2. Cepstrum alanysis of speech signals
3. Vector space representation of words

ELEC-E5521 Speech and language processing methods

Spring 2022

Lecture: *Mikko Kurimo*

Exercises: Juho Leinonen

Contents

<u>Cepstrum</u>

- Literature and other material

Idea and history of cepstrum

Cepstrum and LP model

Mel cepstrum

Pitch detection, formant tracking

Phoneme recognition

Temporal (a.k.a. delta) features

Exercises

<u>Word2vec</u>

- meaning of words
- statistical semantics
- word-document matrix
- word-word matrix
- distributed semantics
- Exercises

Reading material

1. Cepstrum chapter in **John R. Deller, John G. Proakis, and John H. L. Hansen**: Discrete-Time Processing of Speech Signals
2. Homomorphic Speech Analysis chapter (5) in **L. R. Rabiner and R. W. Schafer**: Introduction to Digital Speech Processing (2007).

http://www.ece.ucsb.edu/Faculty/Rabiner/ece259/speech%20course.html

3. Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXivpreprint arXiv:1301.3781 (2013). https://arxiv.org/pdf/1301.3781.pdf
4. Mikolov, Tomas, et al. Distributed representations of words and phrases and their compositionality.Advances in neural information processing systems. 2013. https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

5. Baroni, Marco, Georgiana Dinu, and GermánKruszewski. "Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors." ACL (1). 2014. http://anthology.aclweb.org/P/P14/P14-1023.pdf

Slides

1. Today's lecture
2. Homomorphic Speech Analysis, lecture (12) in **L. R. Rabiner**'s Digital Speech Processing Course (2015)

http://www.ece.ucsb.edu/Faculty/Rabiner/ece259/speech%20course.html

3. Word2Vec, *ELEC-E5550 - Statistical Natural Language Processing,* lecture (3) **Tiina Lindh-Knuutila (**2020)
4. Distributional approaches to word meanings. **Chris Potts**, Stanford course. Ling 236/Psych 236c: Representations of meaning, Spring 2013. https://web.stanford.edu/class/linguist236/materials/ling236-handout-05-09-vsm.pdf

Introduction

In linear systems the useful information can easily be separated from additive noise by filtering, if we know in which frequency range each occur. For example:

- $x[\mathrm{n}] = x_1[\mathrm{n}]+w[\mathrm{n}]$, where n is index of time
- $x_1[\mathrm{n}]$ is the useful signal and $w\,[\mathrm{n}]$ high frequency noise
- lin. operator $\blacksquare\,[.]$ is a low-pass filter

$$\blacksquare\,[x[\mathrm{n}]] = \blacksquare\,[x_1[\mathrm{n}]+w[\mathrm{n}]] = \blacksquare\,[x_1[\mathrm{n}]]+\blacksquare\,[w[\mathrm{n}]] \approx x_1[\mathrm{n}]$$

But this is much harder, if the signal and noise are **convoluted** (*). For example the source-filter model of speech production:

- $s[\mathrm{n}] = e[\mathrm{n}] * h[\mathrm{n}]$
- $e[\mathrm{n}]$ is the flowing air (source) and $h[\mathrm{n}]$ vocal tract (filter)

$\mathbf{I}\ [s[\mathrm{n}]] = \mathbf{I}\ [e[\mathrm{n}] * h[\mathrm{n}]]$ will not help, so

=> We need a new operator that could *separate* convoluted components!

$H\ [s[\mathrm{n}]] = H\ [e[\mathrm{n}] * h[\mathrm{n}]] = H\ [e[\mathrm{n}]] + H\ [h[\mathrm{n}]]$

The complex *cepstrum* operator transforms *convolution* into *addition*.

- Cepstrum was developed to *separate convoluted signals*: $e[n]*h[n]$
- Fourier: $\mathcal{F}[e*h] = E[k]\,H[k]$, where k is index of frequency
- $\mathrm{Log}[\,E\,H\,] = \mathrm{Log}[\,E\,] + \mathrm{Log}[\,H\,]$
- Linear combination may be separated by linear band-pass ”filtering” (called *liftering* in cepstral domain)

History

- **Bogert, Healy, and Tukey**, ”The quefrency analysis of time series for echoes: Cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking” In M. Rosenblatt, ed., *Proceedings of the Symposium on Time Series Analysisı.* J. Wiley & Sons, pp. 209-243, NY, 1963.
- Tukey = ”The FFT man”
- spectrum <-> cepstrum
- "quefrency," "gamnitude," ”lifter”, ”alanysis”, “saphe”

- **Noll A. M.,** ”Cepstrum pitch determination”, *JASA (Journal of Acoustical Society of America)* vol. 41, pp. 293-309, Feb. 1967.

- Homomorphic signal processing
  - Oppenheim (1967, 1969)
  - Shafer (1968)
  - Homomorphic ≈ ”*same shape*”
  - ”+” <-> ”*” ; ”linear domain” <-> ”convolution domain”

Homomorphic System

$$\mathcal{H}[s[n]] = \mathcal{H}[e[n] * h[n]] = \mathcal{H}[e[n]] + \mathcal{H}[h[n]]$$

Typically, used to separate ”noise” i.e. impulse $e[n]$ from system response $h[n]$ using operator $\mathcal{H}$, hoping that:

$\mathcal{H}[e[n]] \approx \delta[n]$ ja $\mathcal{H}[h[n]] \approx h[n]$.

Cepstrum operator is not an ideal separator, but can *approximate* a **homomorphic system**.

How to recognize speech sounds?

A simple procedure:

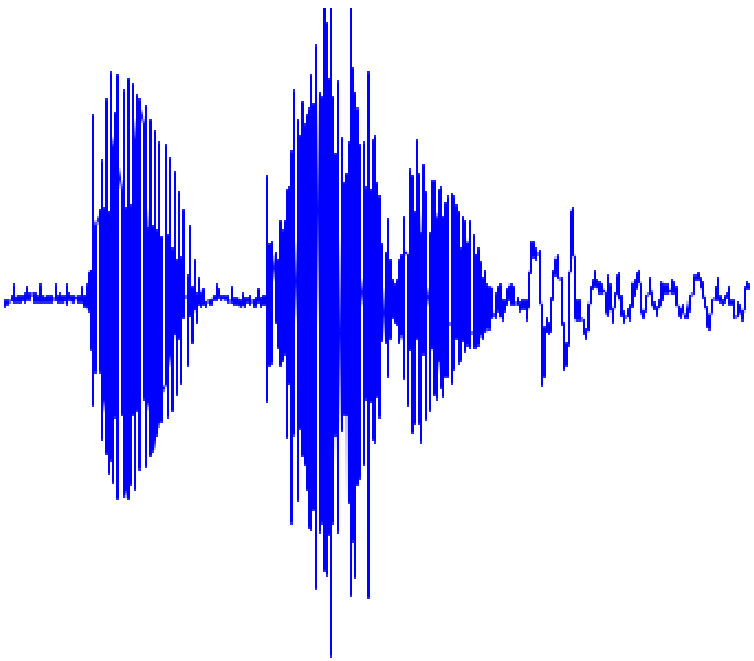
Measure some **characteristic features** of the signal and train statistical models for them

Good features should be:

1. Compact
2. Discriminative for speech sounds
3. Fast to compute
4. Robust for noise

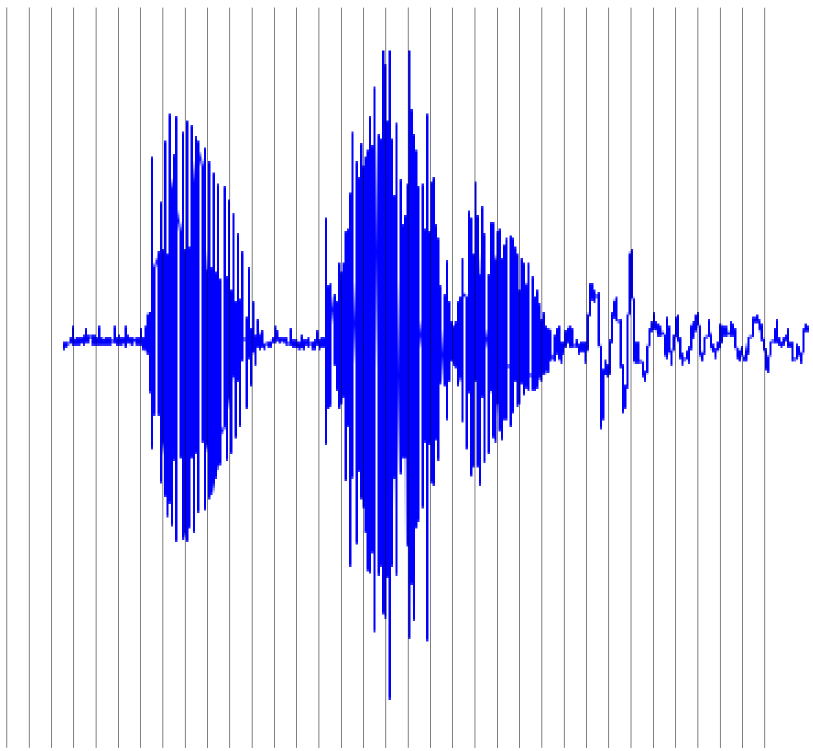
Frequency analysis

Calculate the short-time spectrum in short intervals



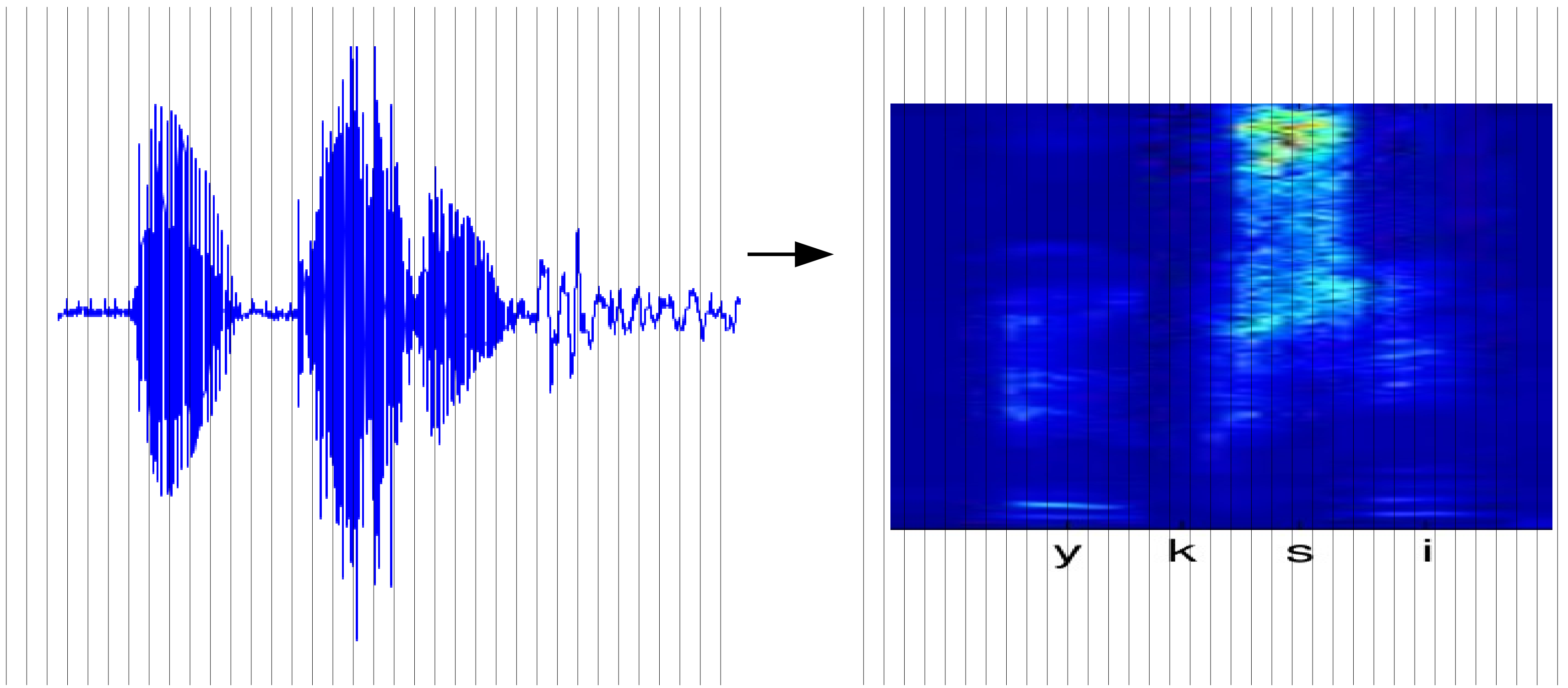
Frequency analysis

Calculate the short-time spectrum in short intervals



Frequency analysis

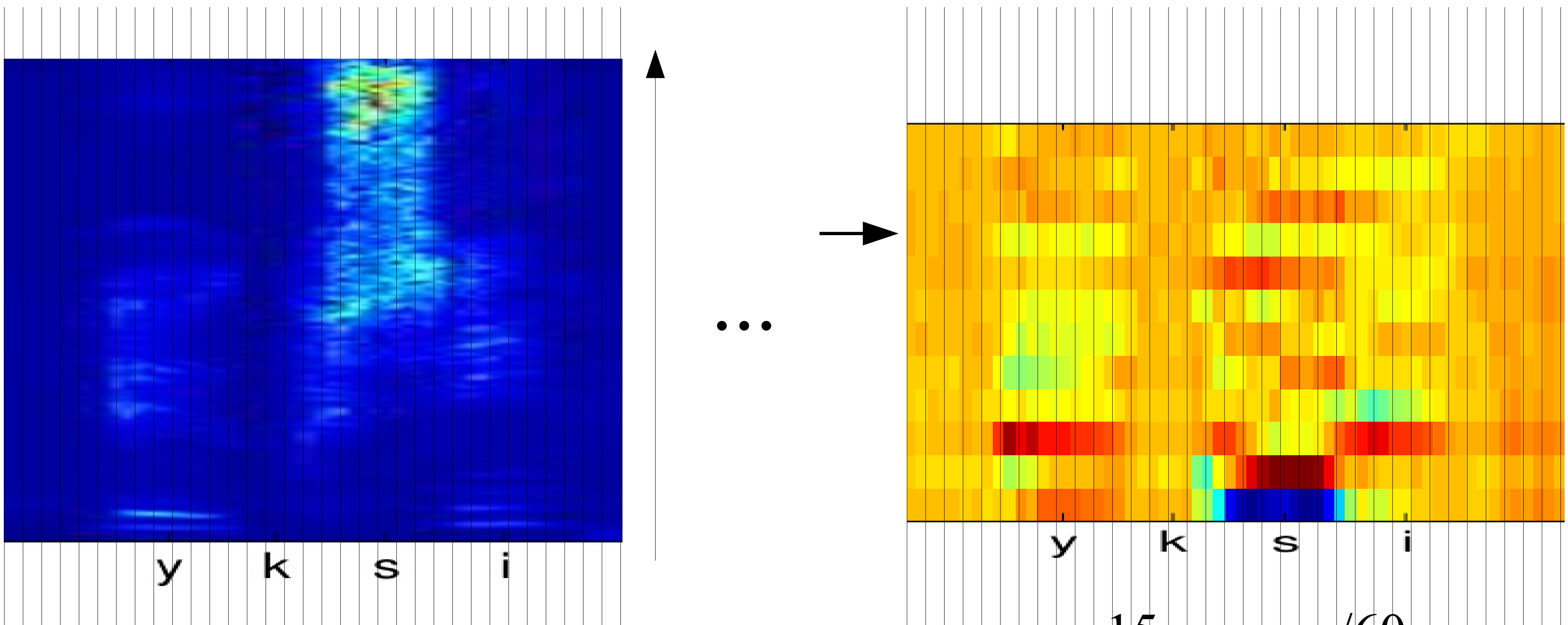
Calculate the short-time spectrum in short intervals



Cepstrum

Short-time analysis in frequency scale (vertical direction)

