Based on material by
- Stuart Russell, UC Berkeley
- T. Stadelmann, R. Ewerth & B. Freisleben, U Marburg

Zurich University of Applied Sciences and Arts
InIT Institute of Applied Information Technology (stdm)

Educational objectives

- **Remember Bayesian learning**, especially Bayes' theorem and the Bayes classifier
- **Grasp** how **the concept of probability** is extremely useful **in AI**, especially for learning

- **Explain** how a **Gaussian Mixture Model** (GMM) is **trained** and **evaluated**, given the respective **equations** and the **EM** algorithm
- **Apply GMMs** for pattern recognition tasks **on audio** data

"In which we view learning as a form of uncertain reasoning from observations."

➔ Reading: AIMA, ch. 20 (optional: 13-14)

1. PROBABILISTIC MODELING

Probability distributions and density functions

Terminology: its probability density function (pdf) is one way to describe a distribution.

What does a pdf tell about a set of data?

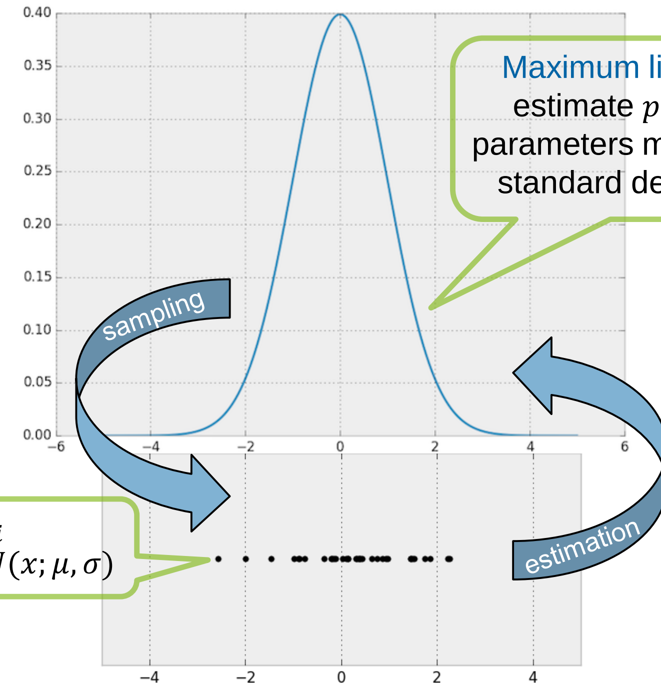
- Where to expect samples
  …with which probability
- Correlation/covariance of dimensions

➔ For data coming from some stochastic processes, the pdf tells **everything there is to know** about the data

➔ **Allows for sampling** data from the underlying distribution (generative modeling)

An example generative model

- The univariate Gaussian
  A parametric pdf, recoverable from data (Gaussianity given)



Maximum likelihood estimate $p(x)$ with parameters mean μ and standard deviation σ

Given data points x_i
Assumption: $x_i \sim p(x;\theta) = N(x;\mu,\sigma)$

Source: Brandon Amos, «*Image Completion with Deep Learning in TensorFlow*», 2016, https://bamos.github.io/2016/08/09/deep-completion/

Bayes' theorem
One of the cornerstones of modern data analysis

$$p(h|X) = \frac{p(X|h) \cdot p(h)}{p(X)}$$

with (in a machine learning context with training data X and model h)

- $p(X|h)$ the **likelihood** of the data, given the model → called the **evidence** for h
- $p(X)$ the a priori probability of the training data X → this normalization factor is **rarely needed**/used
- $p(h)$ is the **a priori** probability of hypothesis h → **often neglected** in practice due to dominance of evidence

Rev. Thomas Bayes,
1701-1761

Use cases

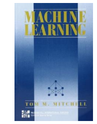
- Generally: **Convert** between prior and posterior probabilities
- Specific example: **Model selection**

  ➔ Given competing $h_i \in \mathcal{H}$, one can calculate the likelihood $p(X|h_i)$,
  then select best $\hat{h} = \max_{h_i} p(h_i|X) \approx \max_{h_i} p(X|h_i)$

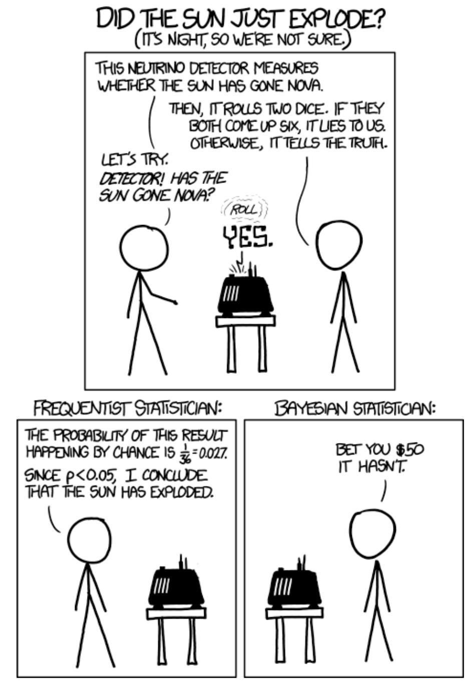
There’s a long-standing controversy pro/con Bayesianism in statistics, see e.g. http://lesswrong.com/lw/1to/what_is_bayesianism/; for the meaning of Bayesianism in machine learning, see e.g. https://www.reddit.com/r/MachineLearning/comments/6dbwnf/d_what_is_exactly_a_bayesian_guy_in_machine/

Zurich University of Applied Sciences

Bayesian reasoning & learning
Based on [Mitchell, 1997], ch. 6



Bayesian reasoning
- Built upon Bayes’ theorem to convert **prior** probabilities into **posteriors**
- Quantities of interest are governed by probability distributions
- **Optimal decisions** are made by taking them plus observed data into account



Pro
- Provides explicit probabilities for hypotheses
- **Helps to** understand/**analyze algorithms** that don’t emit probabilities
  (e.g., why to minimize sum of squares; what the inductive bias of decision trees is)
- **Everything done probabilistically**
  (e.g., every training instance contributes to the final hypothesis according to its prior probability; **prior knowledge** can be incorporated as prior probabilities for candidate hypotheses or distributions over training data; predictions can be easily combined)

Con
- Many needed **probabilities** are **unknown** in practice (approximations like sampling needed)
- Direct application of Bayes theorem often **computationally intractable**

