Speech recognition

- For graduate and post-graduate students
- Home page: *https://mycourses.aalto.fi* **> ELEC-E5510**
- Registration at Oodi
- **Lectures: Mikko Kurimo**
- **Exercises: Juho Leinonen, Aku Rouhe, Katja Voskoboinik, Dejan Porjazovski**
- **Project works: Katja &** Juho, Aku, Anja, Dejan, Ragheb, Anssi, Hemant, Tamas, Georgios

Goals

- Become familiar with speech recognition methods and current applications
- Learn the structure of a typical speech recognition system
- Learn to construct one in practice

Discussion (in breakout groups):

- What is your name – where do you come from?
- What is your goal – why are you here?

Content today

1. **General organization of the course**
2. What is automatic speech recognition (ASR)?
3. Speech as an acoustic signal
4. GMMs and DNNs
5. Home exercise 1:
   - Build a system to classify speech features into phonemes
6. Kick-start of the group works

Content of the course

Week 1
Features

Week 2
Phonemes

Week 3
Words

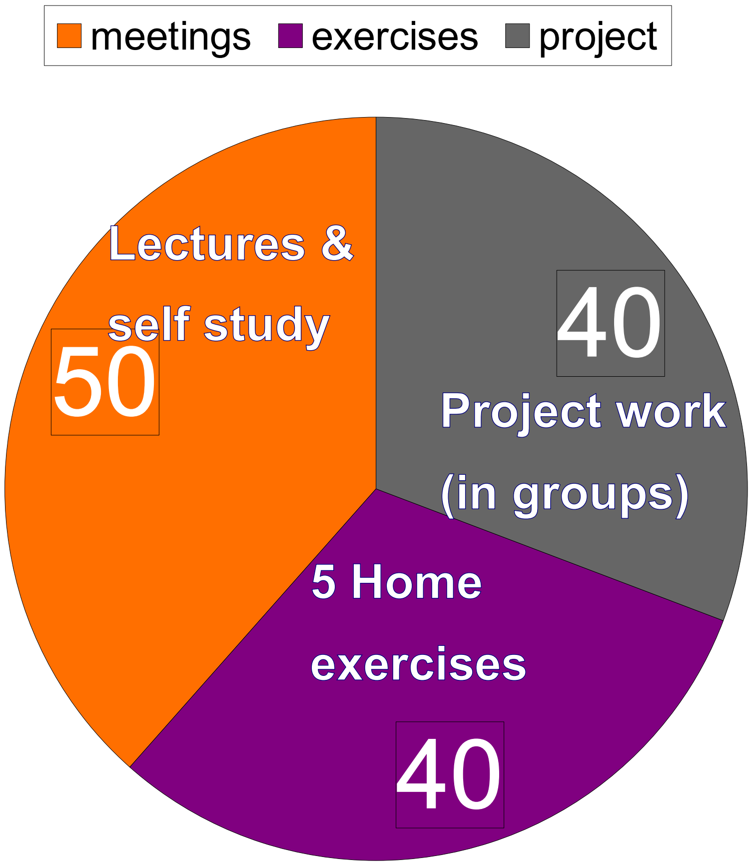
Week 3
Language

Week 4: ASR systems and decoding
Week 5: End-to-end ASR

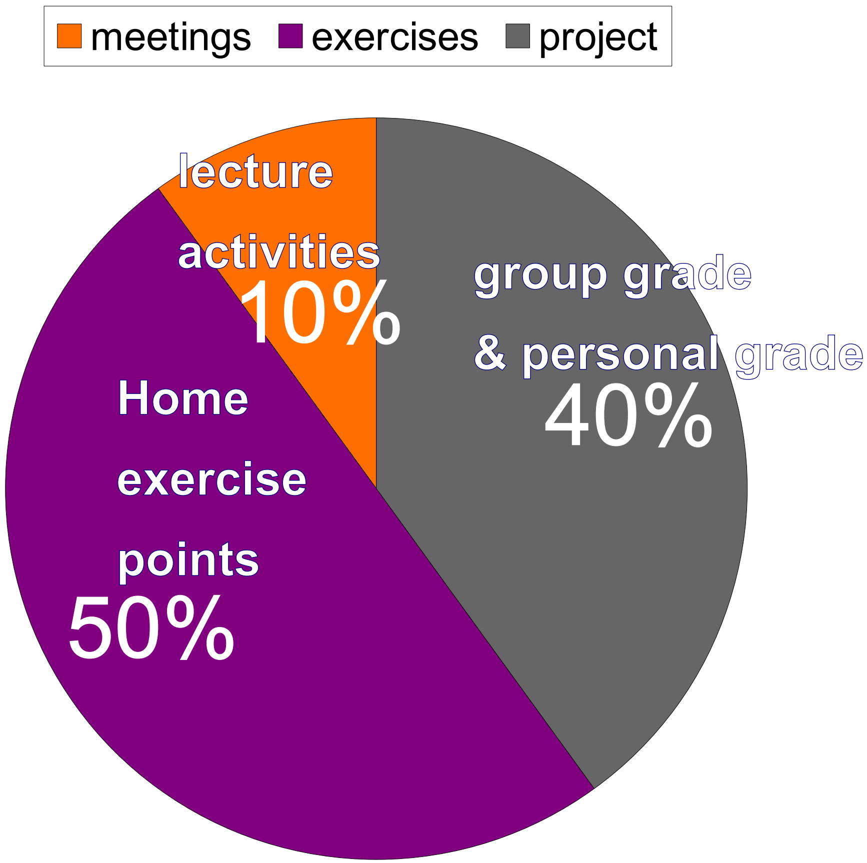
Weeks 6 – 7
Group presentations & reports

Course Format

"Hour pie": 130h



"Grade pie": 1..5



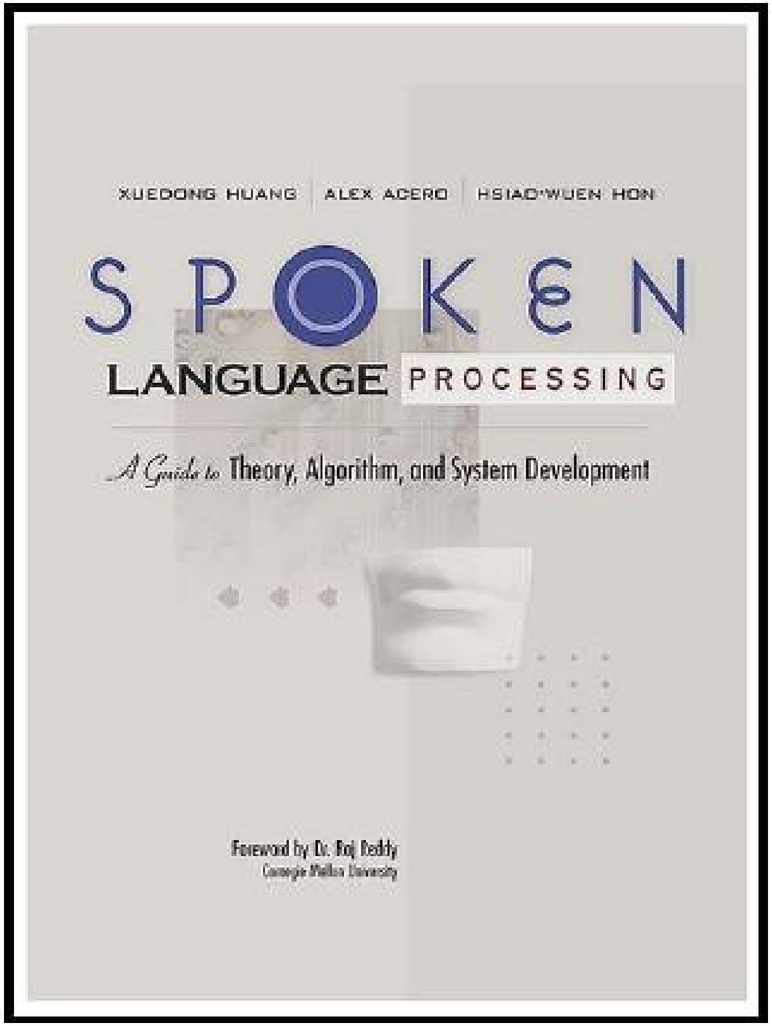
Meetings

- 14 meetings during 7 weeks:
  - 5 general lectures on Wed (Nov 3 - Dec 1)
    - Lecture 09:15 - 11:00 + project meeting slot 11:15 - 12
  - 5 meetings for computer work (Nov 4 - Dec 3)
    - Maari-B Thu 10:15 – 12 & Zoom Fri 14:15 – 16
    - The content of Thu and Fri are identical. Maari-B fits 29.
  - 4 seminar meetings for project results (Dec 8 - Dec 17)
    - Wed 09:15 - 12 and Fri 14:15 - 16

Timeline in the course

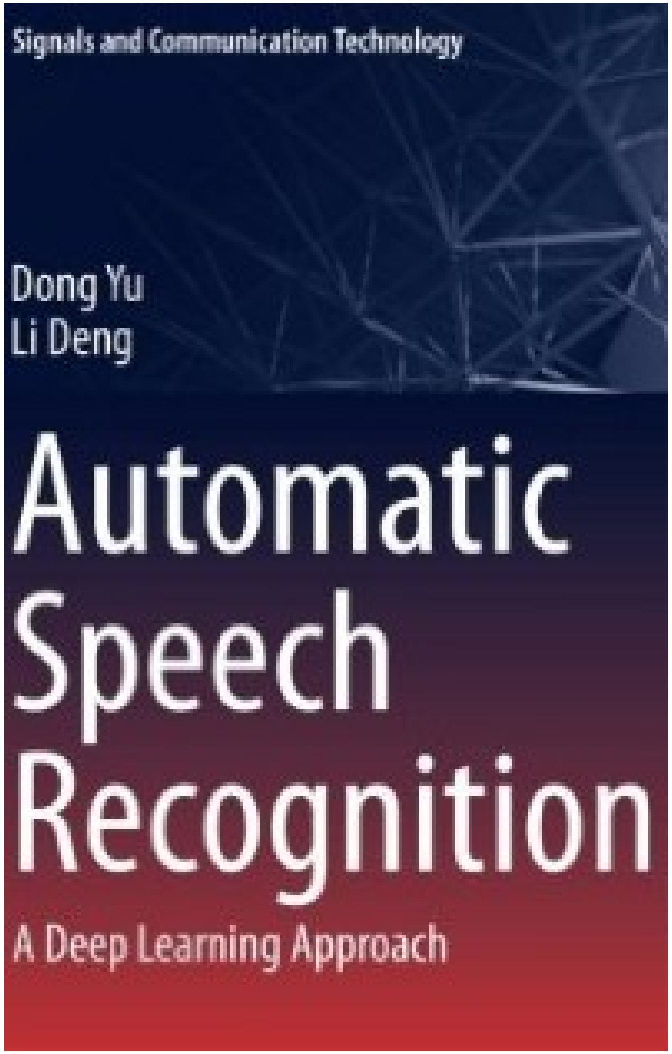
	Meetings Wednesdays	Thursdays or Fridays	Home exercises	Project work status
Week1	Speech features entry test	Classification	Feature classifier	Literature study Meet tutors Wed
Week2	Phoneme modeling	Recognition	Word recognizer	Work plan Meet tutors Wed
Week3	Lexicon and language	Language model	Text predictor	Analysis Meet tutors Wed
Week4	Continuous speech advanced search	LVCSR	Speech recognizer	Experimentation Meet tutors Wed
Week5	End-to-end ASR	End-to-end	End-to-end recognizer	Preparing reports Meet tutors Wed
Week6	Projects1	Projects2		Presentations
Week7	Projects3	Projects4 Conclusion		Report submissior

The main text book



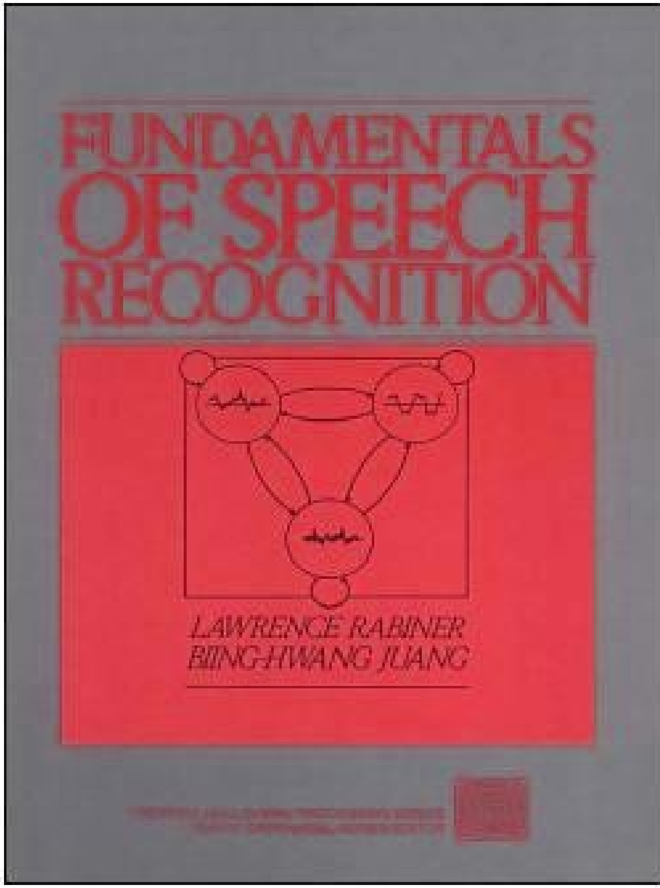
- You may survive without one, but this is recommended
- Huang, Acero: *Spoken Language Processing*
- Prentice Hall, 2001 ISBN: 0-13-022616-5