MFCC = Mel-Frequency Cepstral Coefficients

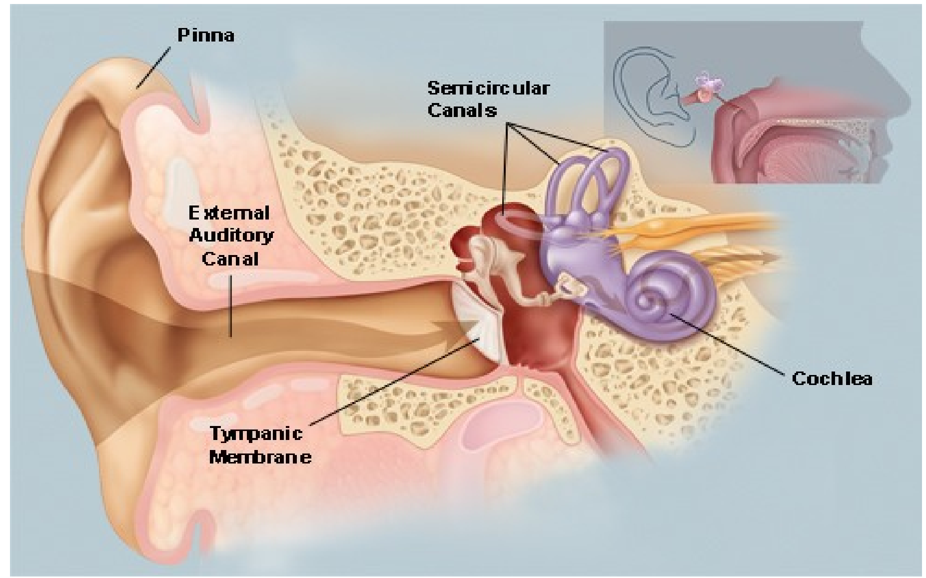


Mel scale

Approximation of **human** perception of speech

"*Divide the frequency scale into perceptually equal intervals*":

Linear below 1 kHz, logarithmic above 1 kHz



Mel-Cepstrum

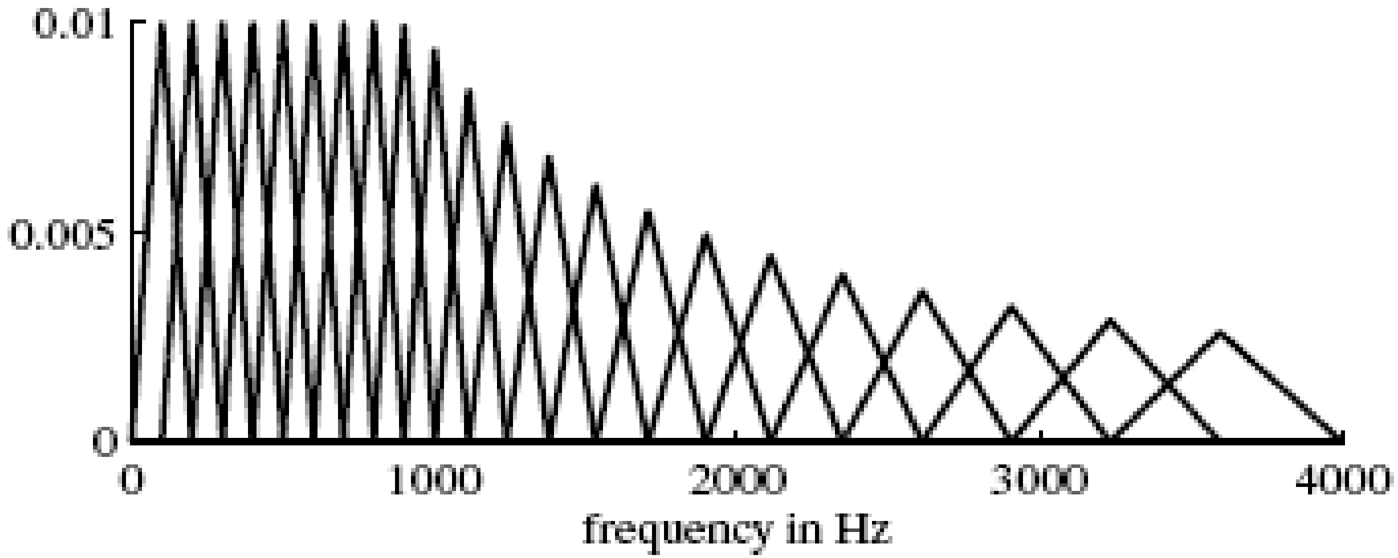
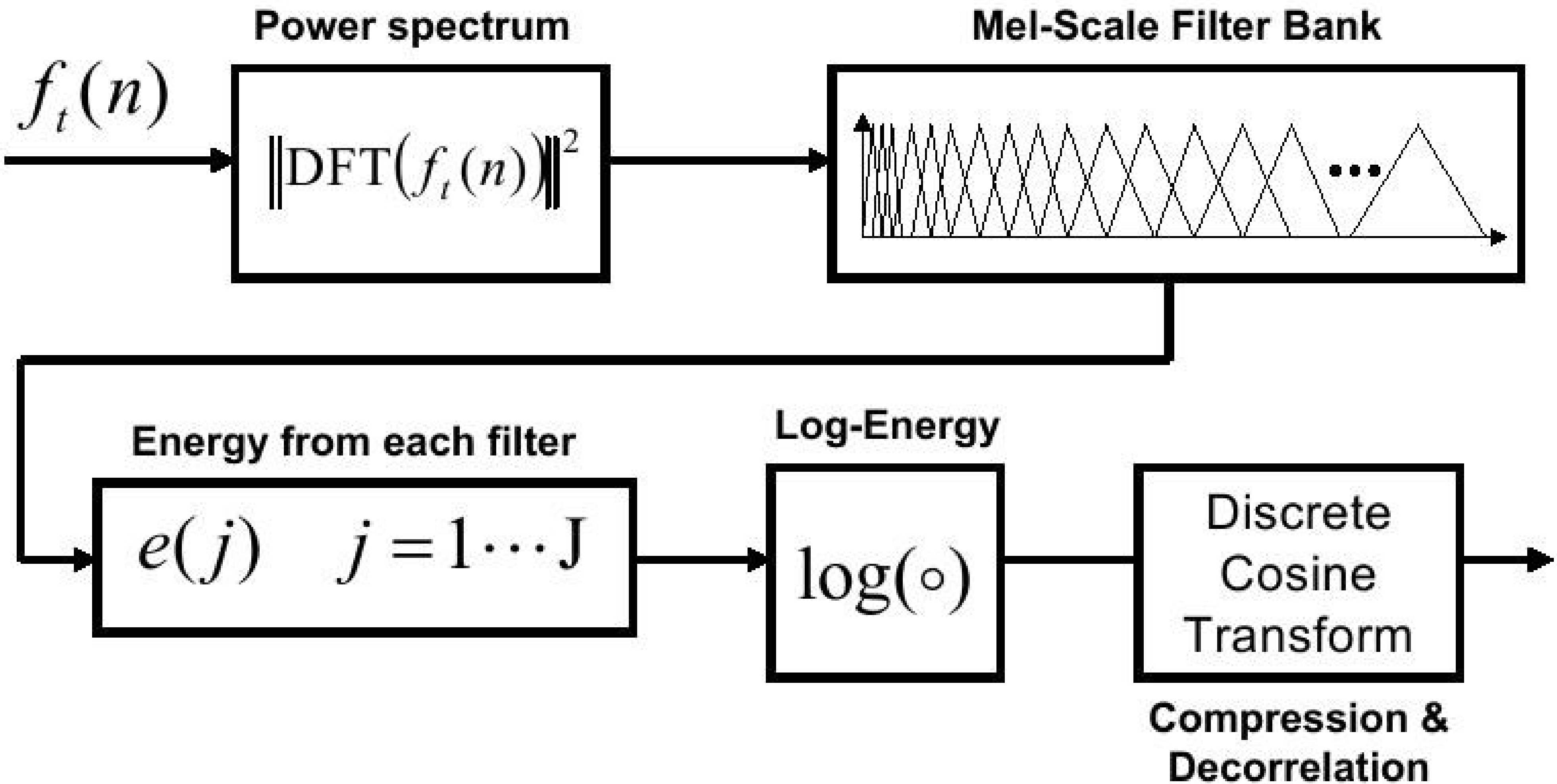
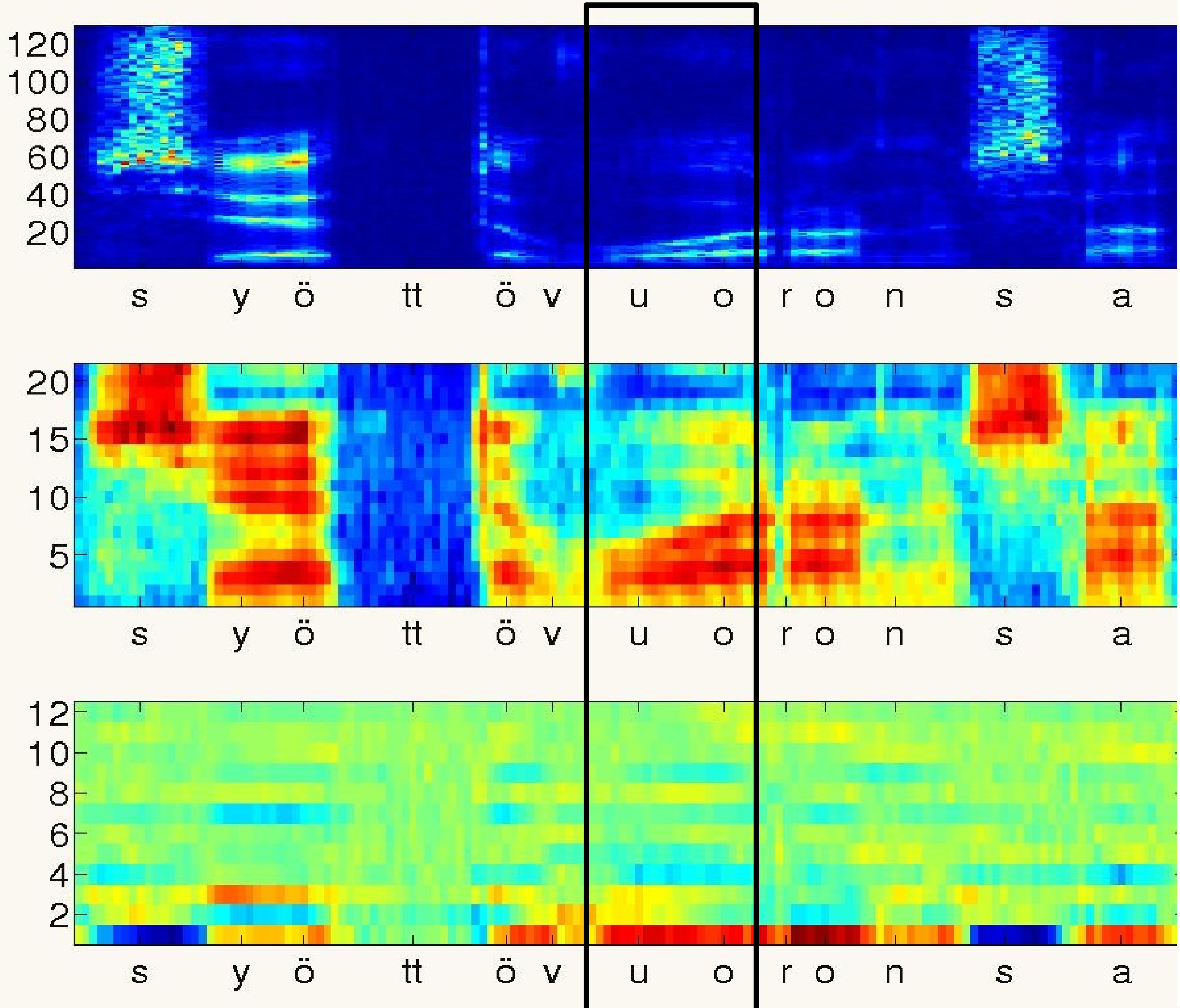