Zurich University
of Applied Sciences

The Bayes optimal classifier
Classification's *«gold standard»*

Theoretically **optimal** (**=most probable**) classification

- **Combine predictions** of all hypotheses, **weighted by** their **posterior** probabilities:

$$\underset{y_j \in Y}{\operatorname{argmax}} \sum_{h_i \in \mathcal{H}} p\left(y_j \middle| h_i, X\right) p(h_i|X)$$

  (where y_j is a label from the set Y of classes, h_i is a specific hypothesis out of the hypothesis space $\mathcal{H}$, and $p(h_i|X)$ is the posterior of h_i given the data X)
- No other method using the same $\mathcal{H}$ and X can do better *on average*

The maximum a posteriori (MAP) hypothesis is the one with the largest $p(h|X)$

Pro

- In particular **outperforms** simply taking the classification of the **MAP hypothesis**
  Example: Let 3 classifiers predict tomorrows weather as $h_1(x) = sunny$, $h_2(rainy)$, $h_3(rainy)$ with posterior probabilities of $.5$, $.4$ and $.1$, respectively; let the true weather tomorrow be $rainy$. The MAP hypothesis h_1 wrongly predicts $sunny$ weather; the Bayes classifier truly predicts $rainy$.
- Enforces the idea of **ensemble learning**

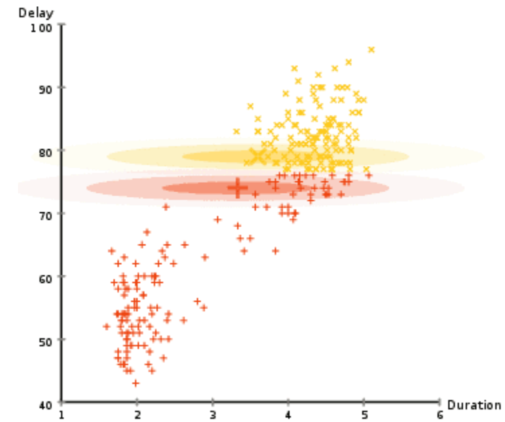
Con

- Computationally **intractable** (linear in | $\mathcal{H}$ | → see http://www.cs.cmu.edu/~tom/mlbook/NBayesLogReg.pdf)

The EM algorithm
A general-purpose, *unsupervised* learning algorithm

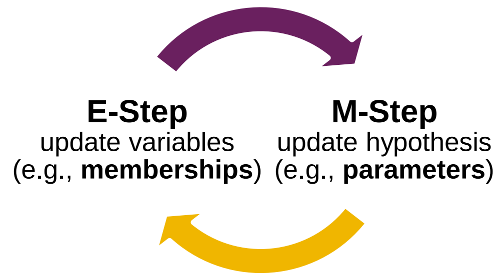
EM (expectation maximization)
- Iterative method to **learn in the presence of unobserved variables**
  ➔ A **typical hidden variable** is some sort of **group**/cluster **membership**
- Good convergence guarantees (finds local maximum)

Example
- A given dataset is known to be generated by either of 2 Gaussians (with equal probability)
- Only the data is observed
  ➔ **Which** Gaussian **generated** a certain point **is unobserved**
  ➔ The Gaussians' **parameters** are unknown
- The means & variances of these Gaussians shall be learned
  ➔ Needs an estimation of the membership probability of each point to either Gaussian



EM algorithm used to iteratively optimize the parameters of 2 Gaussians (animated)
Source: https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm)

The EM algorithm (contd.)



Algorithm

1. Start with a random initial hypothesis
   Example: **Pretend to know the parameters** μ, σ^2 of the 2 Gaussians
   (e.g., pick random values)

2. **E-Step**: Estimate expected values of unobserved variables,
   assuming the current hypothesis holds
   Example: **Compute probabilities** $\boldsymbol{p_{ti}}$ that feature vector x_t was produced by Gaussian i
   (i.e., $p_{ti} = p(G = i|x_t) = \frac{p(x_t|G=i)p(G=i)}{p(x_t)} \approx p(x_t|G = i) = g_i(x_t, \mu_i, \sigma_i)$ with g_i being the Gaussian pdf
   and G the unobserved random variable indicating membership to one of the Gaussians)

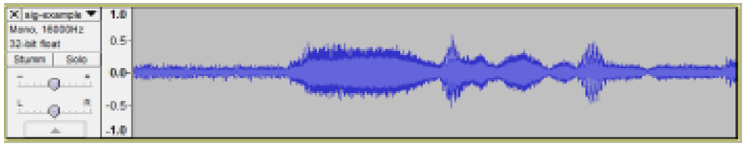
3. **M-Step**: Calculate new **Maximum Likelihood** (ML) estimate of hypothesis,
   assuming the expected values from (2) hold
   Example: **Calculate the** $\boldsymbol{\mu_i}$**,** $\boldsymbol{\sigma_i}^{\boldsymbol{2}}$, given the currently assigned membership
   (i.e., using standard ML estimation: $\mu_i = \frac{1}{T}\sum_{t=1}^{T} p_{ti} \cdot x_t$, $\quad \sigma_i{}^2 = \frac{1}{T}\sum_{t=1}^{T} p_{ti} \cdot (x_t - \mu_i)^2$)

4. Repeat with step 2 until convergence
   Always replacing old estimates with new ones

2. EXAMPLE DOMAIN: SPEECH PROCESSING

The audio signal

The waveform $s[n]$ (a 1D array of N integer samples)

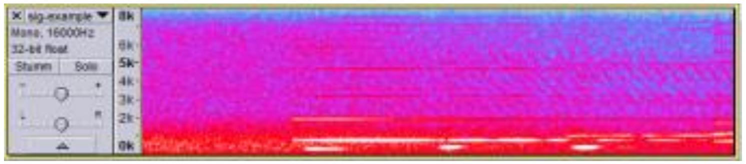


Time domain information (2D: time, amplitude):

- Energy (~loudness): $NRG = \frac{1}{N}\sum_n s[n]^2$
- Zero crossing rate (~prominent frequency for monophonic signals): $ZCR = \frac{1}{N}\sum_n I(s[n] \cdot s[n-1] < 0)$

Frequency domain information (3D: time, frequency, amplitude):

- Time frequency representations via FFT or DWT (phase information typically discarded)