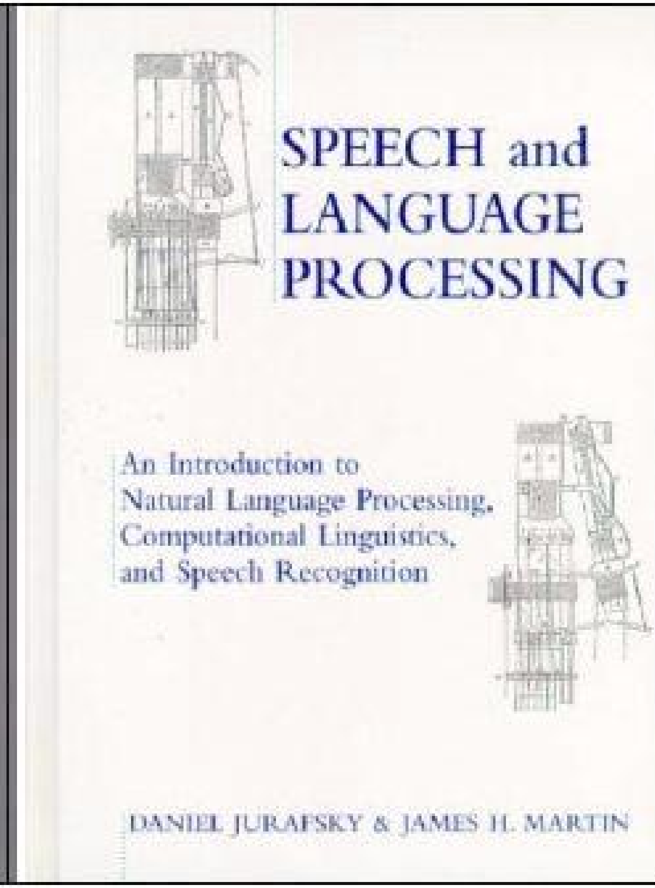
A new text book

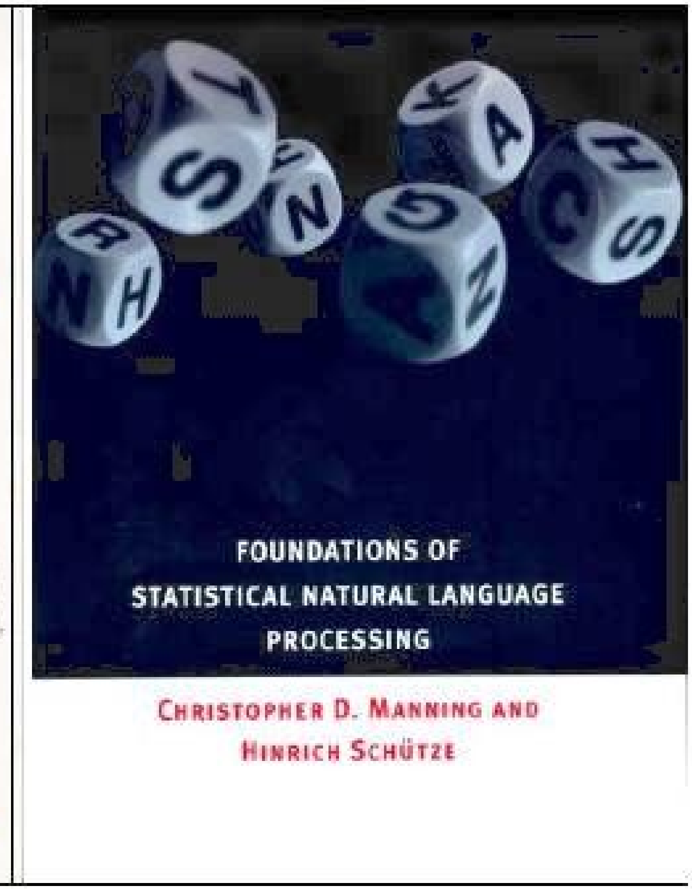


- This is very advanced level, but worth studying to understand the latest trends
- Yu, Deng: *Automatic Speech Recognition A Deep Learning Approach*
- Springer, 2015 ISBN: 978-1-4471-5779-3

Other useful text books







Lectures mapped to pages of Jurafsky & Martin, see:
MyCourses > Materials > (last item in the list)

Some useful online resources

- **Gales, Young: HMMs applications in ASR (book):**
  http://dx.doi.org/10.1561/2000000004
- **Cambridge: HTK Book (detailed manual):**
  http://htk.eng.cam.ac.uk/docs/docs.shtml
- Slides from **MIT** open course: **6.345 ASR** (2003)
  h*ttp://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-345-automatic-speech-recognition-spring-2003/*

  http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-345-automatic-speech-recognition-spring-2003/

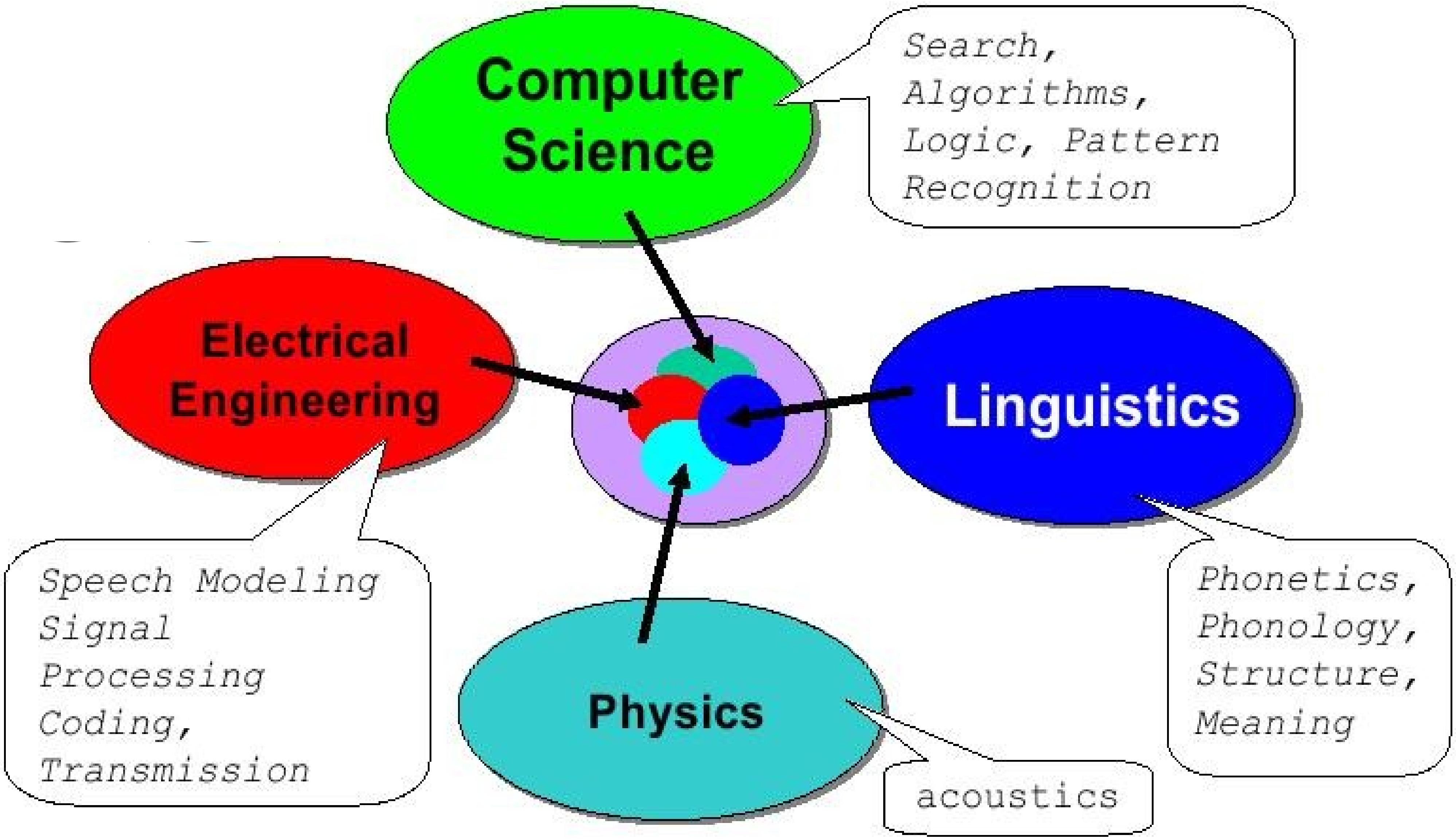
Useful software

- Software used in this course:
  - Python, PyTorch
  - Cambridge HMM toolkit (HTK)
  - SRI language modeling toolkit (SRILM)
- Other useful software for ASR:
  - **Kaldi**, Aachen RWTH, KTH Snack, OGI speech, Nagoya's Julius
  - CMU Sphinx-II ASR, ESPNET, SpeechBrain
  - AaltoASR tools, Aalto Morfessor tools
  - TensorFlow
  - NIST ASR scoring utilities
  - CMU / Cambridge language model toolkit

Content today

1. General organization of the course
2. **What is automatic speech recognition?**
3. Speech as an acoustic signal
4. GMMs and DNNs
5. Home exercise 1:
    - Build a system to classify speech features into phonemes
6. Kick-start of the group works

All these are needed to build ASR systems!



Milestones for ASR systems

- **1952** Bell Labs Digit Recognizer
- **1976** CMU Harpy 1000-word connected recognizer with constrained grammar
- **1980 TKK**: 1000-word LSM recognizer (separate words w/o grammar)
- **1988 TKK**: phonetic typewriter
- **1993** Read texts (WSJ news)
- **1998** Broadcast news, telephone conversations
- **1998** Speech retrieval from broadcast news

Milestones for ASR systems (2)

- **2002** Rich transcription of meetings
- **2004 TKK**: Finnish online dictation for unlimited vocabulary
- **2006** Machine translation of broadcast speech
- **2006** Voice interface in Windows Vista
  - *https://www.youtube.com/watch?v=kX8oYoYy2Gc&feature=related*
- **2008** Google voice search
- **2009 Aalto**: Cross-lingual speaker adaptation by speech recognition
  - *https://www.youtube.com/watch?v=wqv7uYAyAQ0*
- **2011** Siri voice assistant
- **2013** Big performance boost by applying deep learning
- **2017** New end-to-end paradigmas

Performance depends on: 1. Speaking environment, microphone, speaker

1. Office, headset, close-talking
2. Telephone speech, mobile
3. Noise, outside, microphone far away
4. Voice, accents

Acoustic modeling

2. Style of speaking and language

Language modeling

1. Isolated words, small vocabulary
2. Continuous speech, read or planned, large vocabulary
3. Spontaneous speech, open vocabulary, hesitations

https://www.youtube.com/watch?v=UK_2dF9zXl4

Useful entry skills

- linear algebra (basic matrix operations)
- probability and statistics
- signal processing and natural language processing
- programming
  - perl/python for text processing
  - shell scripts, C/C++ for running/modifying programs
  - matlab
- familiarity with Linux

Test of your skill level

Individual test for everyone, now:

1. Go to ***https://kahoot.it*** with your phone/laptop
2. Type in the number you see in the chat
3. Give your **surname** (this will give you an activity point)
4. Answer the questions by selecting **only one** of the options

- There may be several right answers, but just pick one
- 1 min time per question
- This first test is not graded, everyone will get one point

Useful skills - 1

- linear algebra (basic matrix operations)
  - multiplication, determinant, transpose, inverse

$$f(X = x \mid \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-u)\right]$$

Determinant of covariance matrix

Distribution covariance matrix

Distribution mean vector

Observed vector of random variables (features)

Useful skills - 2

- probability and statistics

Bayes' Rule allows us to update prior beliefs with new evidence,

Likelihood

$$P(W_i)\left[\frac{P(O\,|\,W_i)}{\sum_j P(O\,|\,W_j)P(W_j)}\right] = P(W_i\,|\,O)$$

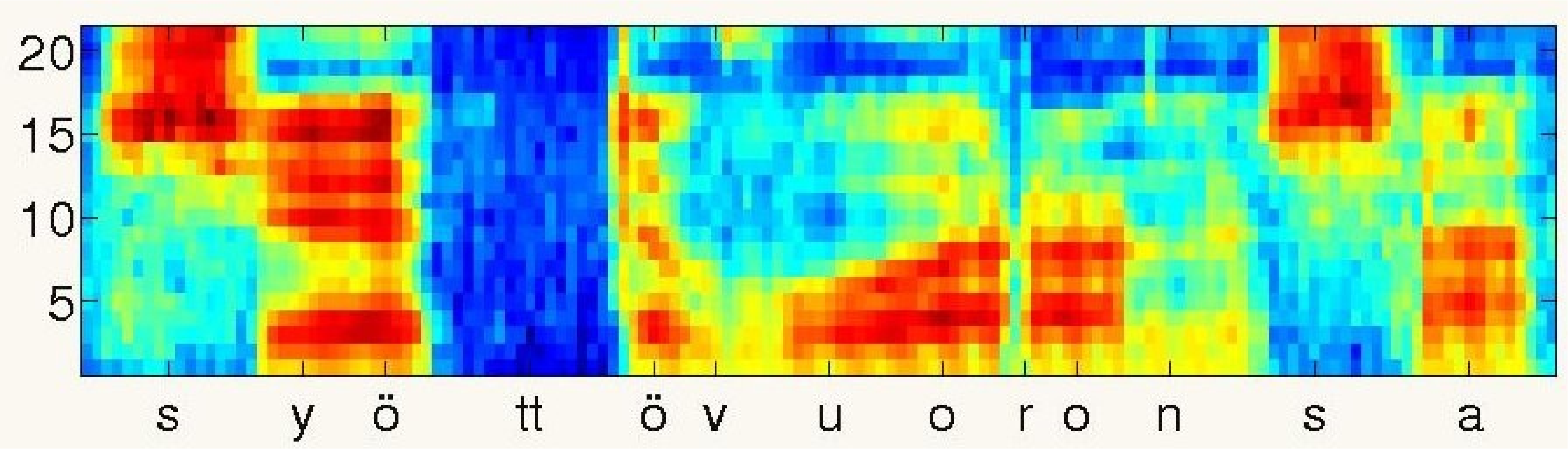
Prior Probability

Evidence
P(O)

Posterior Probability

Useful skills – 3

- signal processing and natural language processing
- Examples:
  - Spectrum and spectrogram of a signal
  - count the frequency of all word pairs in text



Useful skills - 4

- programming
  - Matlab
  - Linux
  - shell scripts, C/C++, perl/python
- example tasks:
  - Use Matlab toolboxes to compute a spectrum
  - Run programs in Linux and store their output in a file
  - Make a script to run commands many times in loop using increasing parameter values
  - Make a simple program to compute the error rate between the speech recognition result (a string) and the reference text

Content today

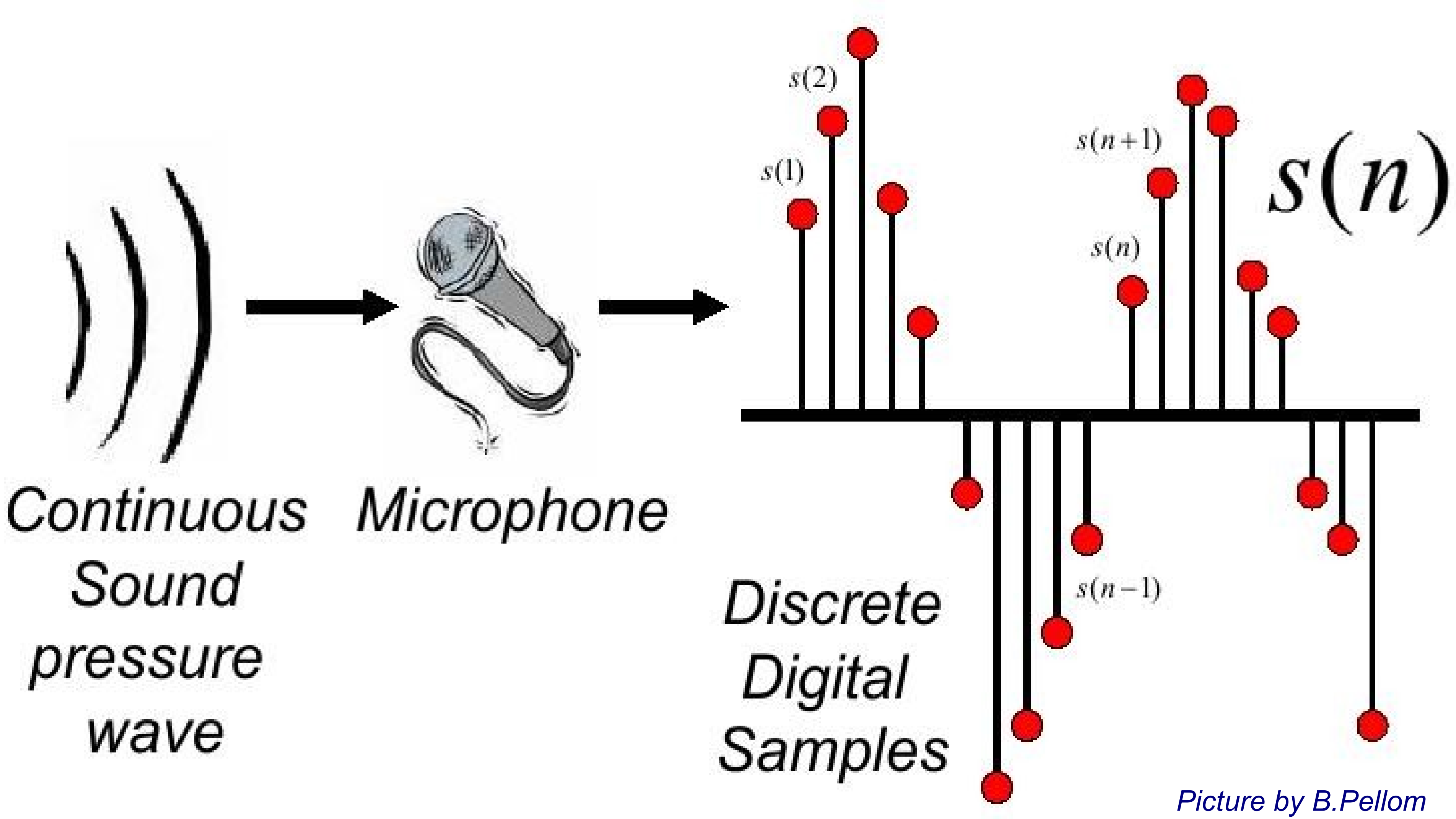
1. General organization of the course
2. What is automatic speech recognition?
3. **Speech as an acoustic signal**
4. GMMs and DNNs
5. Home exercise 1:
   - Build a system to classify speech features into phonemes
6. Kick-start of the group works

Break 5 min

What is speech recognition?