Fig. 5.7 Weighting functions for Mel-frequency filter bank.

Computation of MFCC (Mel Frequency Cepstral Coefficients)



Picture by B.Pellom

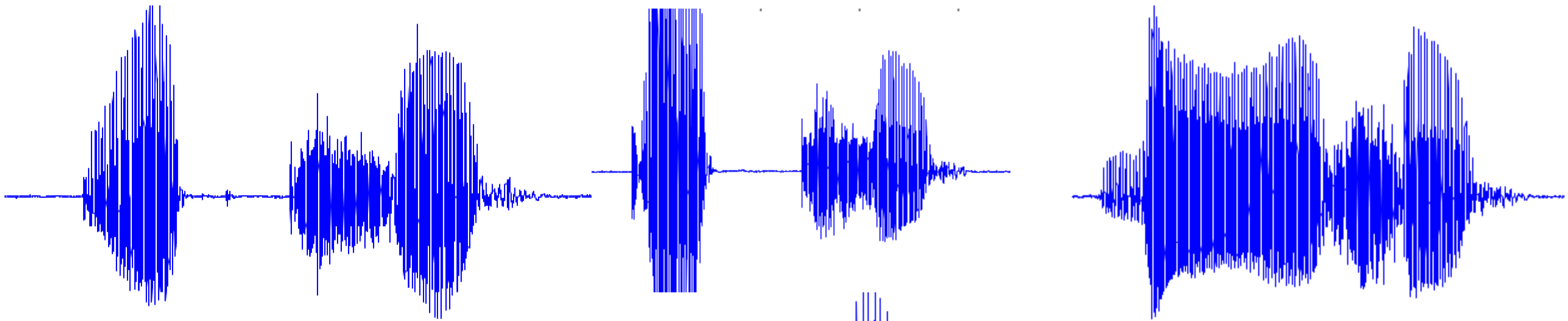


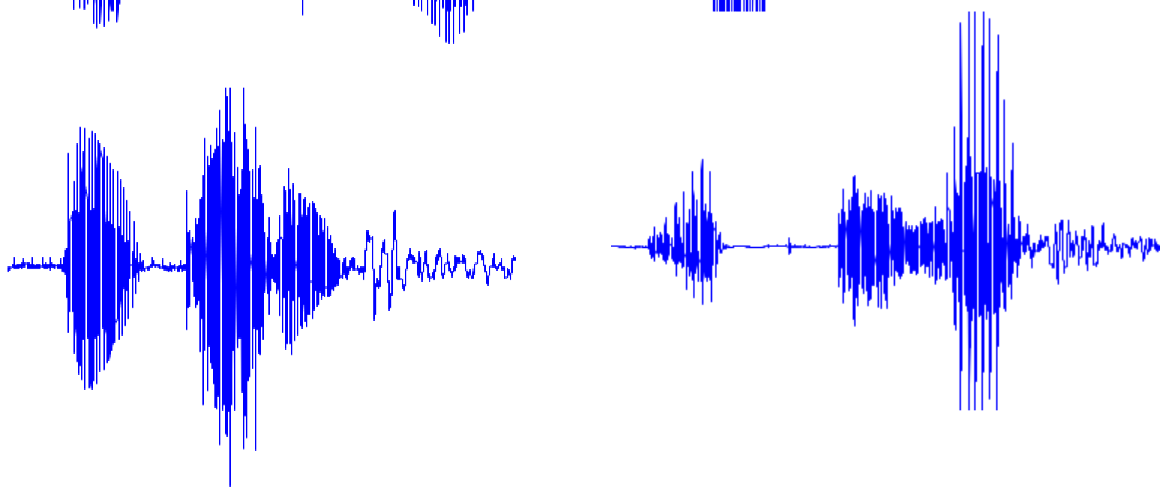
1. Frames:
short 10ms windows
2. FFT:
power spectrum
spectrogram

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

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

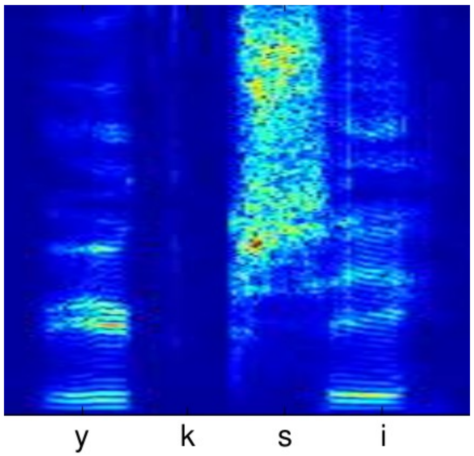
5 speech samples

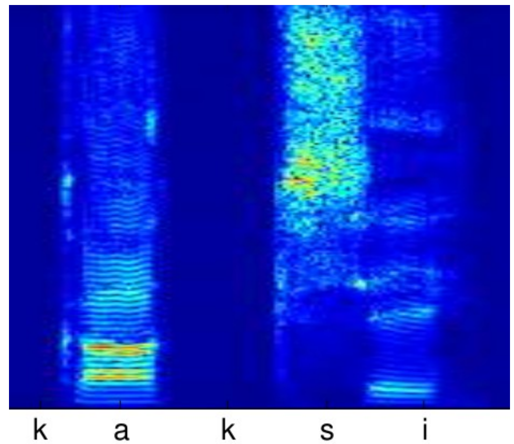


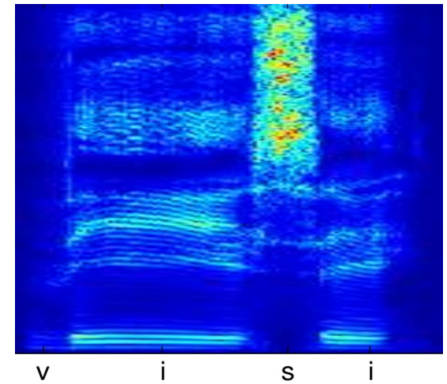


Very difficult to recognize speech from **this picture**...