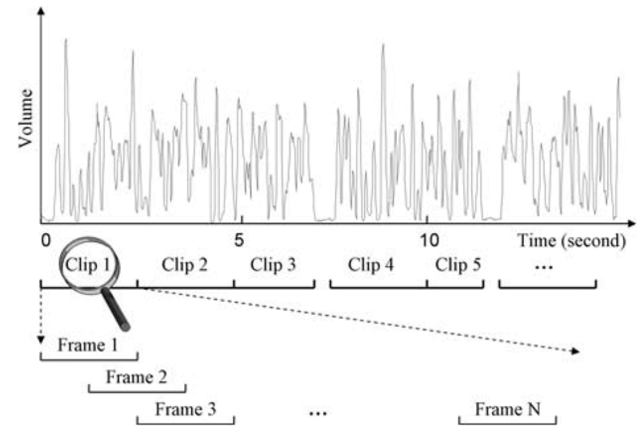
More on signal processing: Smith, *"Digital Signal Processing - A Practical Guide for Engineers and Scientists"*, 2003

Zurich University of Applied Sciences

Frame-based processing
From signal to features

Feature extraction in general
- **Reduction** in **overall** information
- …**while** maintaining or even **emphasizing** the **useful** information

Challenging audio signal properties
- Neither stationary (i.e., statistical figures change over time)
  ➔ **problem** with transformations like Fourier transform **when analyzed in whole**
- …nor conveys its meaning in single samples
  ➔ **problem** when analyzing **per sample**



Source: http://what-when-how.com/video-search-engines/audio-features-audio-processing-video-search-engines/

Solution
- **Chop into** short, usually overlapping chunks called frames
  ➔ extract features per frame
- Prominent parameters: **32ms** frame-**size**, **16ms** frame-**step** (i.e., 50% overlap)
  ➔ Technically a `double` matrix $f[T][D]$
  with $T = 1 + floor\left(\frac{ceil(N-frameSize)}{frameStep}\right)$ the frame count, D the feature dimensionality

Mel Frequency Cepstral Coefficients (MFCC)
The predominantly used multi-purpose audio feature

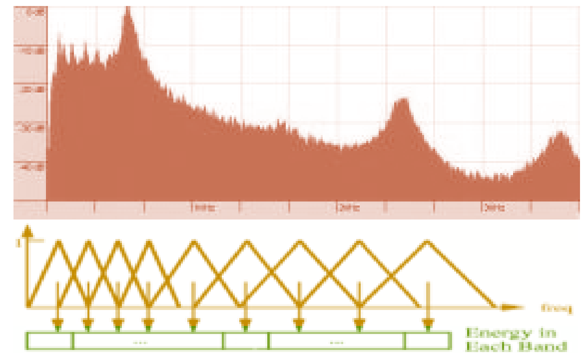
MFCC extraction process

1. Pre-emphasize: $s[n] = s[n] - \alpha \cdot s[n-1]$
   (**boost high** frequencies to improve SNR; α close to 1, e.g. 0.97)

2. Compute magnitude spectrum: $|FFT(s[n])|$
   (i.e., **time-frequency decomposition** neglecting phase)

3. Accumulate under triangular Mel-scaled filter bank
   (resembles **human ear**)

4. Take DCT of filter bank output, discard all coefficients $> M$
   (i.e., low-pass → **compression**; typically $M \in [8..24]$)



Source: http://developer.nokia.com &
http://phys.unsw.edu.au/~jw

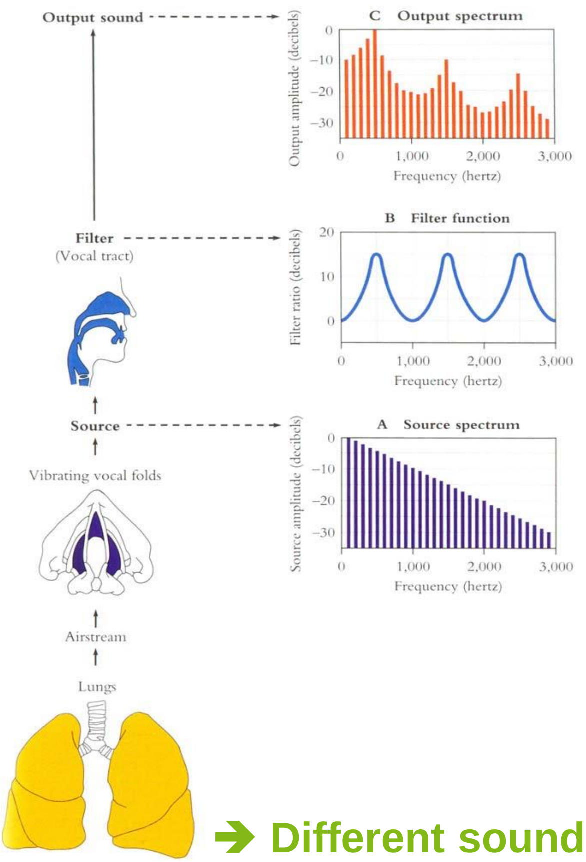
Content and meaning of MFCCs

> A play with the word "spectrum" and the involved math. operation of convolution

- Low-pass filtered spectrum of a spectrum: "Cepstrum"
- Intuitively: A **compact representation of** a frame's **smoothed spectral shape**
  ➔ Convey **most** of the **useful information in** a **speech** or **music** signal, but **no** pitch information

> Pitch: The perceived tone height (i.e., the tone you would whistle, the melody)

The source filter model of speech production

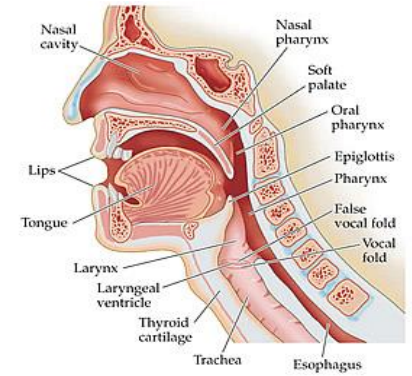


Source

- Air flows from the lungs through the vocal chords
- Produces noise-like (unvoiced) or periodic (overtone-rich, voiced) excitation signal

Filter

- Vocal tract shapes the emitted spectrum



The vocal tract; source: DUKE Magazine, Vol. 94, No. 3, 05/06 2008

Important physiological parameters

- Size of the glottis determines fundamental frequency (F0) range
- Shape of the vocal tract and nasal cavity determines formant frequencies (F1-5), thus "sound“