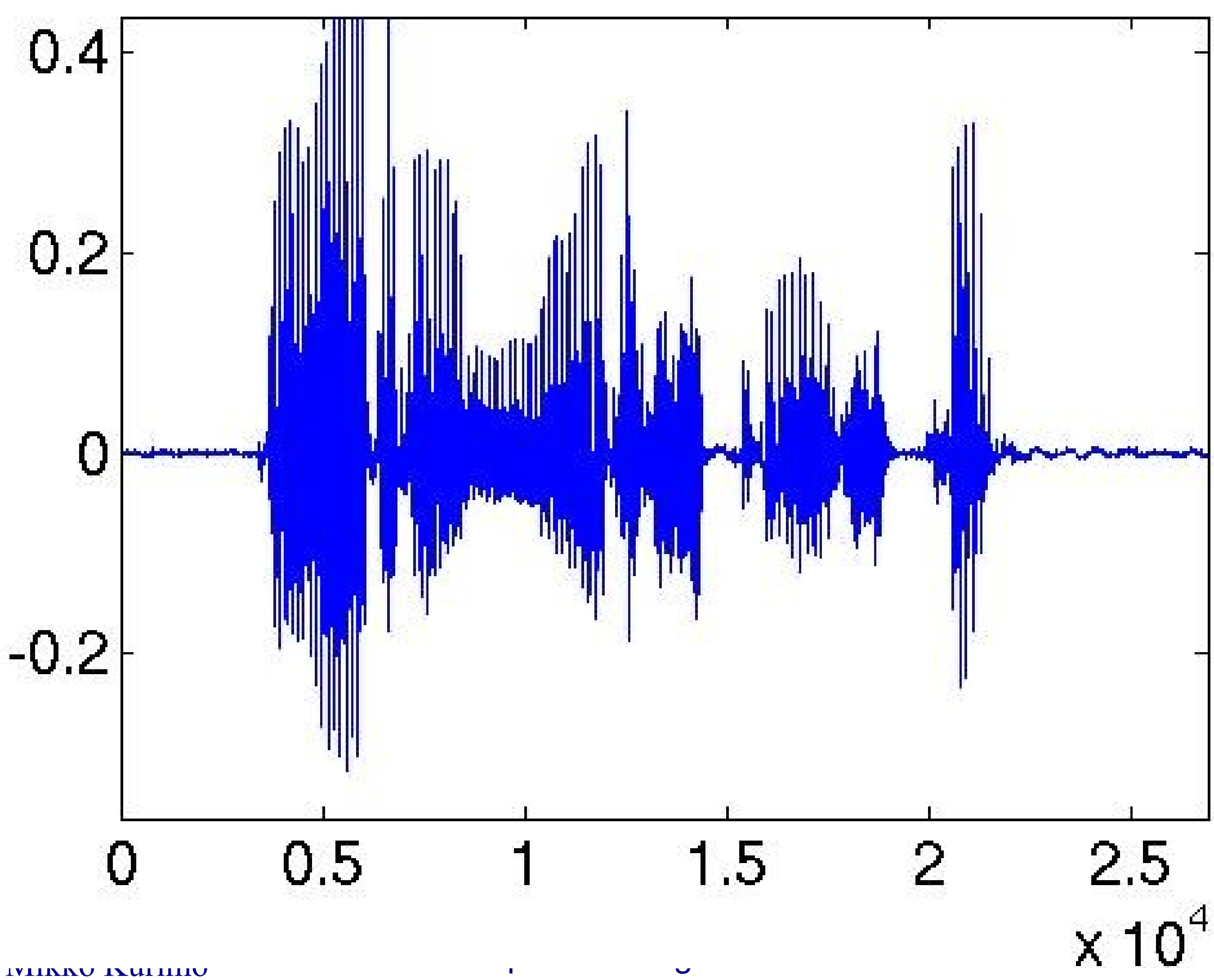
- **Find the most likely word or word sequence given the acoustic signal and our models!**
- **Language model** defines words and how likely they occur together
- **Lexicon** defines how words are formed from sound units
- **Acoustic model** defines the sound units independent of speaker and recording conditions

Speech recording

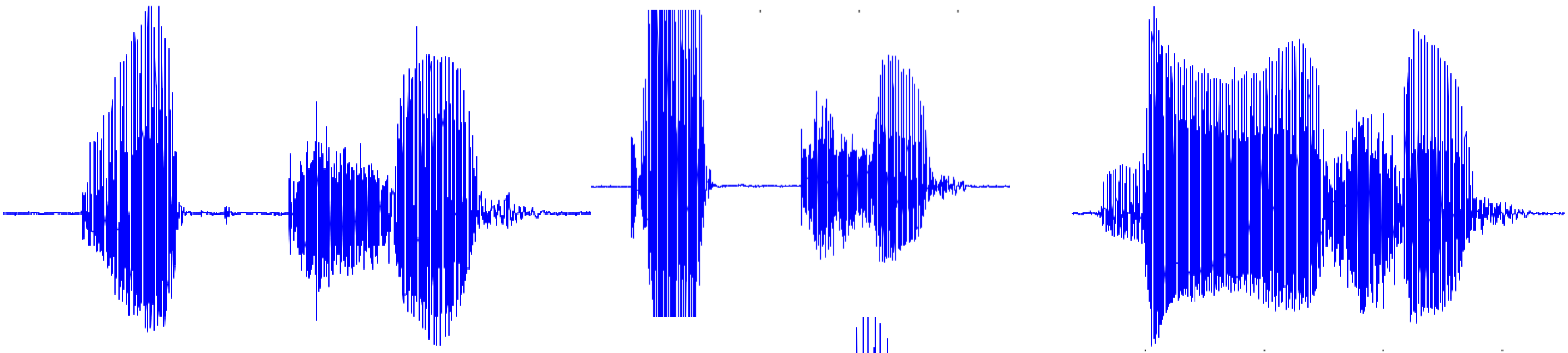


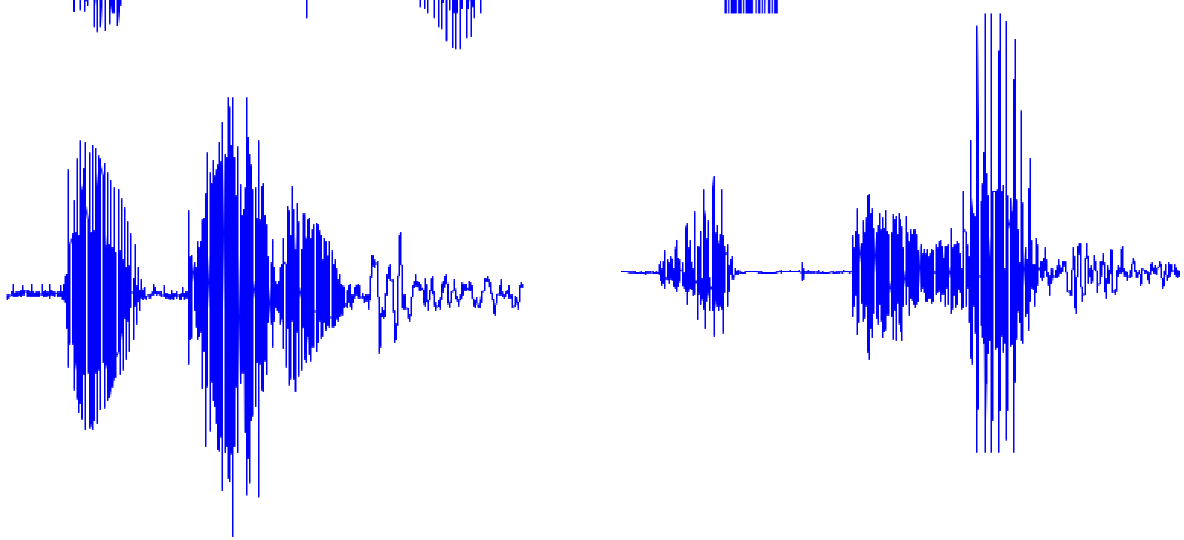
Picture by B.Pellom

A sample of speech waveform



Other words, other speakers





Very difficult to
recognize speech
from the picture...

Modelling the speech signal

Discussion: What separates speech from all the other sounds that the microphone has recorded?

- computer noise, car noise, human movements, other sounds from the mouth, ...
- so, what is special in speech and common in all speaking situations

Why these discussions?
Learning happens, when:
+ brains are active and alert
+ new knowledge contradicts your old beliefs

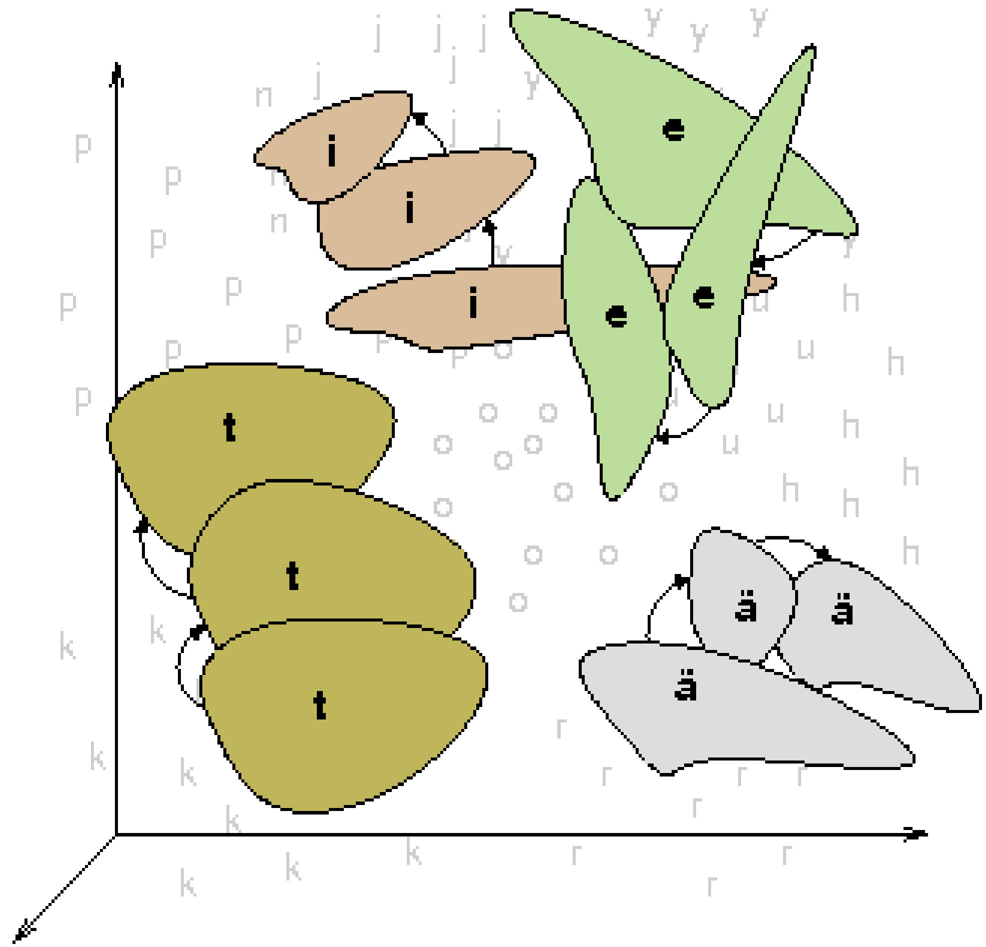
How to recognize speech?

A simple procedure:

- Measure some **characteristic features** of the signal and estimate statistical models for them

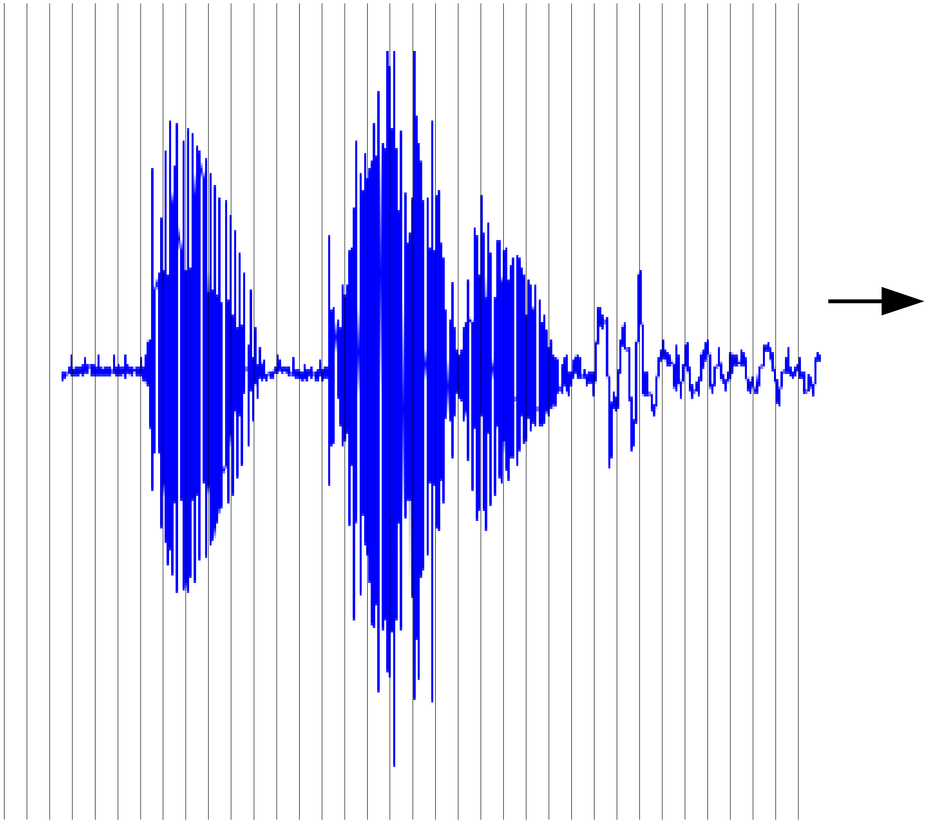
Good features should be:

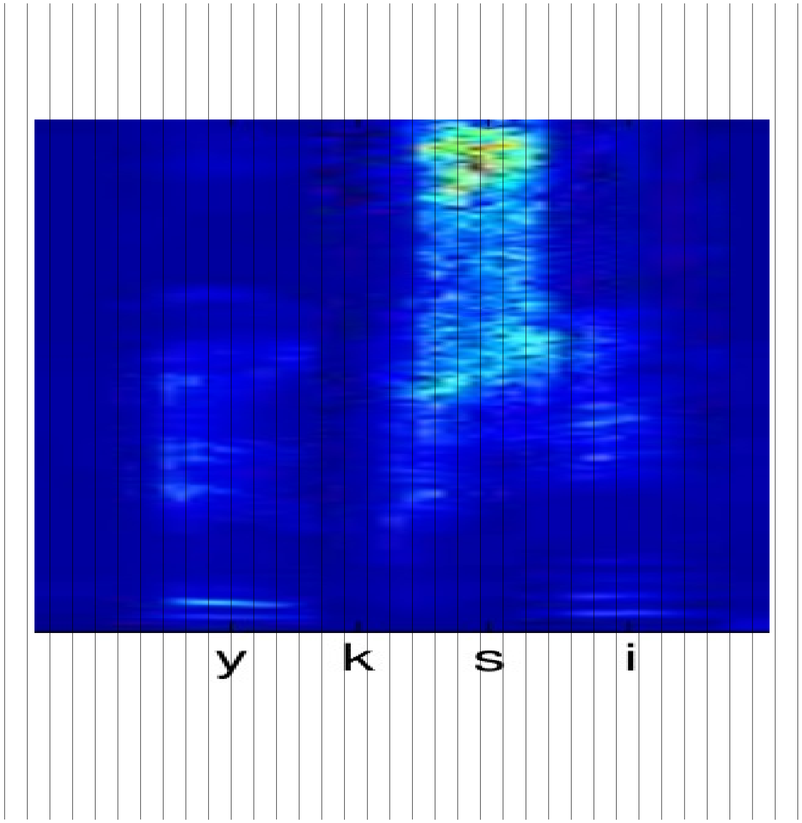
- Compact
- Discriminative for speech sounds
- Fast to compute
- Robust for noise



Frequency analysis

Calculate the spectrum in short time intervals





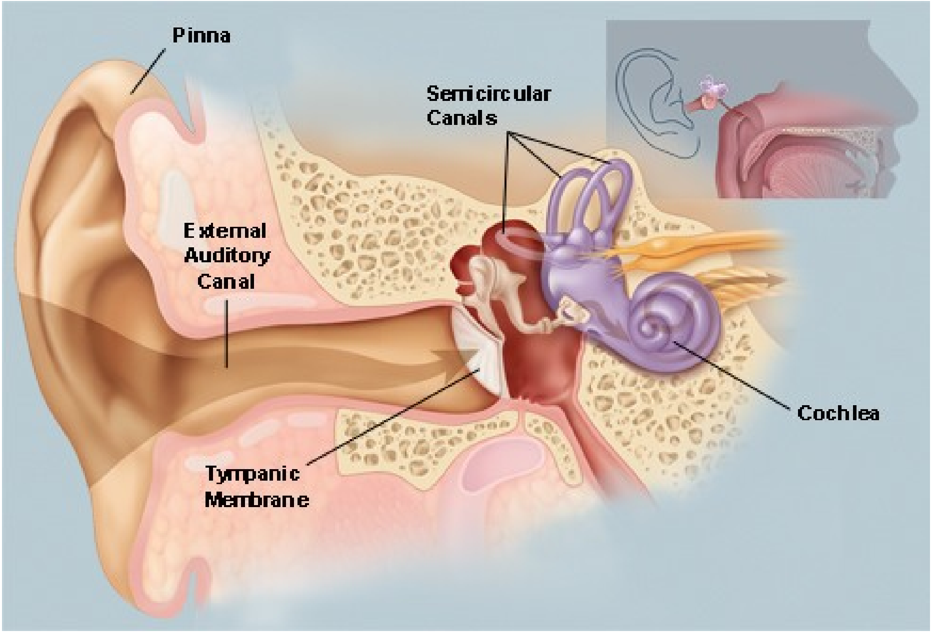
Mel scale

Approximation of **human** perception of speech

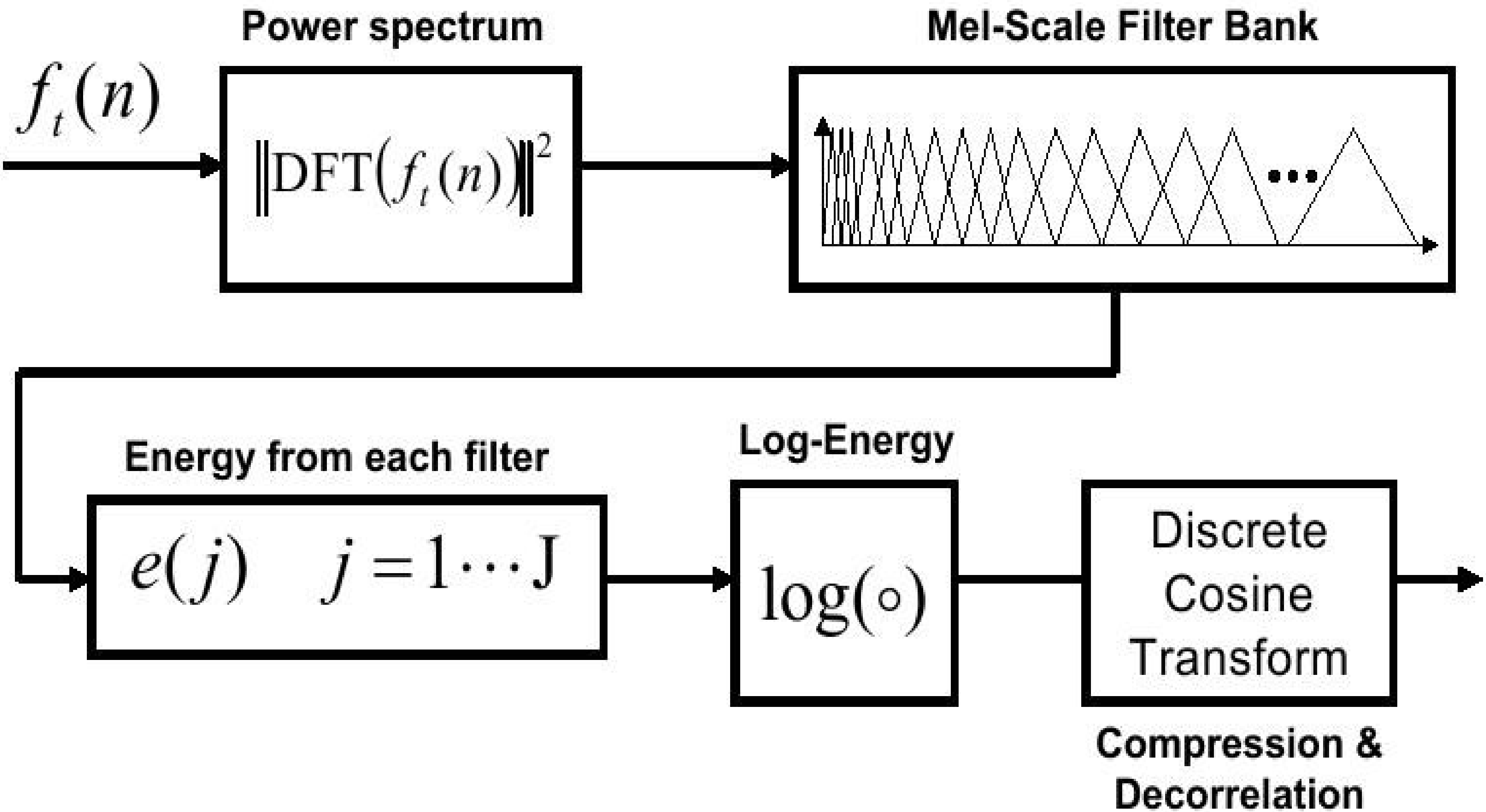
“*Divide the frequency scale into perceptually equal intervals*”:

Linear below 1 kHz, logarithmic above 1 kHz

Mel-Frequency Cepstral Coefficients (MFCC) are commonly used features in ASR

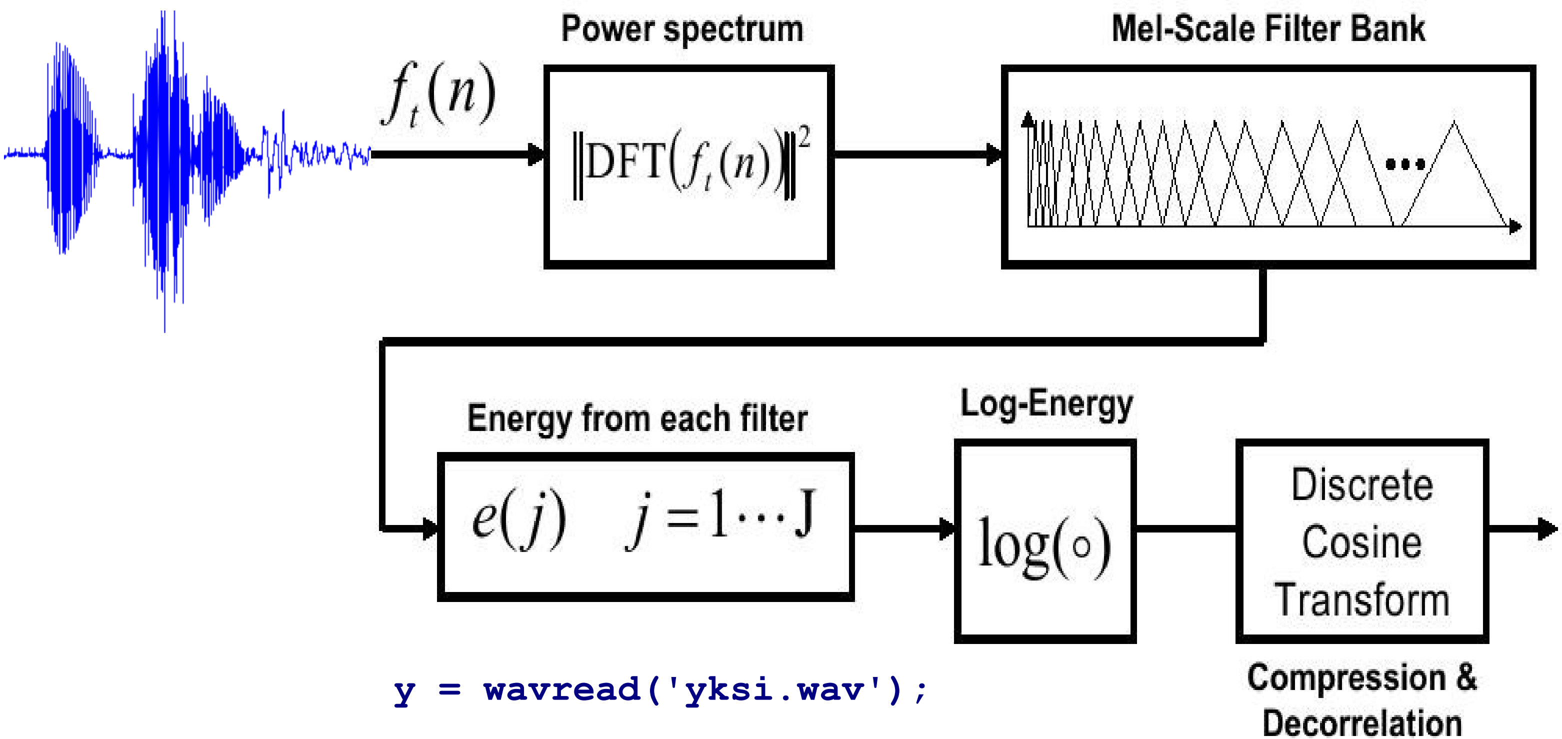


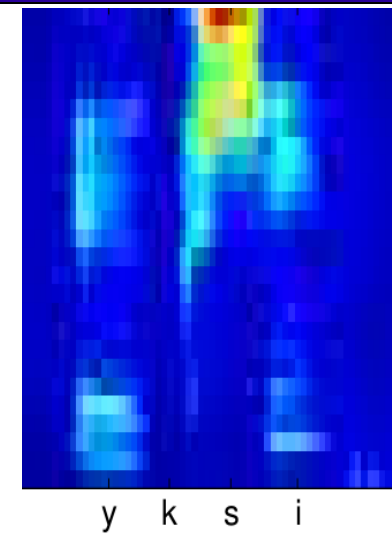
Computation of MFCC



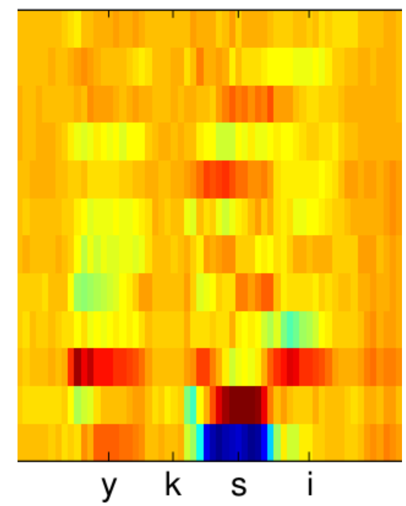
Picture by B.Pellom

In Matlab: computation of MFCC






```
y = wavread('yksi.wav');
s = spectrogram(y,hamming(400),240);
spec = sqrt(abs(s));
mfcc = D*log(1+M*spec);
```