Power spectrogram





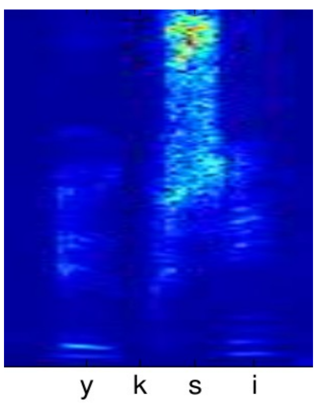


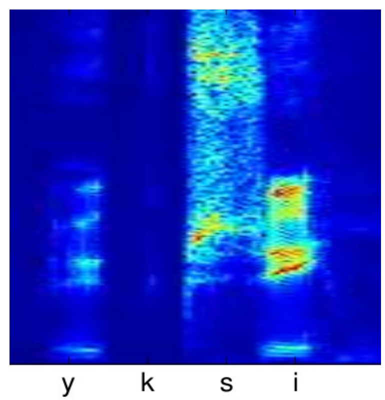
Speech recognition possible

Lot of data

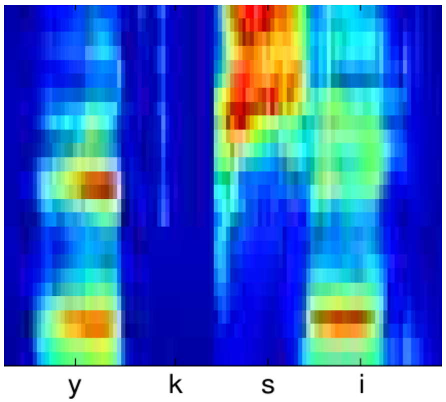
Lot of redundancy

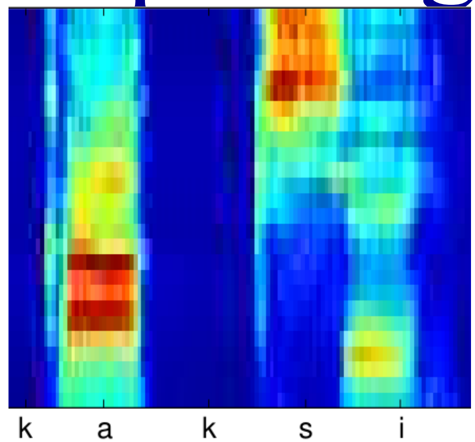
Lot of noise

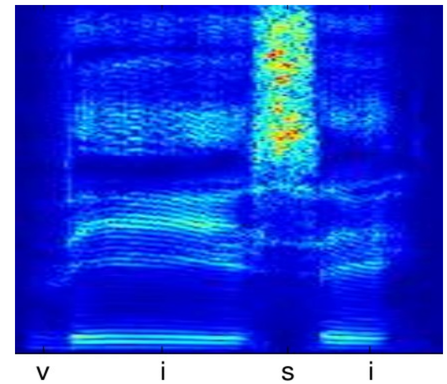


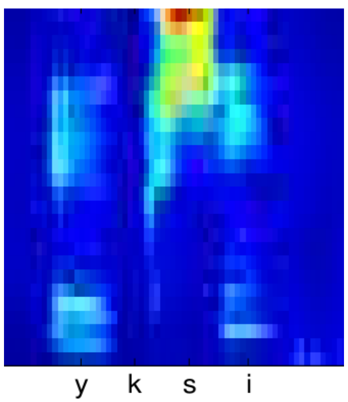


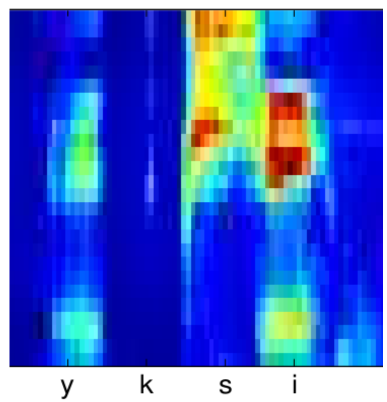
Mel spectrogram











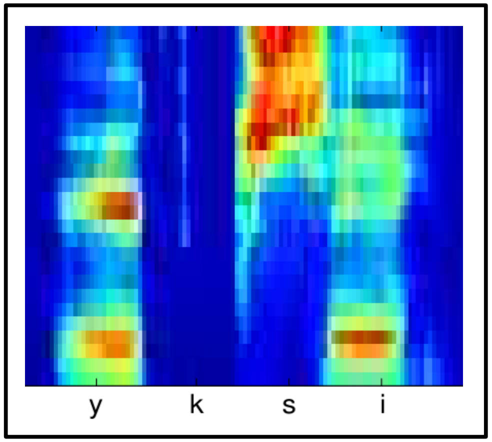
Speech recognition maybe easier?