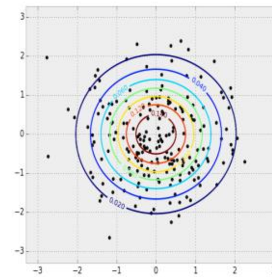
➔ **Different sounds** are **produced by changing** the **source/filter configuration**

Source-filter interaction; source: http://www.spectrum.uni-bielefeld.de/~thies/HTHS_WiSe2005-06/session_05.html

3. GAUSSIAN MIXTURE MODELS

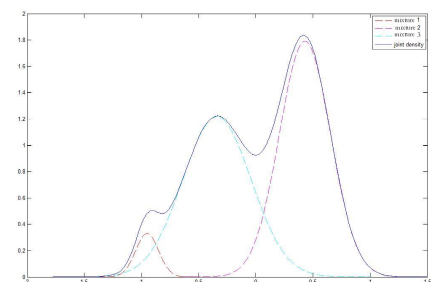
Probabilistic mixture models
Generative models for unknown, multivariate distributions

Mixture Models
- **Approximate** an **arbitrary** distribution **by** a **linear combination of** a **simpler**, “well-behaved” distribution
  ➔ Mathematically **tractable**, **compact** formulation, allows **sampling** & **inference**



Example of a multivariate (2D) Gaussian distribution: samples and contour plot.

The Gaussian Mixture Model (GMM)
- **Modeled by** a weighted sum of $\boldsymbol{N}$ **multivariate Gaussians** (N being sufficiently large)
- Often used because of “**nice**” mathematical **properties** of Gaussian pdf and central limit theorem
  (~ data from natural phenomena tend to be Gaussian distributed)
- The Gaussians’ **parameters** can be estimated efficiently **using** the **EM** algorithm



Example of a multimodal (but univariate) distribution, approximated by a GMM with 3 mixtures.

GMMs as generative models for voice modeling

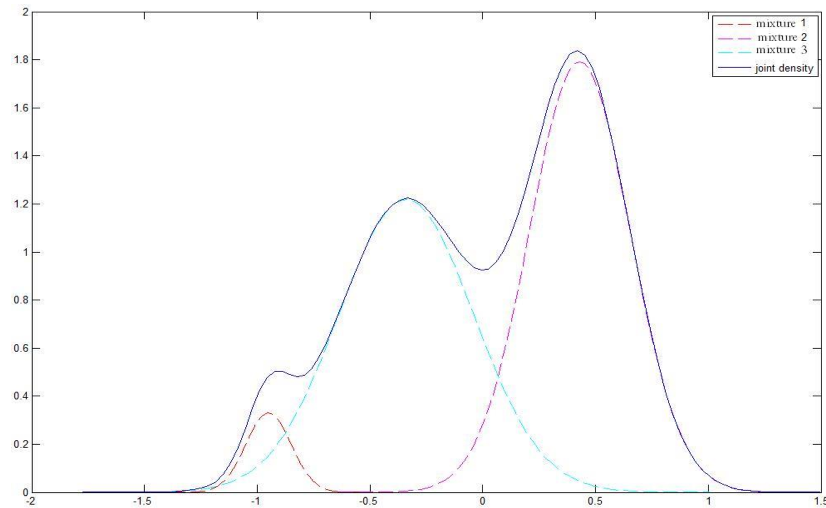
Reference

- Reynolds, Rose, *«Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models»*, 1995

Key ideas

- **Take** the estimated probability density function (**pdf**) $p(x|h)$ of a speaker's D-dim. training vectors x **as** a **model of** his **voice**

- Model the **pdf as** a **weighted sum of** M D-dimensional **Gaussians** (e.g., $M = 32$, $D = 16$)



GMM with 3 mixtures in 1 dimension. Solid line shows **GMM density**, dashed lines show **constituting Gaussian densities**.

GMM rationale

Hybrid solution between non-parametric clusters (vector quantization) and compact smoothing (single Gaussian):

- **Smooth approximation** of arbitrary densities
- **Implicit clustering** into broad phonetic classes

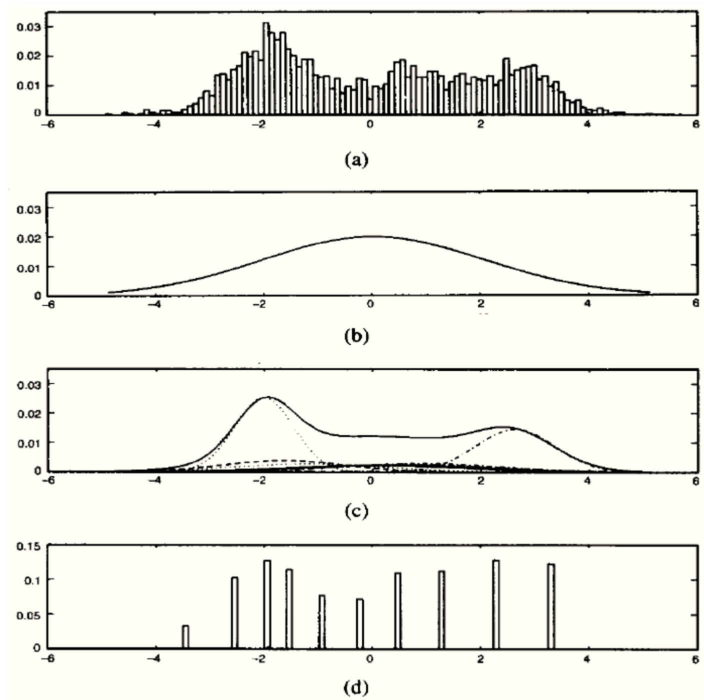
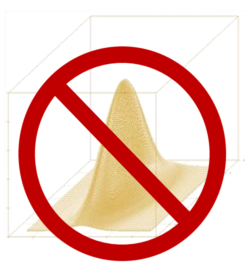


Fig. 3. Comparison of distribution modeling: (a) Histogram of a single cepstral coefficient from a 25 second utterance by a male speaker; (b) maximum likelihood unimodal Gaussian model; (c) GMM and its 10 underlying component densities; (d) histogram of the data assigned to the VQ centroid locations of a 10-element codebook.

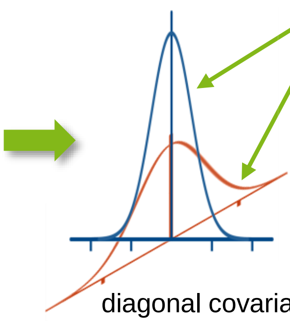
GMM comparison with other techniques; from [Reynolds and Rose, 1995].