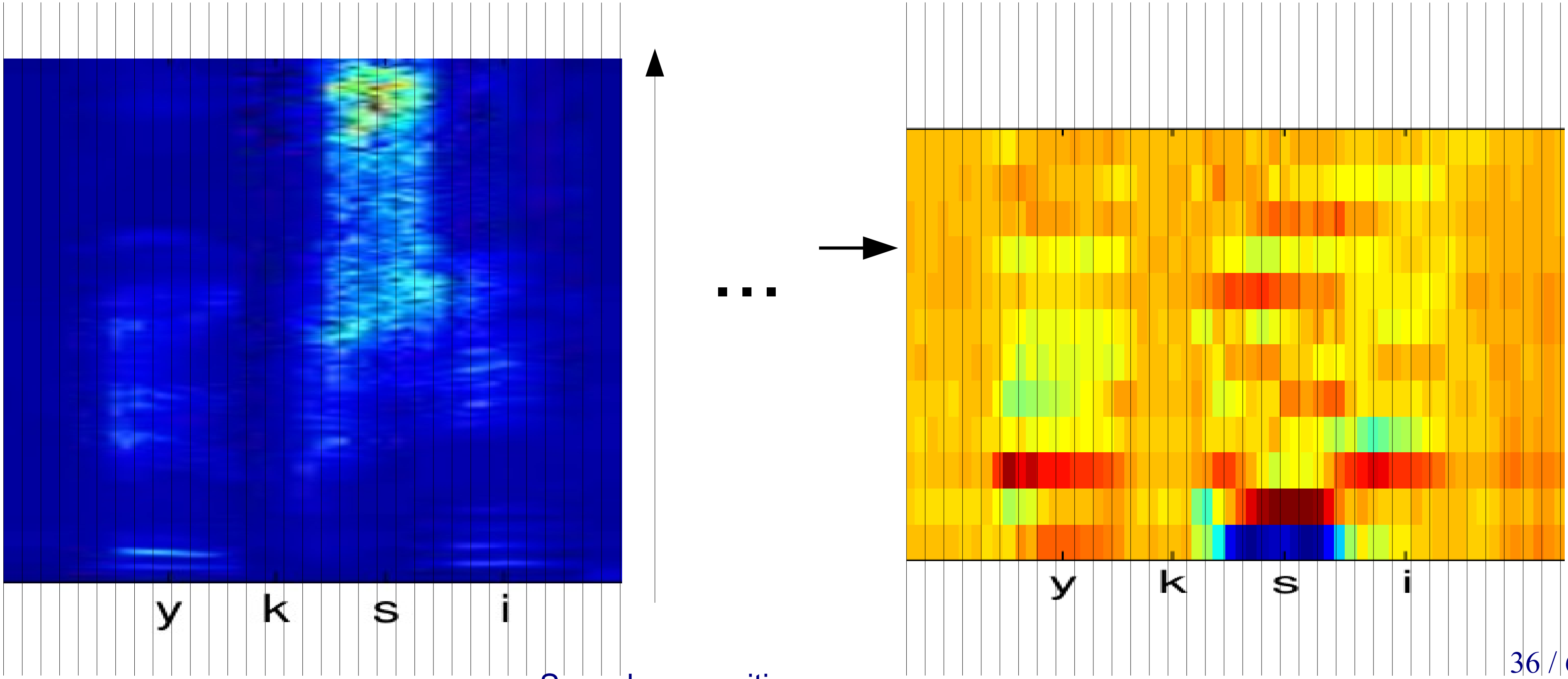


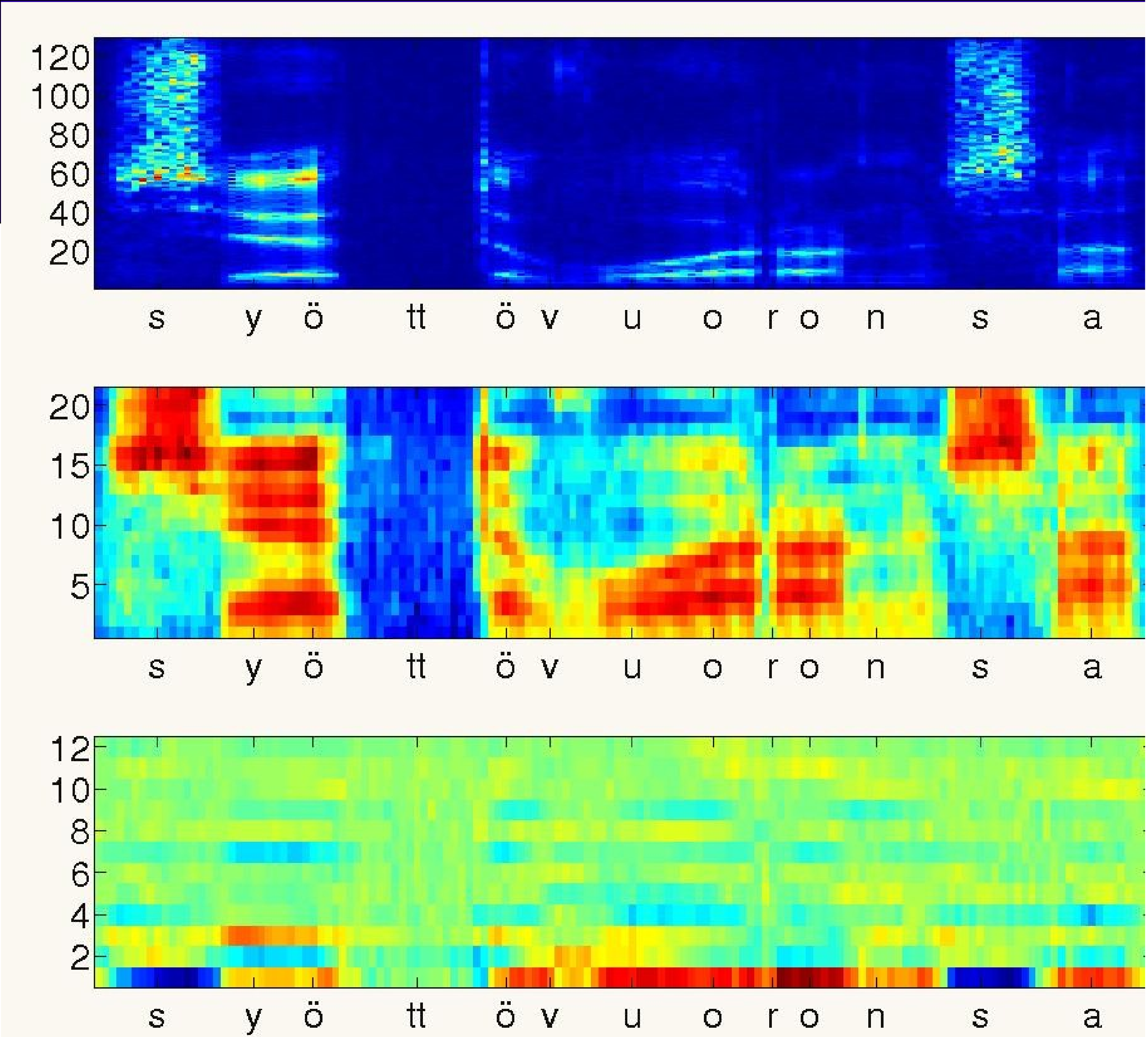
Cepstrum

Cepstrum is essentially “a spectrum of a spectrum”:

- Analysis in frequency scale (vertical direction)

MFCC = Mel-Frequency Cepstral Coefficients



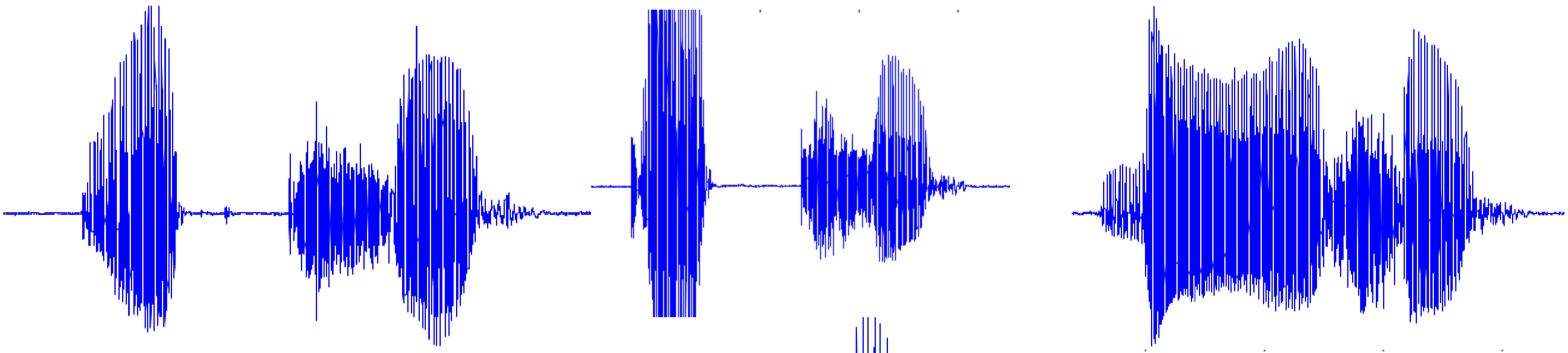


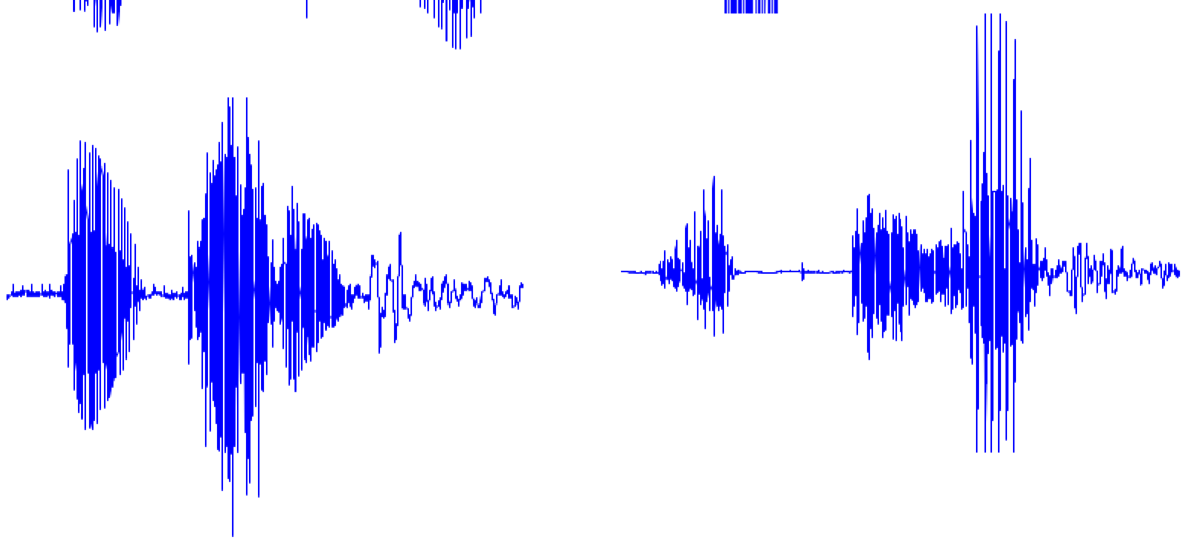
1. Frames:
short 10ms
windows
2. FFT:
power spectrum
spectrogram

3. Filtering:
mel filter
motivated by
human ear
“essential data”

4. Features:
DCT transform
mel cepstrum
MFCC
-less features
-less correlation

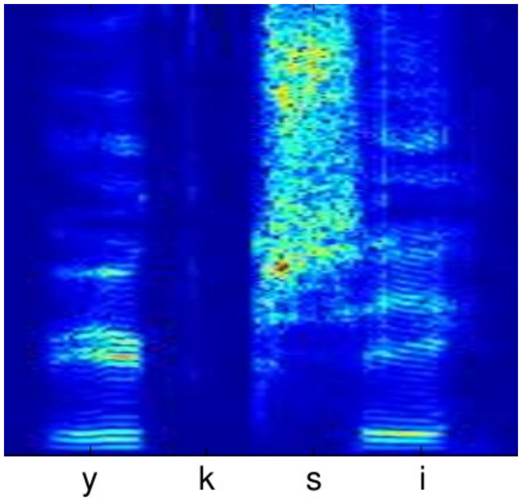
The same 5 samples again

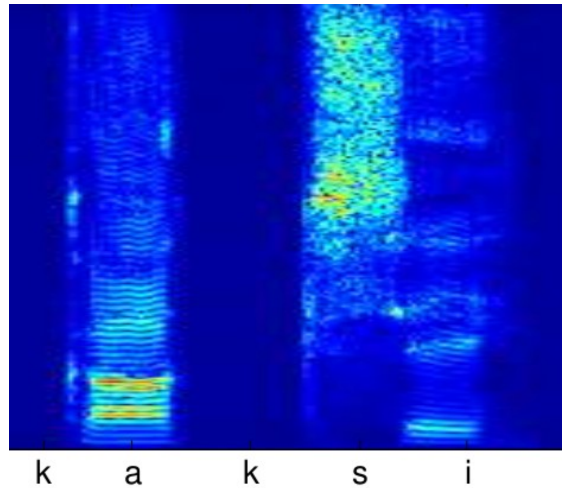


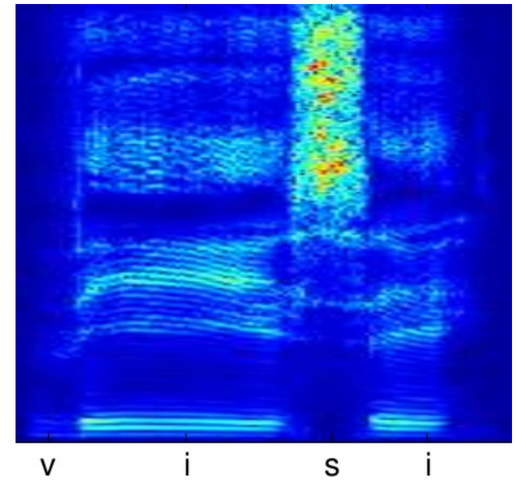


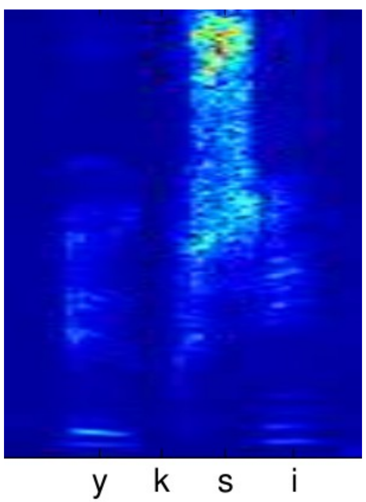
Very difficult to recognize speech from **this picture**...

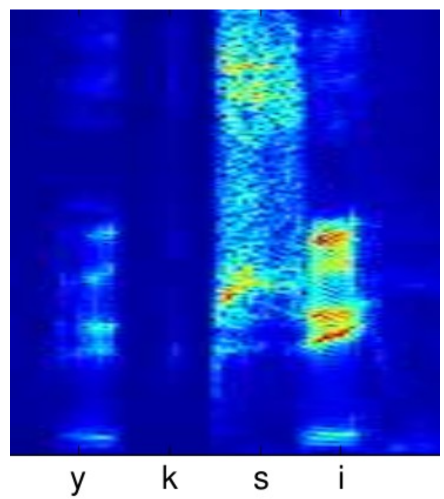
Power spectrogram





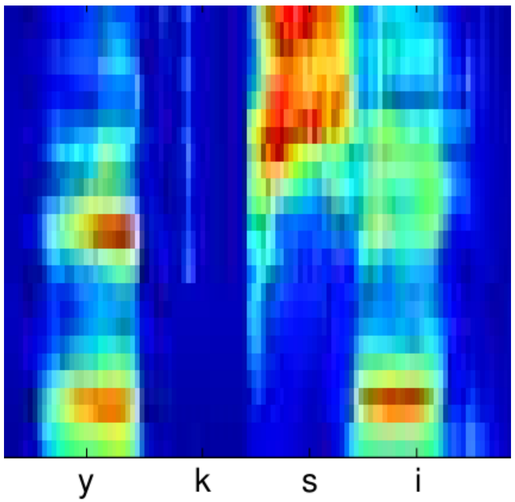


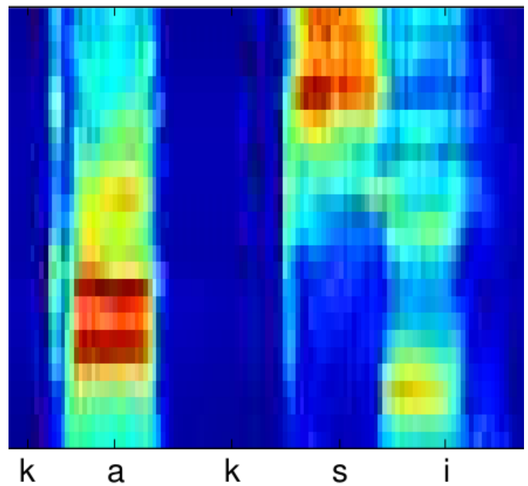


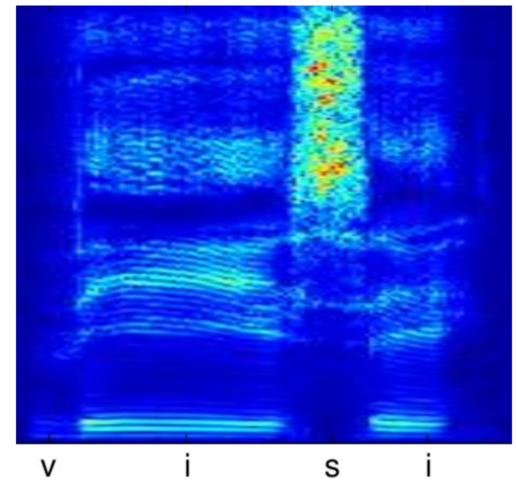


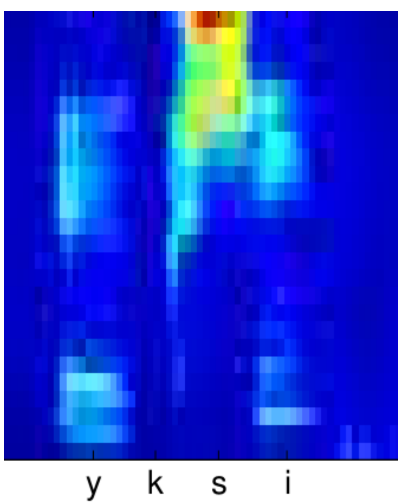
- Speech recognition possible
- Lot of data
- Lot of redundancy
- Lot of noise