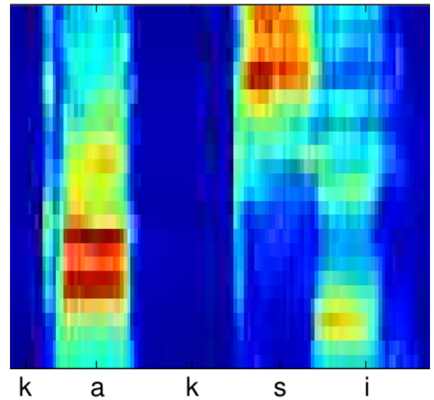
10 x less data

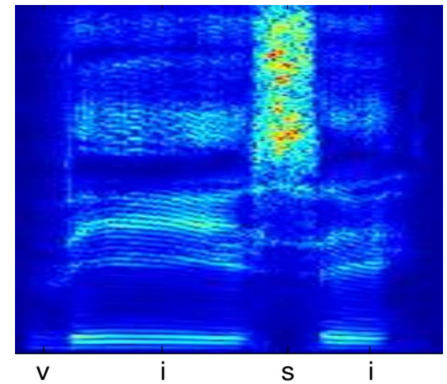
Less redundancy

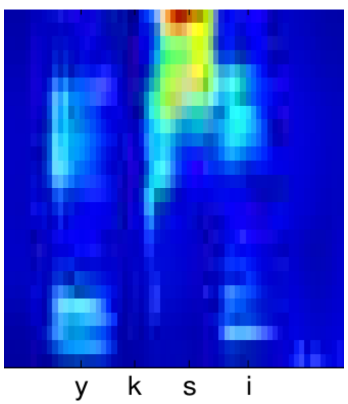
Less noise

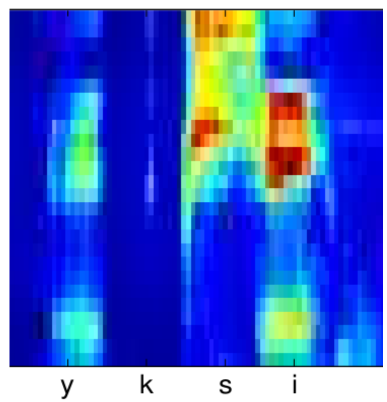
Mel spectrogram



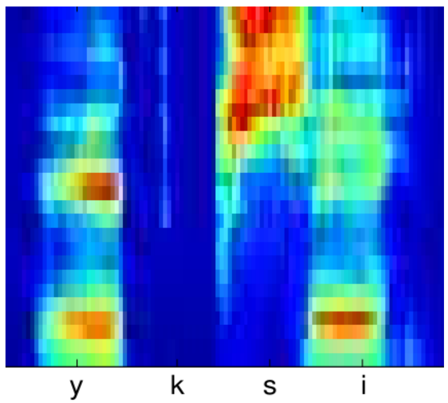


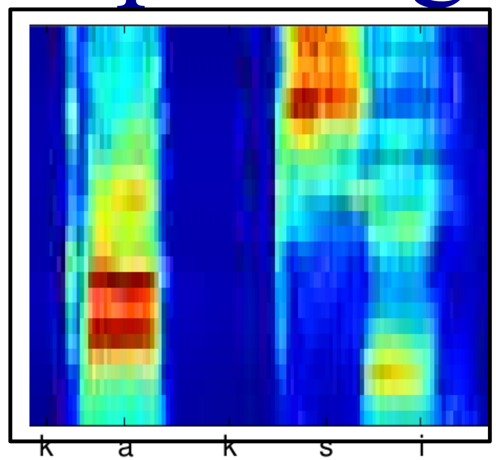


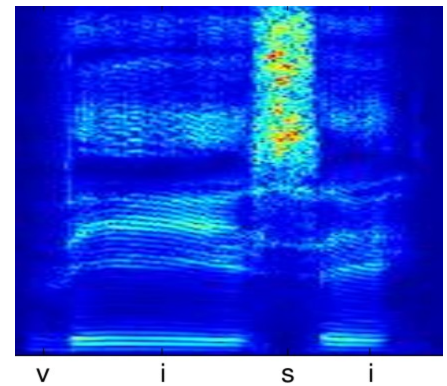


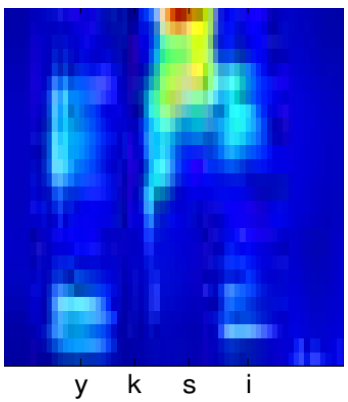


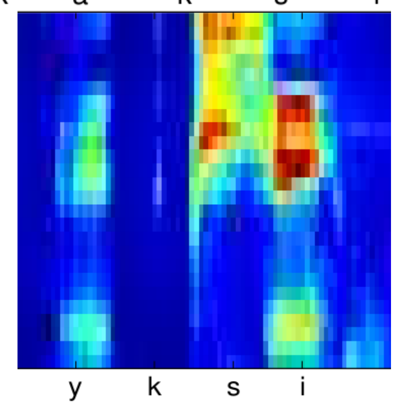
Mel spectrogram



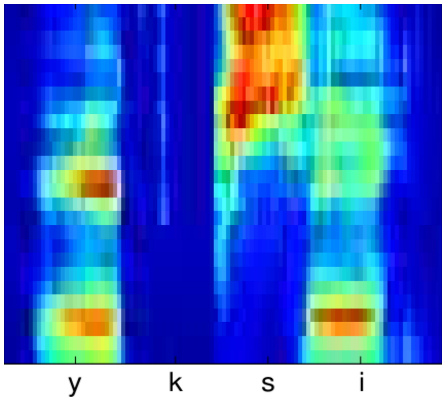


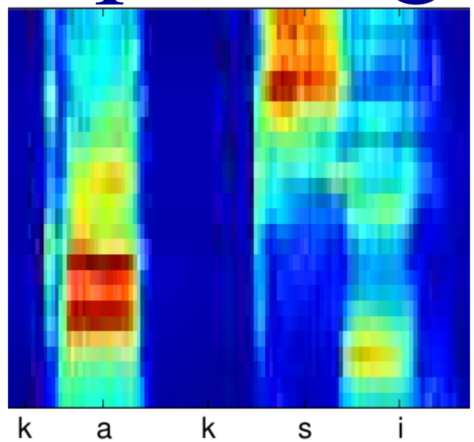


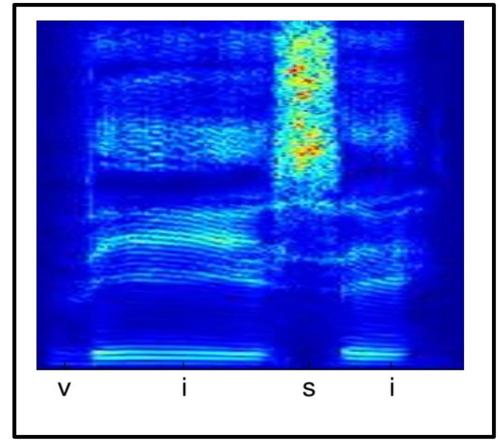