Mathematical formulation of the GMM
Using diagonal covariance (→ see appendix for reasons)

Notation
- h: **model** (GMM)
- w: **weight** (scalar)
- μ: **mean** vector
- σ^2: **variance** vector (the diagonal of the covariance matrix Σ)
- g_i: **Gaussian pdf** of i^{th} (out of M) mixtures
- x: **feature vector**
- D: **dimensionality** of x, μ, σ^2
- p: density/**likelihood** of a feature vector given the model



full covariance



diagonal covariance

Formulae
- **Model** consists of: $h = \{w_i, \mu_i, {\sigma_i}^2\}$
  → subject to $i = 1..M$ and $\sum_{i=1}^{M} w_i = 1$

Condition on weights to sum up to 1

- The **multimodal Gaussian** with **diagonal covariance** computes as

$$g_i(x, \mu_i, {\sigma_i}^2) = \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi {\sigma_{id}}^2}} \cdot e^{-\frac{(x-\mu_{id})^2}{2{\sigma_{id}}^2}}$$

Just the product over the assumedly independent marginals (dimensions)

The univariate Gaussian pdf

- Model **evaluation**:

$$p(x|h) = \sum_{i=1}^{M} w_i \cdot g_i(x, \mu_i, \Sigma_i)$$

GMM training via the EM algorithm

Maximum likelihood training

- Initialize model $h = \{w_i, \mu_i, {\sigma_i}^2\}$ using data $X = \{x_1 .. x_T\}$
  ➔ Instead of pure random initialization, find good start values via clustering (e.g., with k-means)
- **E**-Step:

$$p_{ti}(i|x_t, h) = \frac{w_i \cdot g_i\left(x_t, \mu_i, I_D \cdot {\sigma_i}^2\right)}{\sum_{i=1}^{M} w_i \cdot g_i\left(x_t, \mu_i, I_D \cdot {\sigma_i}^2\right)}$$

- **M**-Step:

$$w_i = \frac{1}{T}\sum_{t=1}^{T} p_{ti}(i|x_t, h)$$

$$\mu_i = \frac{1}{T \cdot w_i}\sum_{t=1}^{T} p_{ti}(i|x_t, h) \cdot x_t$$

$${\sigma_i}^2 = \left(\frac{1}{T \cdot w_i}\sum_{t=1}^{T} p_{ti}(i|x_t, h) \cdot {x_t}^2\right) - {\mu_i}^2$$

→ see appendix

The (properly normalized) probability of x_t being issued by mixture i

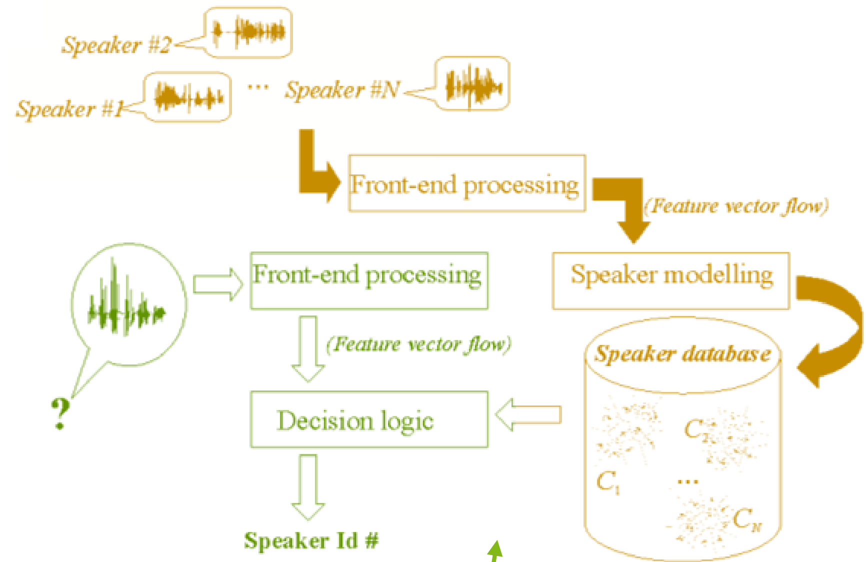
Mixture i's weight is just the mean probability of all training vectors being assigned to it

Alternative: Training via maximum a posteriori (MAP) adaptation (i.e. uses a priori knowledge)
➔ see Reynolds, Quatieri, Dunn, «*Speaker Verification Using Adapted Gaussian Mixture Models*», 2000

The task of speaker recognition

Speaker recognition

- **Tell identity** of an **utterances**' speaker
- Typical: score feature-sequence against a speaker model



Three subsequently more complex settings

- Verification: Verify that a **given utterance fits** a **claim**ed identity (model) or not
- Identification: **Find** the actual **speaker among** a list of **prebuild models**
  (or declare as unknown: open set identification)
- Diarization (a.k.a. tracking, clustering): **Segment** an audio-**stream by voice** identity
  (who spoke when, no prior knowledge of any kind)

Doing speaker identification

Finding the speaker s of a new utterance, given a set of trained speaker models

- Utterance represented by its feature vector sequence $X = \{x_1..x_T\}$
- Speakers models given by $\{h_1..h_S\}$

$$s = \arg\max_s p(X|h_s)$$

$$= \arg\max_s \prod_{t=1}^{T} p(x_t|h_s)$$

The prob. of a set of feature vectors is the product of the individual probs (**independence assumed**)

$$= \arg\max_s \sum_{t=1}^{T} \log p(x_t|h_s)$$

Using the **log** turns the product into a sum ➔ makes the computation **numerically stable**

Model comparison via generalized likelihood ration (GLR)