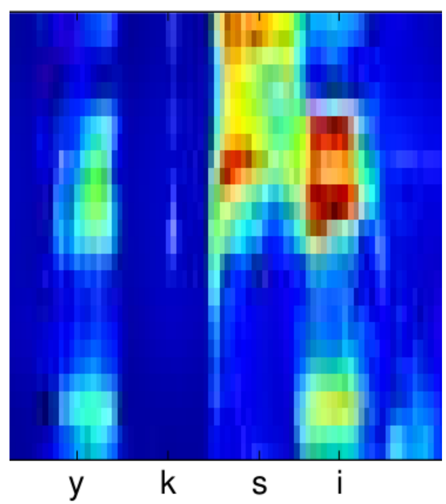
Mel spectrogram





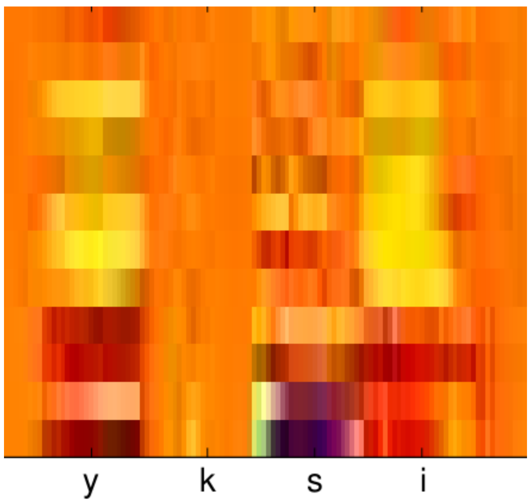


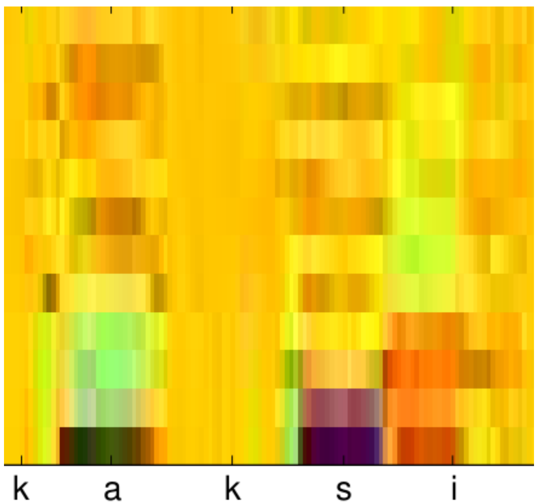


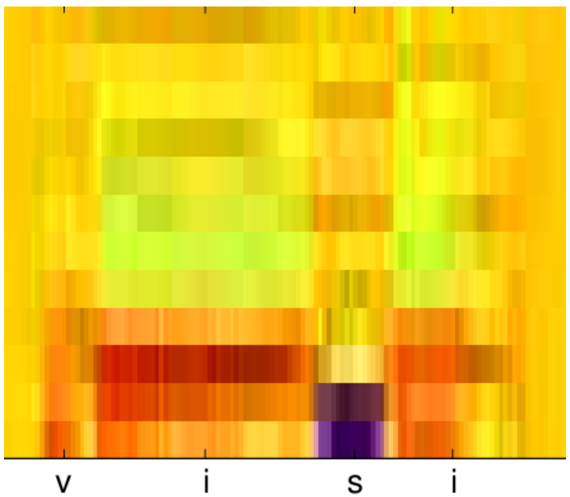


- Speech recognition maybe easier?
- 10 x less data
- Less redundancy
- Less noise

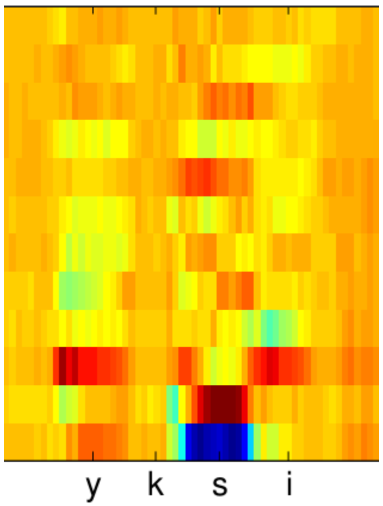
Mel-frequency cepstral coefficients (MFCC)

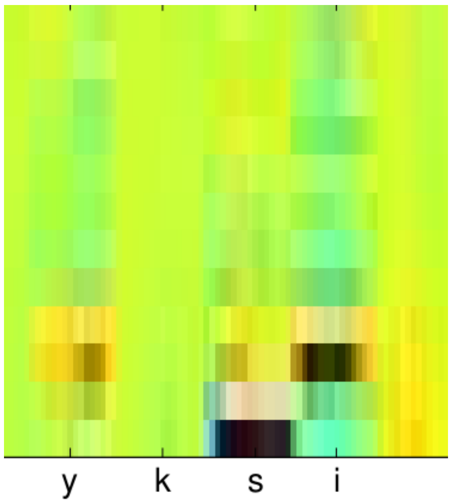




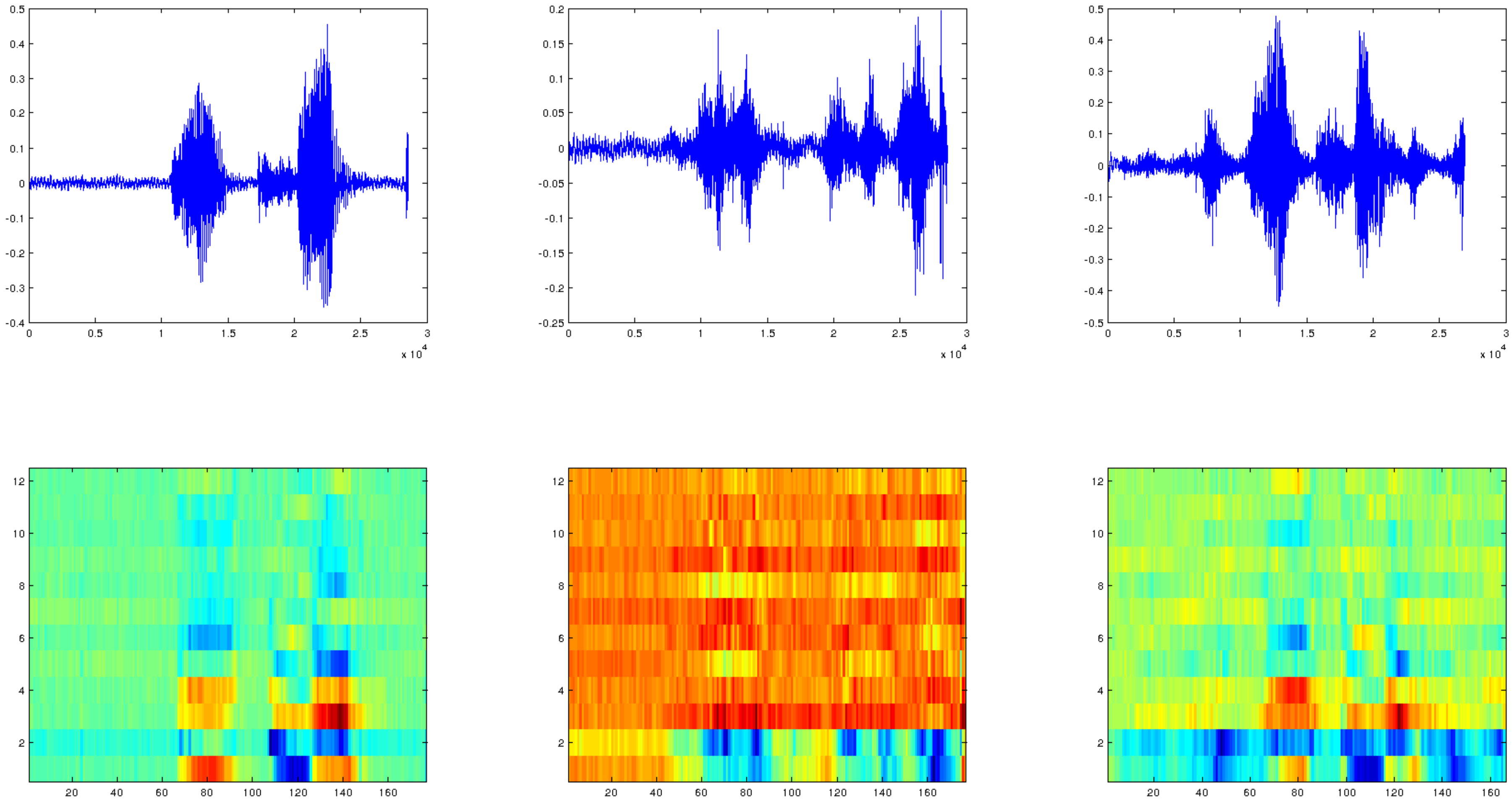


- Even more compact
- Less correlation
- Less noise?





Background noise?



Content today

1. General organization of the course
2. What is automatic speech recognition?
3. Speech as an acoustic signal
4. **GMMs and DNNs**
5. Home exercise 1:
   - Build a system to classify speech features into phonemes
6. Kick-start of the group works

Break 5 min

To classify sounds by features?

Training

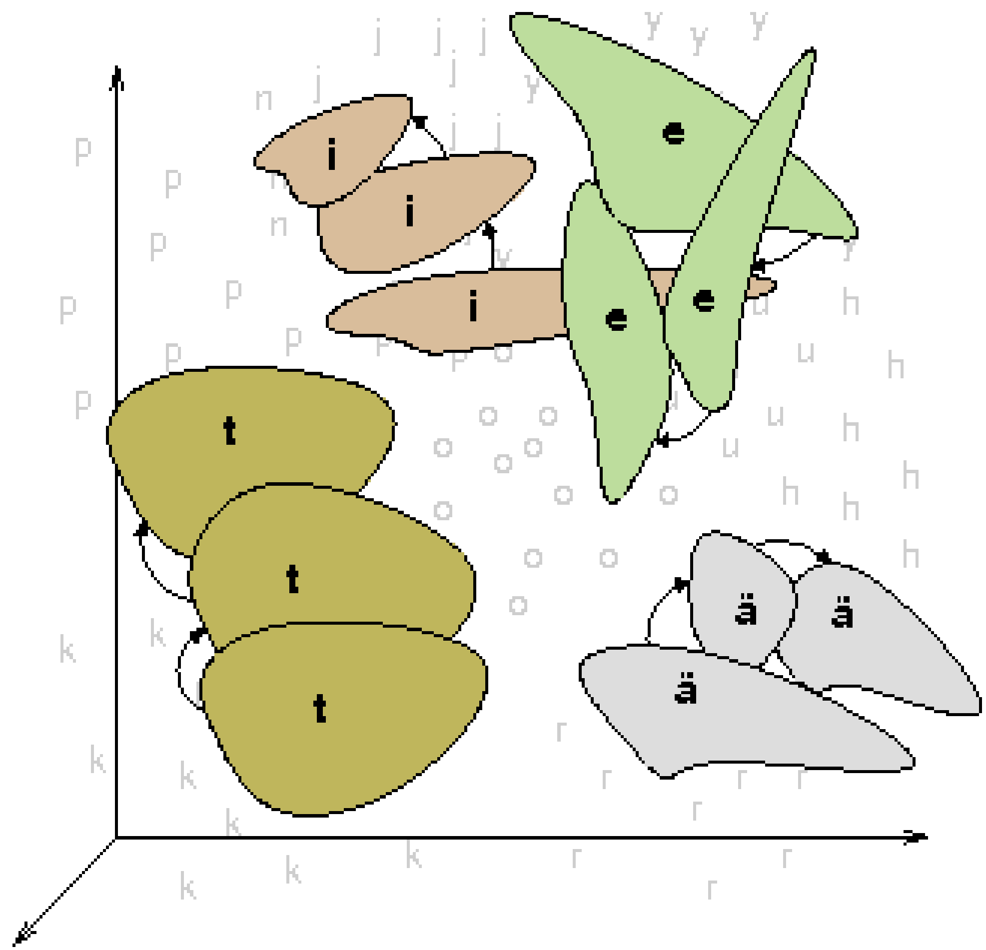
1. Extract MFCC from samples of each sound (e.g. phoneme)
2. Train a statistical model (mean and variance)

Testing

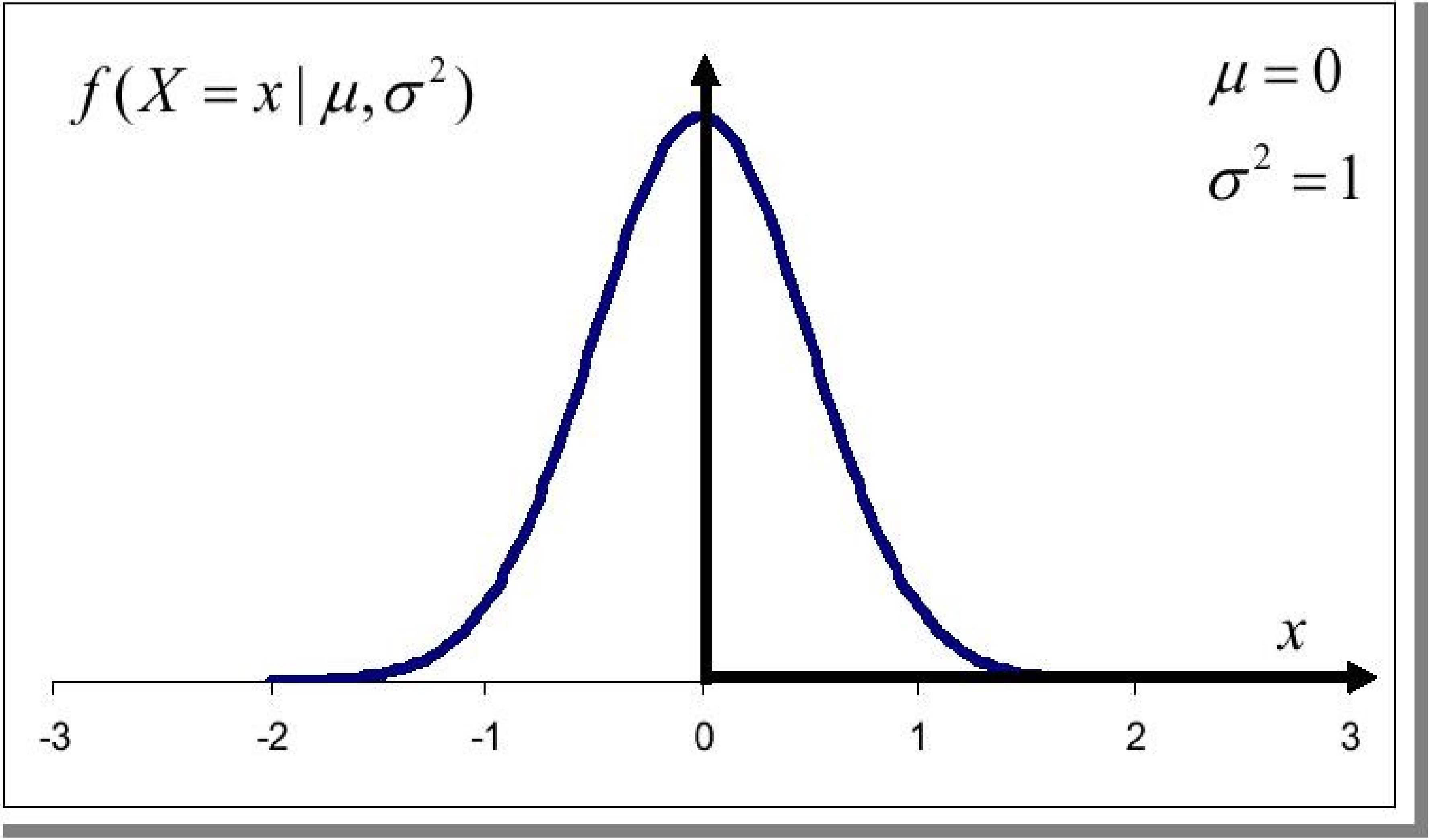
1. Record new samples and extract MFCC
2. Choose the best-matching model to be the class