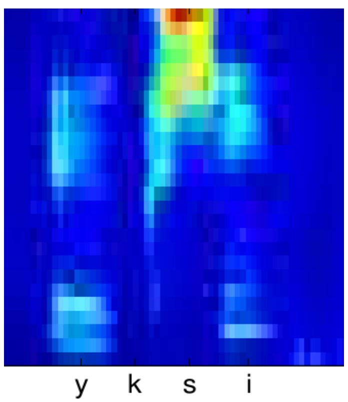


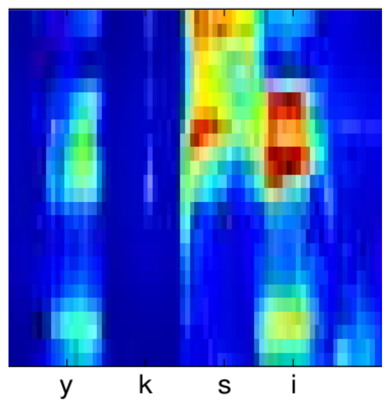
Mel spectrogram



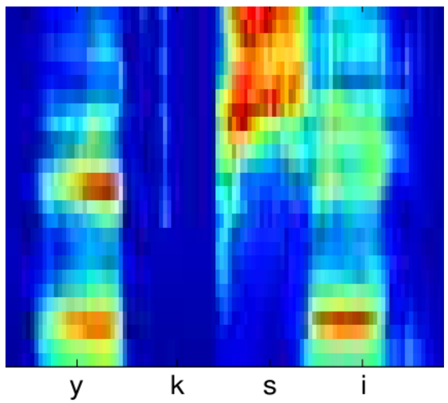


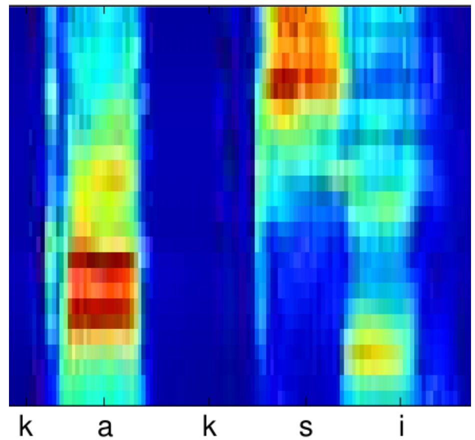


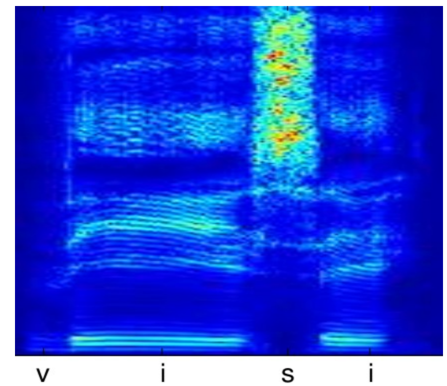


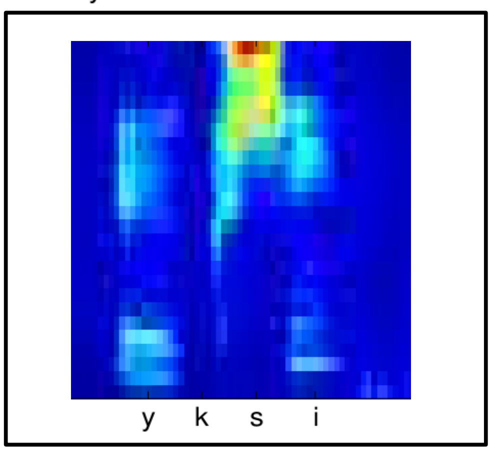


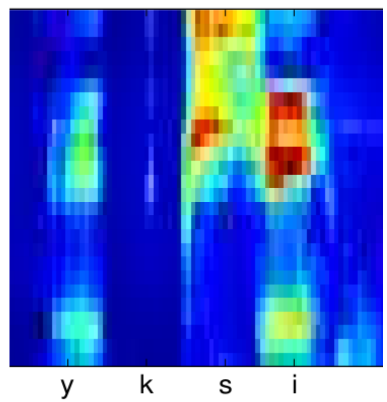
Mel spectrogram



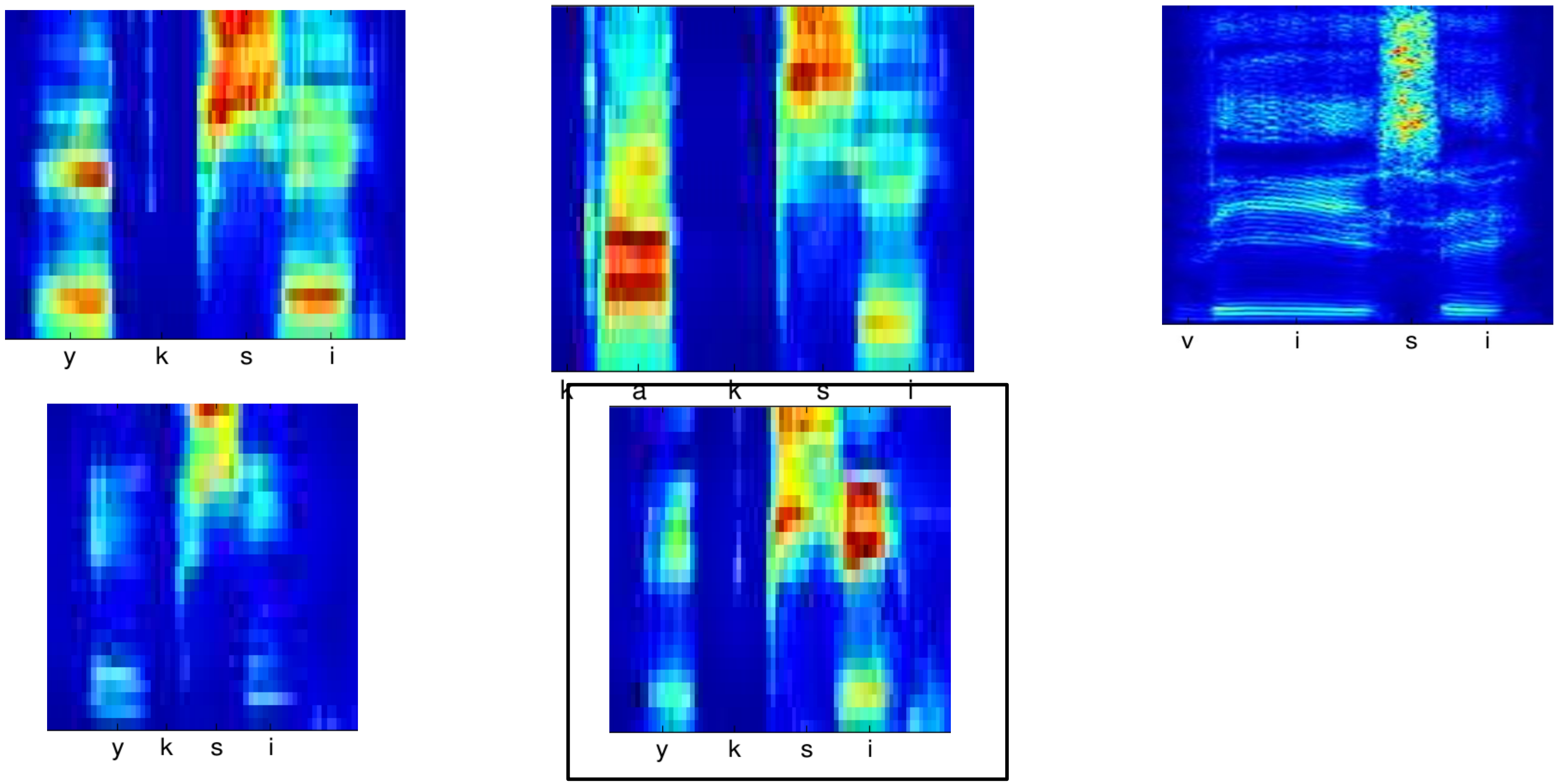




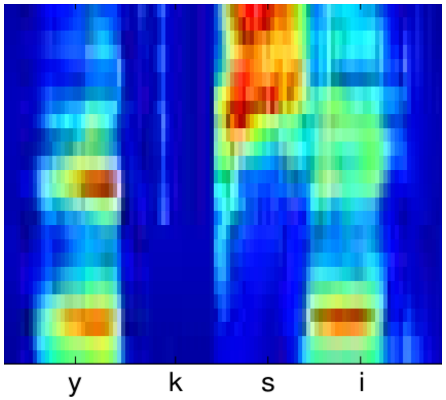


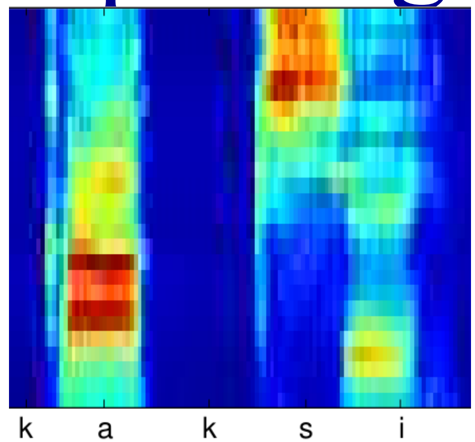


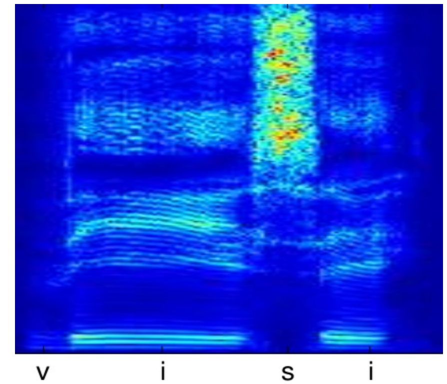
Mel spectrogram

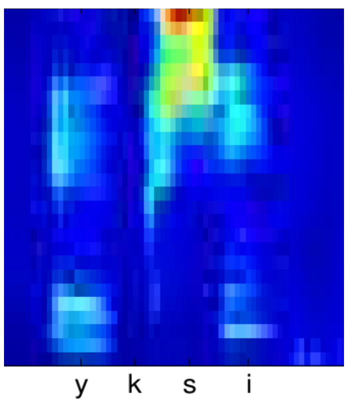