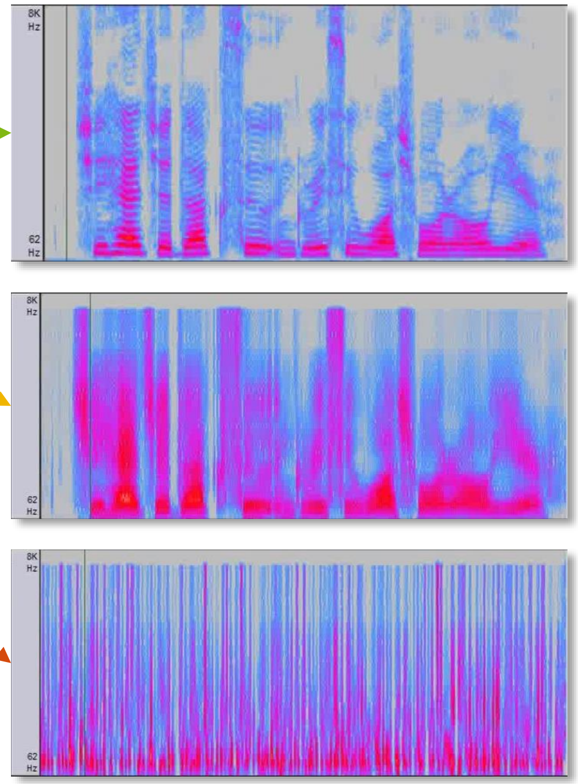
- Absolute likelihood values are not meaningful, but their ratios are
  ➔ To decide if given models h_1, h_2 trained on utterances X_1, X_2 are actually of the same speaker, threshold GLR **distance measure**:

$$GLR(h_1, h_2) = \log\left(\frac{p(X_1|h_1) \cdot p(X_2|h_2)}{p(X_1 \cup X_2|h_{1\cup 2})}\right)$$

What GMMs do not capture

Re-synthesizing speech from intermediate stages of the speaker modeling pipeline

- Original utterance
- Resynthesized feature vectors (MFCCs)
- Resynthesized MFCCs from GMM



Implication

- **Temporal context isn't modeled** by GMMs

More on temporal context modeling:

- Friedland, Vinyals, Huang, Müller, *«Prosodic and other Long-Term Features for Speaker Diarization»*, 2009
- Stadelmann, Freisleben, *«Unfolding Speaker Clustering Potential – A Biomimetic Approach»*, 2009
- **Lukic, Vogt**, Dürr, Stadelmann, «***Speaker Identification and Clustering using Convolutional Neural Networks***», 2016

Where's the intelligence?
Man vs. machine

- Using **probability** theory and statistics to make an agent work in a world of uncertain events **is a very good idea**
  ➔ But: As we already saw with logic, full implementation **without heuristics** is computationally **intractable**

- Particularly in speech processing, **simplifying assumptions** like independence among subsequent feature vectors are **utterly unrealistic**
  ➔ Results of respective systems are clearly worse than human performance
  ➔ But: We have been able to **work around** this using **deep feature learning** [Lukic et al., 2016]

Review

- Understanding uncertain events as random variables gives us a potent arsenal of tools for modeling: E.g., **probability density function** (pdf) of a random variable **tells us everything** there is to know about this function
- Thus, **estimating** the **pdf** is a rewarding **target for** (unsupervised) **learning**

- **Bayes' theorem** is used to turn priors (i.e., prior knowledge) into posteriors (i.e., taking all evidence & priors into account)

- **Speaker recognition** comes in the flavors of **verification**, **identification** or **diarization**
- The classic approach is **MFCC** features and **GMM** models
- Optimal parameters are best found using **best practices** (→ see appendix)

- **EM** training **iterates between** estimating updates values of hidden variables (based on assumed parameters of the sought distribution – **E-step**), and updating these parameters (based on these new estimates – **M-step**)

APPENDIX

Other forms of Bayesian learning

The Naïve Bayes classifier

Basic idea

- The straightforward way of applying Bayes' theorem to yield a MAP hypothesis is intractable (too many conditional probability terms need to be estimated)
- **Simplification**: Assume **conditional independence** among features given target value
  $h(x_i) = \underset{y_j \in Y}{\operatorname{argmax}} P\left(y_j \middle| x_{i1}, x_{i2}, \dots, x_{iD}\right) = \underset{y_j \in Y}{\operatorname{argmax}} P\left(x_{i1}, x_{i2}, \dots, x_{iD} \middle| y_j\right) \cdot P\left(y_j\right) = \underset{y_j \in Y}{\operatorname{argmax}} P(y_j) \cdot \prod_{d=1..D} P(x_{id}|y_j)$

➔ Very successful in text classification (e.g., SPAM filtering, news classification)

Example (from https://alexn.org/blog/2012/02/09/howto-build-naive-bayes-classifier.html)

- Imagine 74 emails: 30 are SPAM; 51 contain *"penis"* (of which 20 are SPAM); 25 contain *"Viagra"* (24 are SPAM)
- Bayes classifier: $p(SPAM|\text{penis}, \text{viagra}) = \frac{p(penis|SPAM \cap viagra) \cdot p(viagra|SPAM) \cdot p(SPAM)}{p(penis|\text{viagra}) \cdot p(viagra)} = \cdots$

  $p(A \cap B) = P(B|A) \cdot P(A)$

  ➔ **intractable** with more words because of **cond. prob. terms** also get numerically small

- Naïve Bayes classifier: $p(SPAM|\text{penis}, \text{viagra}) = \frac{p(penis|SPAM) \cdot p(viagra|SPAM) \cdot p(SPAM)}{p(penis) \cdot p(viagra)} = \frac{\frac{20}{30} \cdot \frac{24}{30} \cdot \frac{30}{74}}{\frac{51}{74} \cdot \frac{25}{74}} = 0.928$

Other forms of Bayesian learning (contd.)

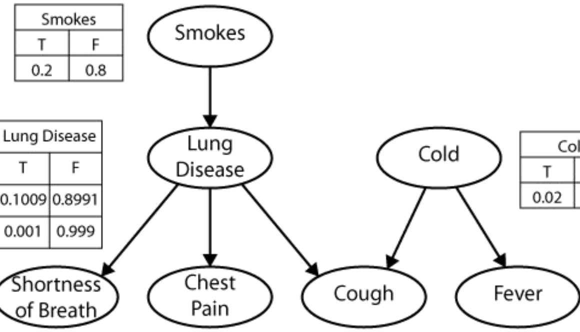
The Bayes net (or Bayesian belief network)

In a nutshell

- **Loosens naïve Bayes constraint**: Assumes only conditional independence among certain sets of features
- Model of joint probability distribution of features (also unobserved ones):
  ➔ a **directed acyclic graph** for independence assumptions and local conditional probabilities
- Inference possible for any feature / target, based on any set of observed variables
  ➔ has to be done approximately to be tractable (NP-hard)
- **Use case**: conveniently **encode prior causal knowledge** in form of conditional (in)dependencies