Classification by features

- Use, for example, a Gaussian mixture model (GMM)
- estimate a set of statistical models (mean and variance parameters) using samples of each sound source
- choose the best-matching statistical model to be the class of an unknown sample



Normal (Gaussian) distribution



Picture by B.Pellom

1dim. Gaussian distribution

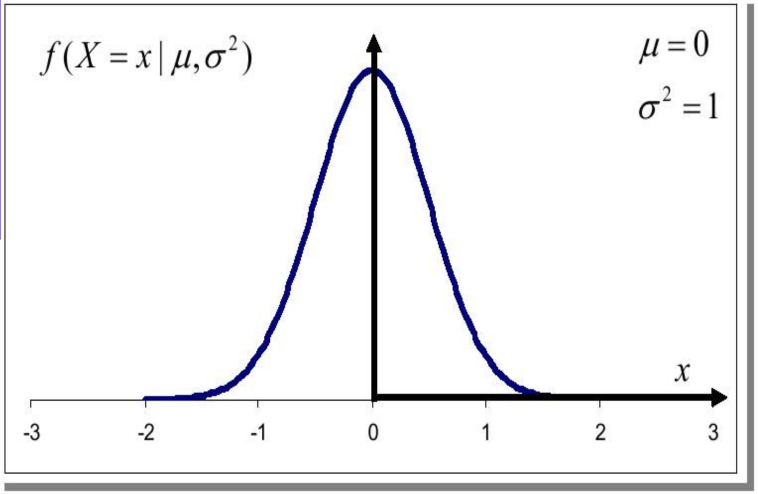
$$f(X = x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2} \right]$$

$$\mu = E[x] = \frac{1}{N} \sum_{n=1}^{N} x_n$$

$$\sigma^2 = E(X^2) - [E(X)]^2 = \frac{1}{N} \sum_{n=1}^{N} x_n^2 - \left[\frac{1}{N} \sum_{n=1}^{N} x_n \right]^2$$

Picture by B.Pellom

GMM example



- 1-dim, 1-mixture, GMM model:

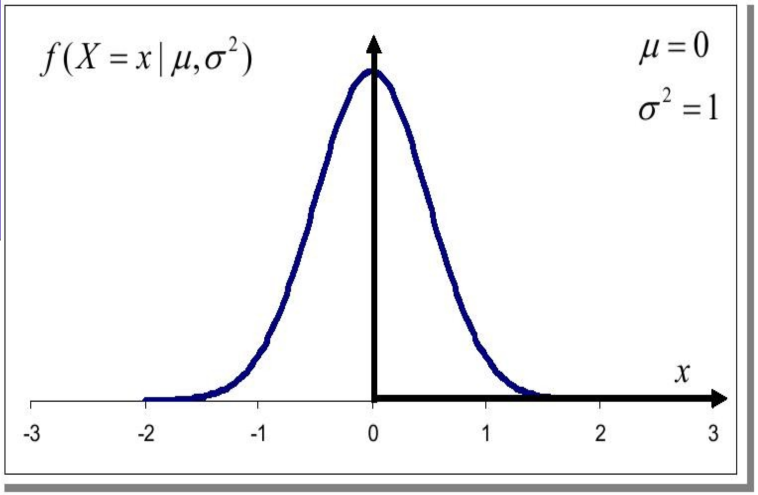
  mean = 100 , variance = 1
- Observed feature **x = 102**, or **x = 99,** then **f(x | 100, 1)** =
- **f(102)** =
- **f(99)** =

Now: Go to **MyCourses > Lectures > Lecture1 exercise** and open the return box
To get an activity point return your solution today. All attempts will be rewarded.

$$f(X=x\,|\,\mu,\sigma^2)=\frac{1}{\sqrt{2\pi\sigma}}\exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

Exp (-2) = 0.14
Exp (-1) = 0.37
Exp (-0.5) = 0.61
1/sqrt(2*pi) = 0.40

GMM example



- 1-dim, 1-mixture, GMM model:
  mean = 100 , variance = 1
- Observed feature **x = 102**, or **x = 99,** then **f(x | 100, 1)** =
- `f(102)=1/(2*pi)**(1/2) *exp(-1/2*(102-100)**2)`

  `= 0.40 *exp (-0.5*4) = 0.40 * 0.14 = 0.054`
- `f(99) =1/(2*pi)**(1/2) *exp(-1/2*(99-100)**2)`

  `= 0.40 *exp (-0.5*1) = 0.40 * 0.61 = 0.24`

$$f(X=x\mid\mu,\sigma^2)=\frac{1}{\sqrt{2\pi\sigma}}\exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

Exp (-2) = 0.14
Exp (-1) = 0.37
Exp (-0.5) = 0.61
1/sqrt(2*pi) = 0.40

Multidim. Gaussian distribution

$$f(X = x \mid \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left[-\frac{1}{2} (x-\mu)^{\mathrm{T}} \Sigma^{-1} (x-u) \right]$$

Determinant of covariance matrix

Distribution covariance matrix

Distribution mean vector

Observed vector of random variables (features)

Picture by B.Pellom

Diagonal Gaussian

- **Most speech recognition systems assume diagonal covariance matrices**

- **Data sparseness issue:**

$$\Sigma = \begin{bmatrix} \sigma_{11}^2 & 0 & 0 & 0 \\ 0 & \sigma_{22}^2 & 0 & 0 \\ 0 & 0 & \sigma_{33}^2 & 0 \\ 0 & 0 & 0 & \sigma_{44}^2 \end{bmatrix} \longrightarrow \boxed{|\Sigma| = \prod_{n=1}^{d} \sigma_{nn}^2}$$

Picture by B.Pellom

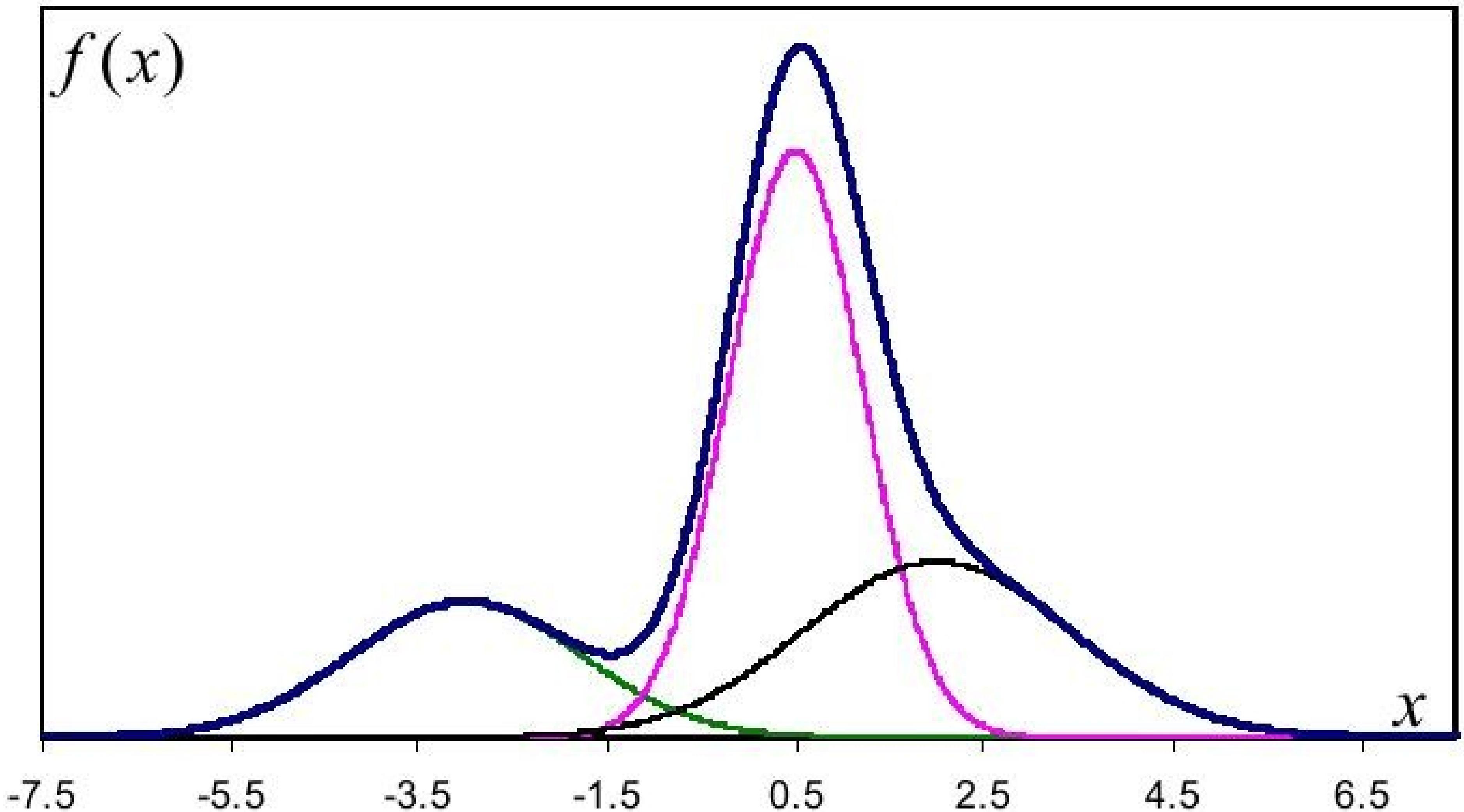
Inverse of the covariance matrix

- **Inverting a diagonal matrix involves simply inverting the elements along the diagonal:**

$$\Sigma^{-1} = \begin{bmatrix} \frac{1}{\sigma_{11}^2} & 0 & 0 & 0 \\ 0 & \frac{1}{\sigma_{22}^2} & 0 & 0 \\ 0 & 0 & \frac{1}{\sigma_{33}^2} & 0 \\ 0 & 0 & 0 & \frac{1}{\sigma_{44}^2} \end{bmatrix}$$

Picture by B.Pellom

1dim. Gaussian mixture model



Picture by B.Pellom

Gaussian mixture model GMM

- **Distribution is governed by several Gaussian density functions,**
- **Sum of Gaussians (w_m = mixture weight)**

$$f(x) = \sum_{m=1}^{M} w_m \mathrm{N}_m\left(x; \mu_m, \Sigma_m\right)$$

$$= \sum_{m=1}^{M} \frac{w_m}{(2\pi)^{n/2} \left|\Sigma_m\right|^{1/2}} \exp\left[-\frac{1}{2}\left(x-\mu_m\right)^{\mathrm{T}} \Sigma_m^{-1}\left(x-u_m\right)\right]$$

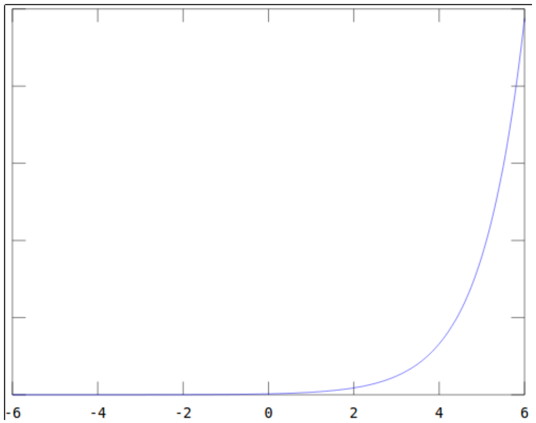
Picture by B.Pellom

Other classifiers

- Probability density functions (such as **GMM**) that model the distribution of the data
- Methods such as **K-nearest neighbors** that directly use the data
- Methods such as **K-means** that learn the clusters in the data
- Discriminative models that directly learn to optimize the classification accuracy
  - Linear: Support Vector Machine (**SVM**)
  - Non-linear: Multilayer Perceptron and other **Deep Neural Networks** (**DNN**)

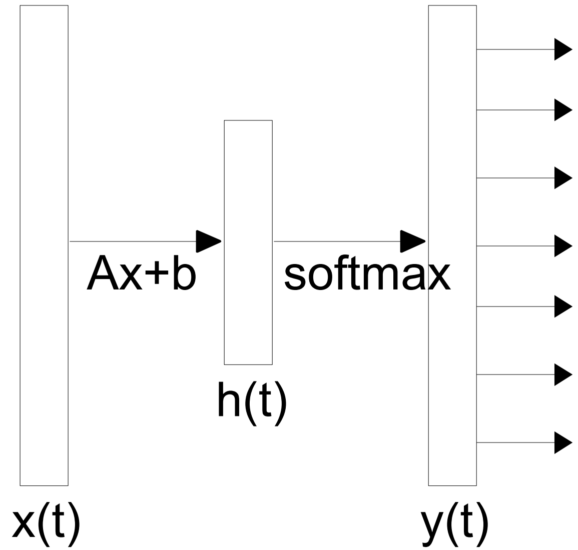
A simple 1-layer NN

- Outputs the **probability of classes** y(t) given the observation x(t)
- **Input layer** is the feature vector x(t) of the current frame
- **Hidden layer** has a linear transform h(t) = Ax(t) + b to compute a representation of **linear distributional features** or factors
- **Output layer** maps the values by y(t) = softmax (h(t)) to range (0,1) that add up to 1
- Resembles a simple linear classifier



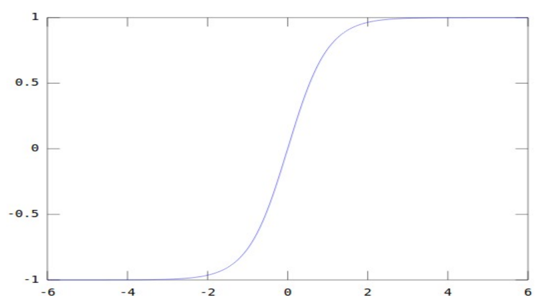
Softmax:

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

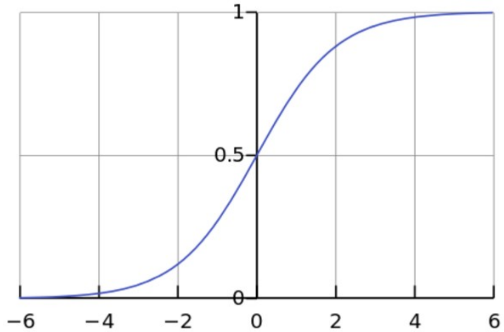


A non-linear 1-layer NN

- The only difference to the simple NN is that the hidden layer h(t) now includes a non-linear function h(t) = U(Ax(t) + b)
- Can learn more complex feature representations
- Common examples of non-linear functions U:

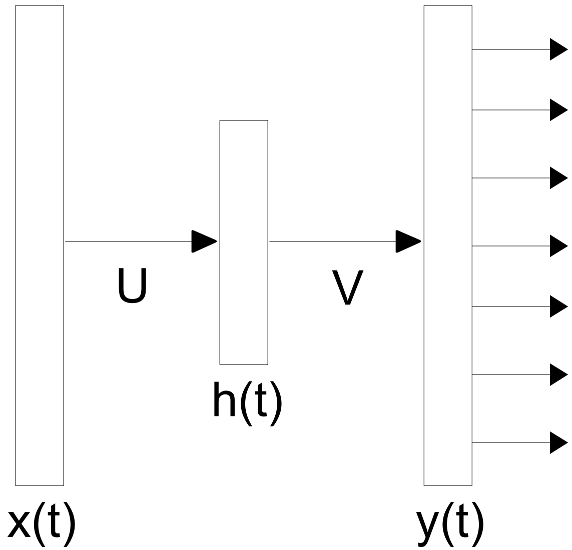


U (t) = tanh (t)



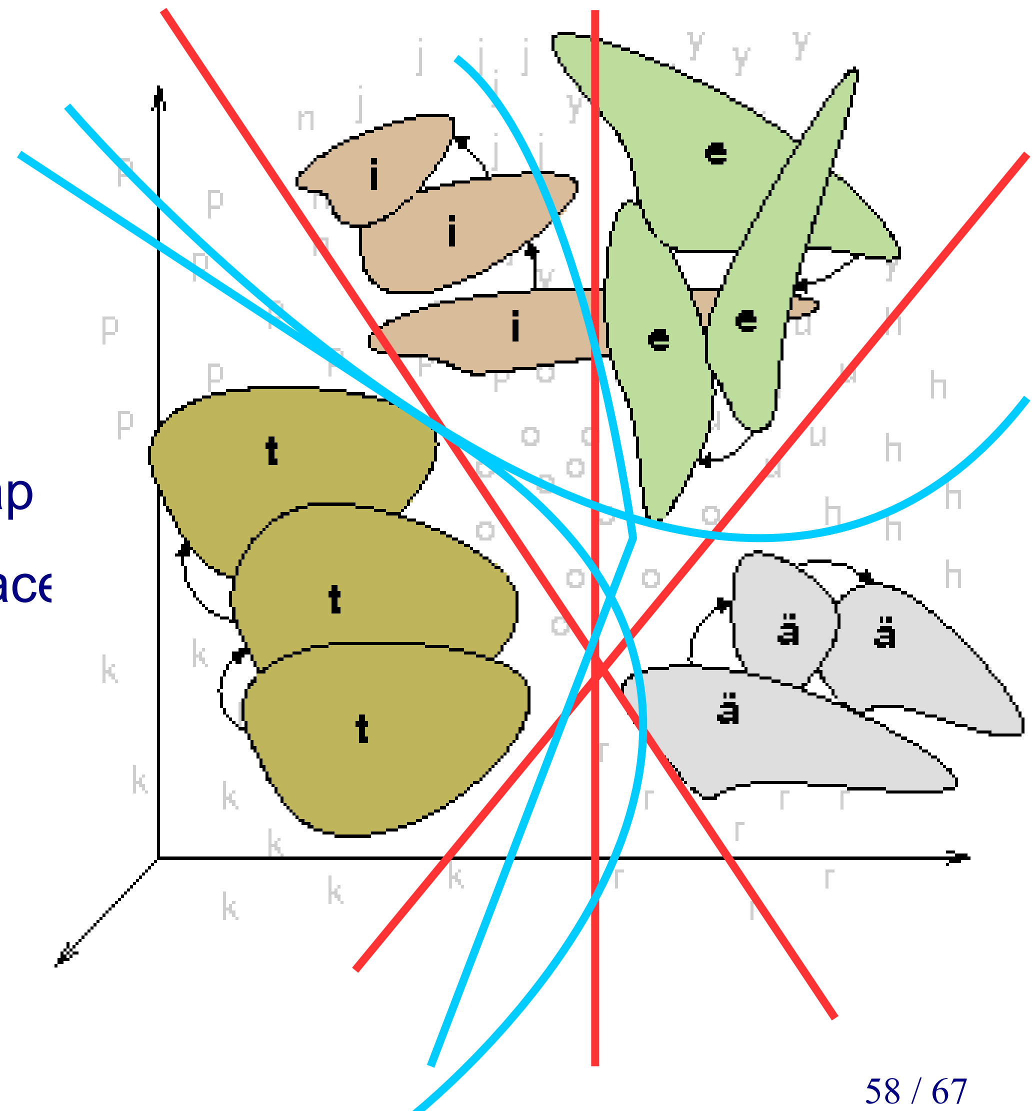
Sigmoid

$$U(t) = \frac{1}{1+e^{-t}}$$



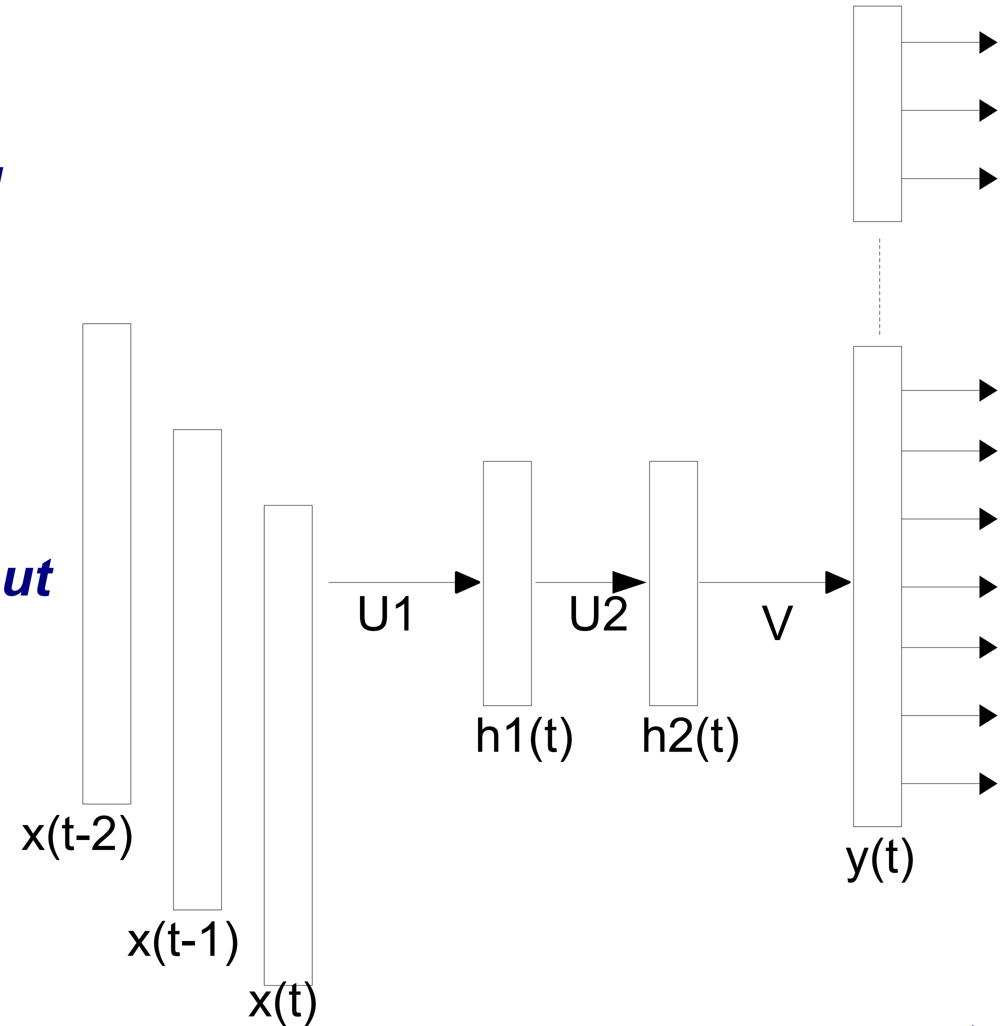
Linear and non-linear classifiers

- Find a **linear** transformation **h = Ax + b** to map the input coordinates to a new space where the classes are easier to separate
- Find a more complex **non-linear** transformation **h = U(Ax + b)** to map the input coordinates into a new space for classification



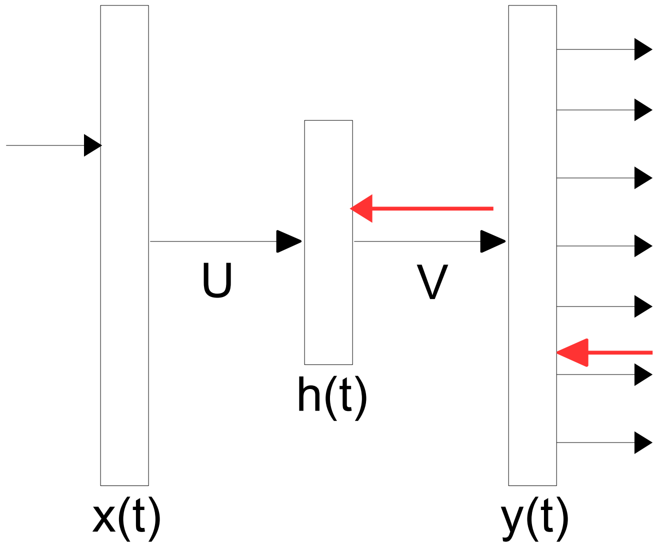
Common NN extensions

- ***Input layer*** is expanded *over several previous frames* x(t-1), x(t-2), .. to learn richer representations
- *Deep neural networks* have *several* ***hidden layers*** h1, h2, .. to learn to represent information at several hierarchical levels
- Can compute probabilities for thousands of context-dependent speech units by extending the ***output layer*** *y(t)*



NN training

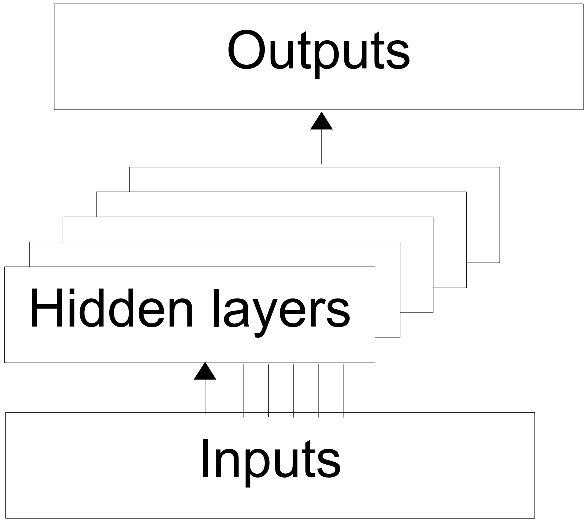
- Supervised training minimizes the **output errors** by training the weights for V by *stochastic gradient descend*
  - Tunes the weights to the direction of giving 1 to correct class and 0 to others
- Propagate the output **error to hidden layer** to train the weights for U
  - Tunes the weights based on how much they contributed to the output
- In practice, deep NNs will require more complex training procedures, since the gradients *vanish* quickly
  - After some propagation steps the individual contributions to the output become roughly equal



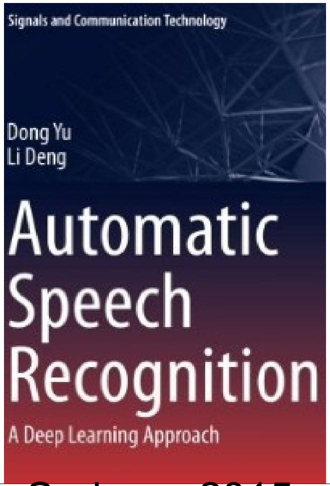
Analysis of DNNs in acoustic models

3 key improvements:

1. Processing in many hierarchical layers
2. Input from many frames
3. Output for context-dependent phones



Other significant improvements: speedups, pre-training, sequence discriminative training, multitask learning, various NN architectures (CNN, RNN, LSTM, Highways)



D.Yu, L.Deng. Automatic Speech Recognition A Deep Learning Approach. Springer 2015.

Home exercise 1

- Build a **classifier** to classify speech features into phonemes !
- Details, instructions and help given in Thursday/Friday meeting this week
- To be returned by Wednesday **next week**

Feedback

Now: Go to **MyCourses > Lectures > Lecture 1 feedback** and fill in your feedback. To get an activity point submit the form today.

- Write down questions from the lecture that troubled your mind
- Comments and suggestions are welcome, too.
- What was missing today, and what too much?

Idea's taken from last years' feedback:

- Tutors to join the project meetings
- Pre-assign groups, topics, and the first meeting date
- Lecture recordings available
- Add a simple GMM example

Next meeting

- Thu 10.15 – 12 or Fri 14.15 – 16
- Speech data and support provided for practical experiments
- Organized by Aku Rouhe
- Get your AALTO account ready!
- Python and Jupyter Notebooks used, links to guides available on request
  - There is an old substitute in Matlab if someone prefers
- Support for Home exercise 1 provided only in the computer sessions of this week (!)

Content today

1. General organization of the course
2. What is automatic speech recognition?
3. Speech as an acoustic signal
4. GMMs and DNNs
5. Home exercise 1:
   - Build a system to classify speech features into phonemes
6. **Kick-start of the group works**

Project work receipt

1. Form a group (3 persons)
2. Get a topic
3. Get reading material from *Mycourses* or your group tutor
4. 1^{st} meeting: Specify the topic, start literature study (DL Nov 10)
5. 2^{nd} meeting: Write a work plan (DL Nov 17)
6. 3^{rd} - 5^{th} meetings: Perform analysis, experiments, and write a report
7. Book your presentation time for weeks 6 - 7 (DL Dec 3)
8. Prepare and keep your 20 min presentation
9. Return the report (DL Dec 17)

Done already?

This week

Project topics & tutors

1. Fine-tuning wav2vec2. Katja
2. Curriculum learning for ASR. Georgios
3. Language Identification. Dejan
4. Spoken language understanding. Aku
5. End-to-end ASR for Timit. Anssi
6. Speaker identification. Anssi
7. Automatic detection of alcohol intoxication. Dejan
8. End-to-end Speech Translation. Ragheb
9. Restore Capitalization and Punctuation in ASR output. Tamas
10. Speech adaptation for children speech recognition: Hemant
11. Audio event tagging. Anja
12. Language model adaptation. Katja
13. Speaker Adaptation Hemant
14. Paralinguistic systems. Tamas
15. Speech emotion recognition. Juho
16. Spoken command recognition. Anja
17. Finite state transducers n ASR. Aku
18. Spoken language modeling. Ragheb
19. Speech/text synthesis. Juho

Check **MyCourses > Projects** to see your group, topic and tutor