Example (from Goodman and Tenenbaum, *"Probabilistic Models of Cognition"*, http://probmods.org)

- A simple Bayes net for medical diagnosis
- One node per random variable
  ➔ Attached is a conditional probability table with the distribution of that node's values given its parents
- A Link between 2 nodes if there is a direct conditional (causal) dependence



Smokes T	Smokes F
0.2	0.8

Smokes	Lung Disease T	Lung Disease F
T	0.1009	0.8991
F	0.001	0.999

Cold T	Cold F
0.02	0.98

Lung Disease	Shortness of Breath T	Shortness of Breath F
T	0.208	0.792
F	0.01	0.99

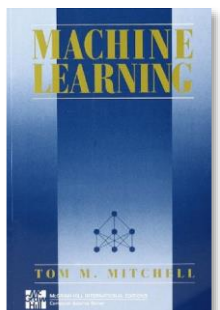
Lung Disease	Chest Pain T	Chest Pain F
T	0.208	0.792
F	0.01	0.99

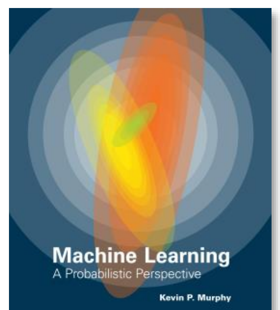
Lung Disease	Cold	Cough T	Cough F
T	T	0.7525	0.2475
T	F	0.505	0.495
F	T	0.505	0.495
F	F	0.01	0.99

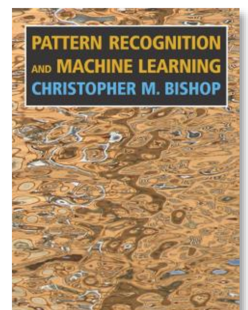
Cold	Fever T	Fever F
T	0.307	0.693
F	0.01	0.99

More on Bayesian learning

- http://fastml.com/bayesian-machine-learning/: Brief overview, explanations and references
- [Mitchell, 1997], ch. 6: Concise introduction to Bayesian learning
- http://www.cs.cmu.edu/~tom/mlbook/NBayesLogReg.pdf: New chapter for [Mitchell, 1997]
- [Murphy, 2012] and [Bishop, 2006]: Two text books embracing the Bayesian perspective
- Reynolds, Rose, *«Robust Text-Independent Speaker Identification using Gaussian Mixture Speaker Models»*, 1995



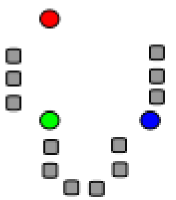




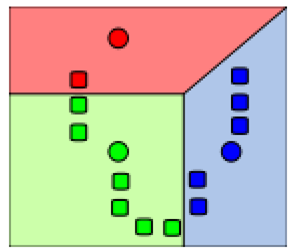

Robust Text-Independent Speaker Identification
Using Gaussian Mixture Speaker Models

k-means clustering in a nutshell
Source: https://en.wikipedia.org/wiki/K-means_clustering

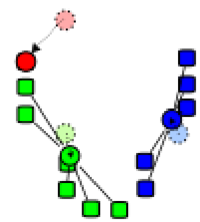
The standard algorithm: non-probabilistic EM



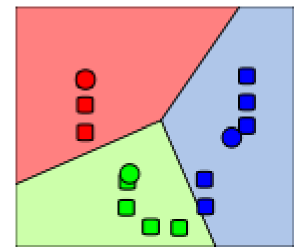
1. k initial "means" (in this case $k = 3$) are randomly generated within the data domain (shown in color).



2. k clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.



3. The centroid of each of the k clusters becomes the new mean.



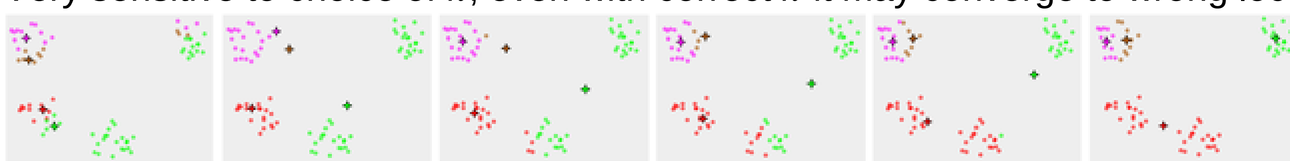
4. Steps 2 and 3 are repeated until convergence has been reached.

…

Properties

- Problems: Very sensitive to choice of k; even with correct k it may converge to wrong local minimum

successive rounds →



- Variants: k-medoids (centroid to be member of data set), k-maxoids (for extremes rather than means)

Glossary of abbreviations

FFT – fast Fourier transform
- Standard algorithm to transform a time-domain (time-amplitude) signal into the frequency domain (frequency-amplitude)

DWT – discrete wavelet transform
- Another transformation to the frequency domain, with higher resolution for higher frequencies

DFT – discrete Fourier transform
- The theoretical basis for the FFT algorithm on an array of samples

SNR – signal to noise ratio
- Amplitude of actual signal (what I want to hear) divided by amplitude of any noise (e.g., background music)

DCT – discrete cosine transform
- As DWT, but decomposes the signal solely based on cosine terms (DFT: sine & cosine)

Mel – from the word “melody”
- Unit to measure the pitch of a sound on a scale where an increase in Mel corresponds to the same increase in perceived pitch

SVM – support vector machine
- An often very well-performing supervised machine learning method: give it data (in form of independent feature vectors) of two classes and it learns the discriminative boundary between them

Glossary of abbreviations (contd.)

ATC – audio type classification

BIC – Bayesian information criterion

- Single-value measure to automatically trade-off model complexity and recognition performance

μ – mean vector

- As estimated on a set of vectors

Σ – covariance matrix

- As estimated on a set of vectors; μ and Σ together determine the multivariate Gaussian distribution

δ – delta coefficients vector

- First temporal derivative of some feature, e.g., a MFCC coefficient: $\delta_{d_t} = MFCC_{d_t} - MFCC_{d_{t-1}}$

δδ – delta delta coefficients vector

- Second temporal derivative, i.e. $\delta\delta_{d_t} = \delta_{d_t} - \delta_{d_{t-1}}$ for the d^{th} dimension and t^{th} time step

AANN – Auto-associative neural network (a.k.a. autoencoder)

- Supervised machine learning method that learns to reproduce its input on the output through a sort of bottleneck (e.g., compression) layer

Glossary of abbreviations (contd.)

LPC – linear predictive coding

- Representing a value of a time series as a linear combination of the last few samples

dB – Dezibel

- Logarithmic unit to express the ratio of two physical quantities, e.g. power or intensity with reference to a “zero” level; a Dezibel is a tenth of a Bel. [after Wikipedia]

mp3 – MPEG-1 or MPEG-2, Audio Layer III

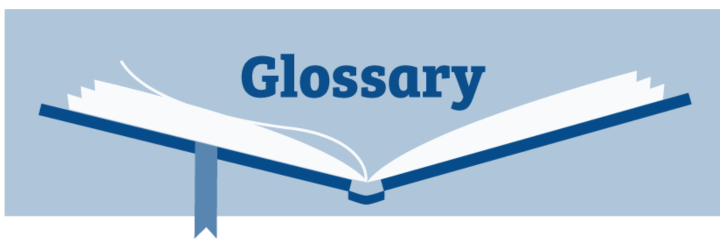
- Lossy audio compression relying heavily on results of psychoacoustics (e.g., masking effects): what can’t be heard doesn’t need to be coded

ASR – automatic speech recognition

- Joint name for all technologies used to analyze and comprehend human speech with machines

VQ – vector quantization

- A method to represent a set of vectors by a few «representative» vectors (called the «codebook»)

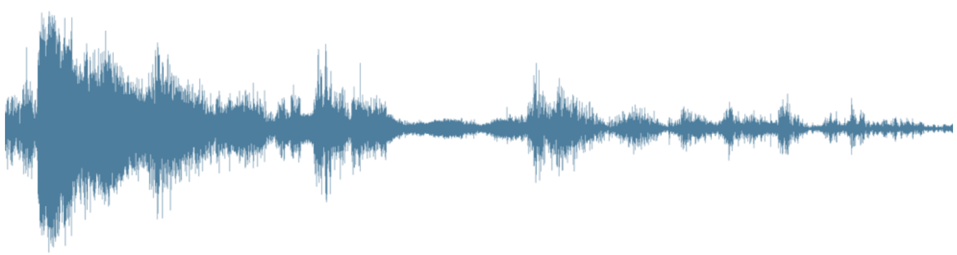


Properties of audio signals
Their content and segmentation

The sample array $s[n]$ is just 1D

But: Sound still carries **information on** many **different layers** or "dimensions“
- Silence ⇔ non-silence
- Speech ⇔ music ⇔ noise
- Voiced speech ⇔ unvoiced speech
- Different musical genres, speakers, dialects, linguistic units, polyphony, emotions, . . .

Definition of audio **segmentation**
- Temporally separate one ore more of the above types from each other into consecutive segments by more or less specialized algorithms



Properties of the speech signal

Slowly time-varying
- **stationary** over sufficiently **short period** (5-100ms, phoneme)

Speech range: 100 - 6800Hz (telephone: 300 - 3400Hz)
- 8kHz sample rate sufficient, 16kHz optimal

Speech frames convey **multiple information**:
- Linguistic (phonemes, syllables, words, sentences, phrases, …)
- Identity
- Gender
- Dialect
- …

➔ fractal structure

Properties of the human auditory system

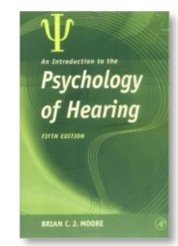
High **dynamic range** ($120dB$, $q_{dB} = 10 \cdot \log_{10}(q/q_{ref})$ for some quantity q)
- Work in the log domain (increase in $3dB$ ➔ loudness doubled)

Performs **short-time spectral analysis** with log-frequency resolution
- Similar to wavelet-/Fourier-transform ➔ Mel filter bank

Masking effects
- That's what makes mp3 successful in compressing audio

Channel decomposition via "auditory object recognition"
- That's what a machine can not do (except **Melodyne**, and nobody knows why)

…and lots of further interesting material
- But no direct/simple applicability to ASR at the moment



→ More on the auditory system: Moore, *"An Introduction to the Psychology of Hearing"*, 2004

More speech features

Directly from source-filter decomposition

Represent source characteristics via **pitch & noise**

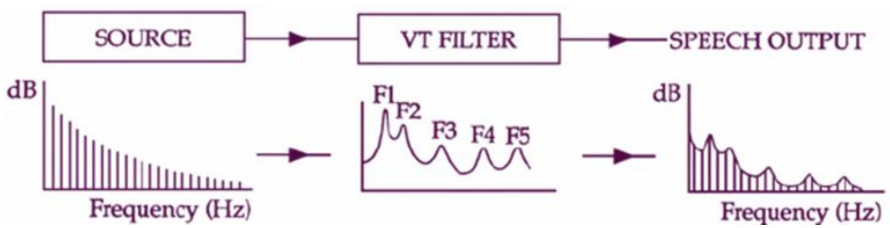
- 1 `double` per frame

Represent filter characteristics with filter coefficients a_k from **LPC analysis**

- $8 - 10$ `double` per frame
- $s[n] = \sum_{k=1}^{p} a[k] \cdot s[n-k] + e[n]$ ($e[n]$ being the residual)
- Btw.: This is the way it is done in mobile phones

LPC coefficients are also applied as speaker specific features

- Sometimes after further processing
- But **typically**, **MFCCs** are used



Source: Keller, “The Analysis of Voice Quality in Speech Processing”, 2004

GMM best practices

- Use **log-likelihoods** instead of likelihoods
  ➔ Likelihoods become so small that one ends up with numerical instabilities otherwise
- Use a **diagonal covariance** matrix
  ➔ Simpler/faster training, same/better results due to more compact model (with more mixtures)
- Use a **variance limit** and beware of **curse of dimensionality**
  ➔ Prohibit artifacts through underestimation of components

- Use **16-32 mixtures** and a minimum of **30s of speech** (ML)
- Adapt only means from 512-1024 mixtures per gender (MAP)
  - Score only with top-scoring mixtures

- **Find optimal number of mixtures** for data via brute force and BIC

- **Compare** models via
  - **Score-wise** (more precise): Generalized Likelihood Ratio (GLR)
  - **Parameter-wise** (faster): Earth Mover's Distance (EMD) or this paper:
    Beigi, Maes, Sorensen, *«A distance measure between collections of distributions and its application to speaker recognition»*, 1998