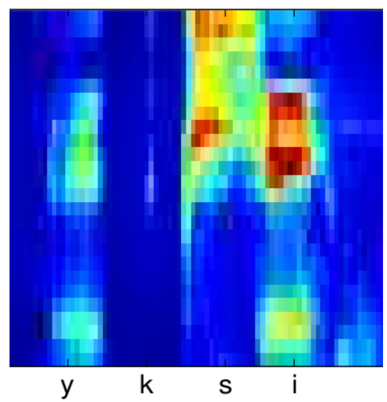
Mel spectrogram



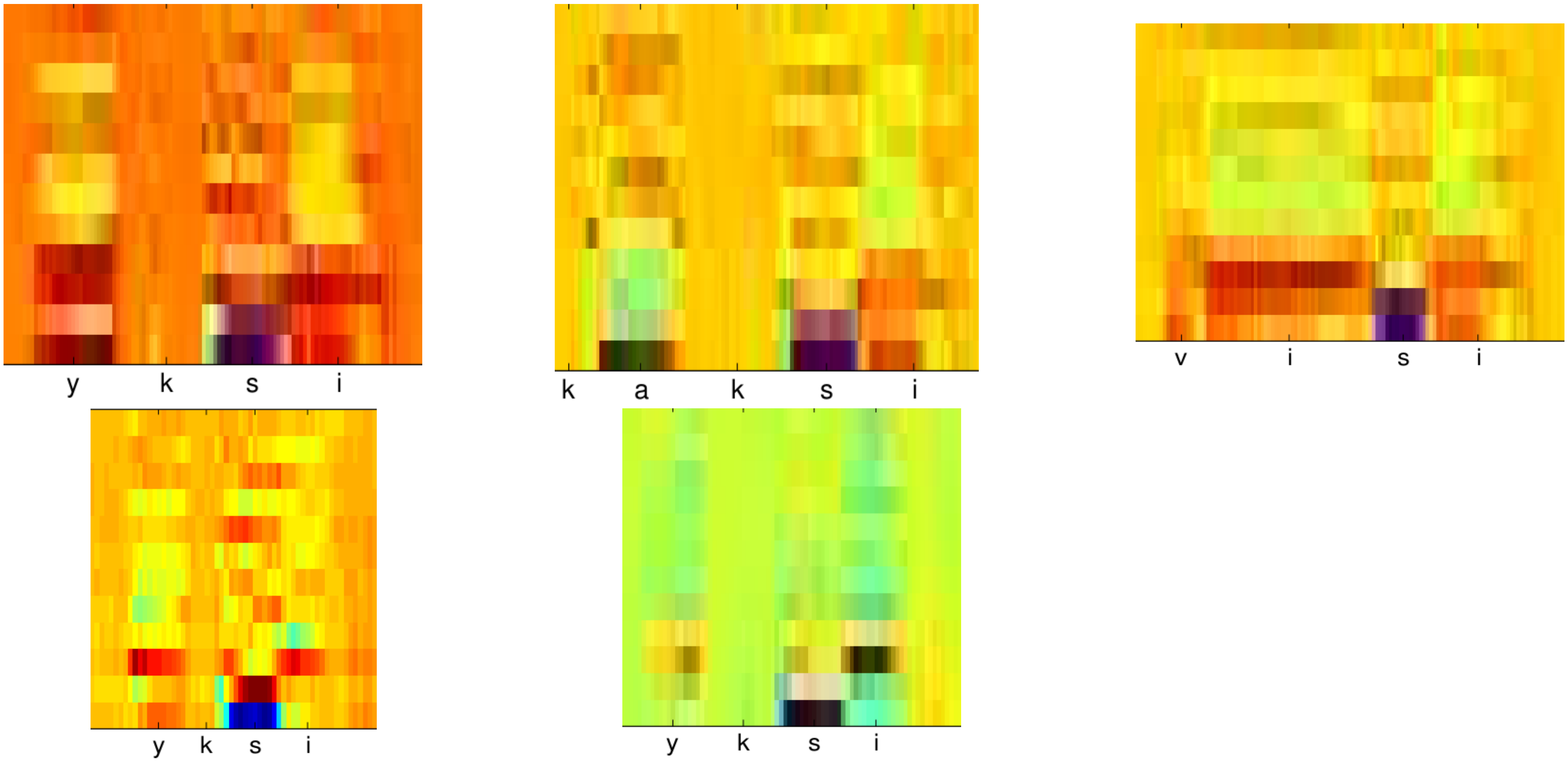




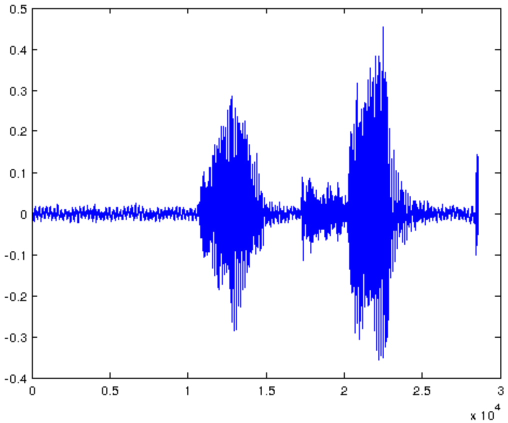


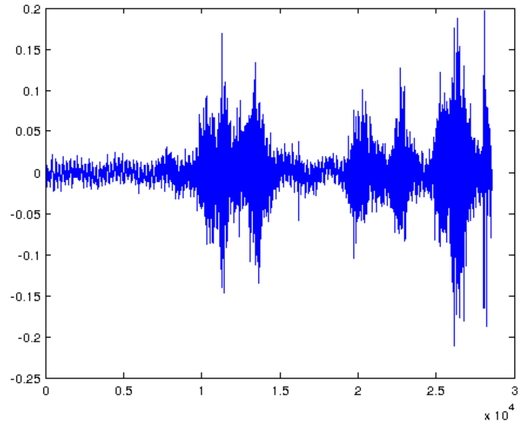


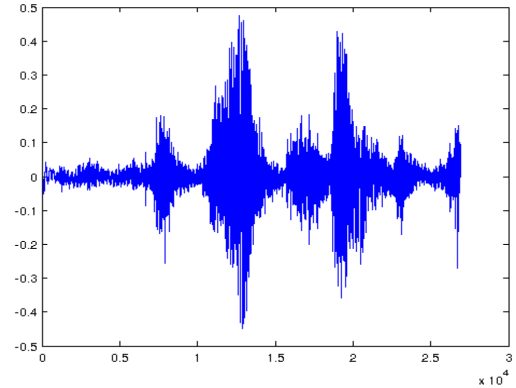
Mel-frequency cepstral coefficients (MFCC)



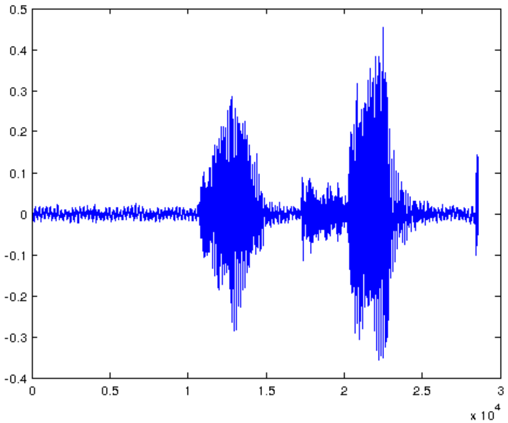
Background noise?

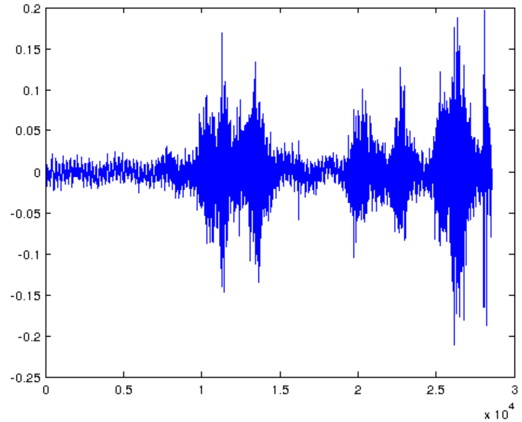


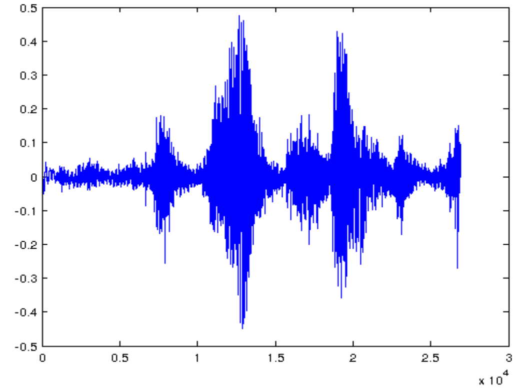


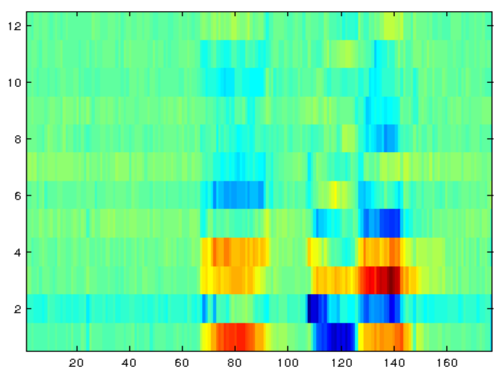


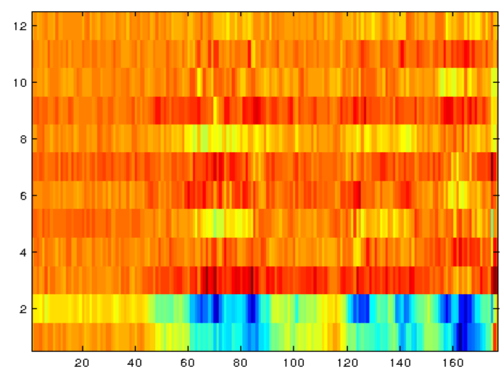
Background noise?

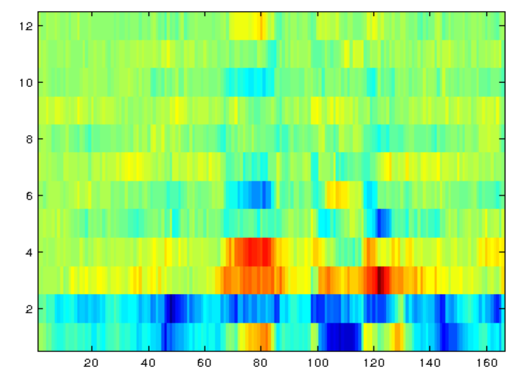




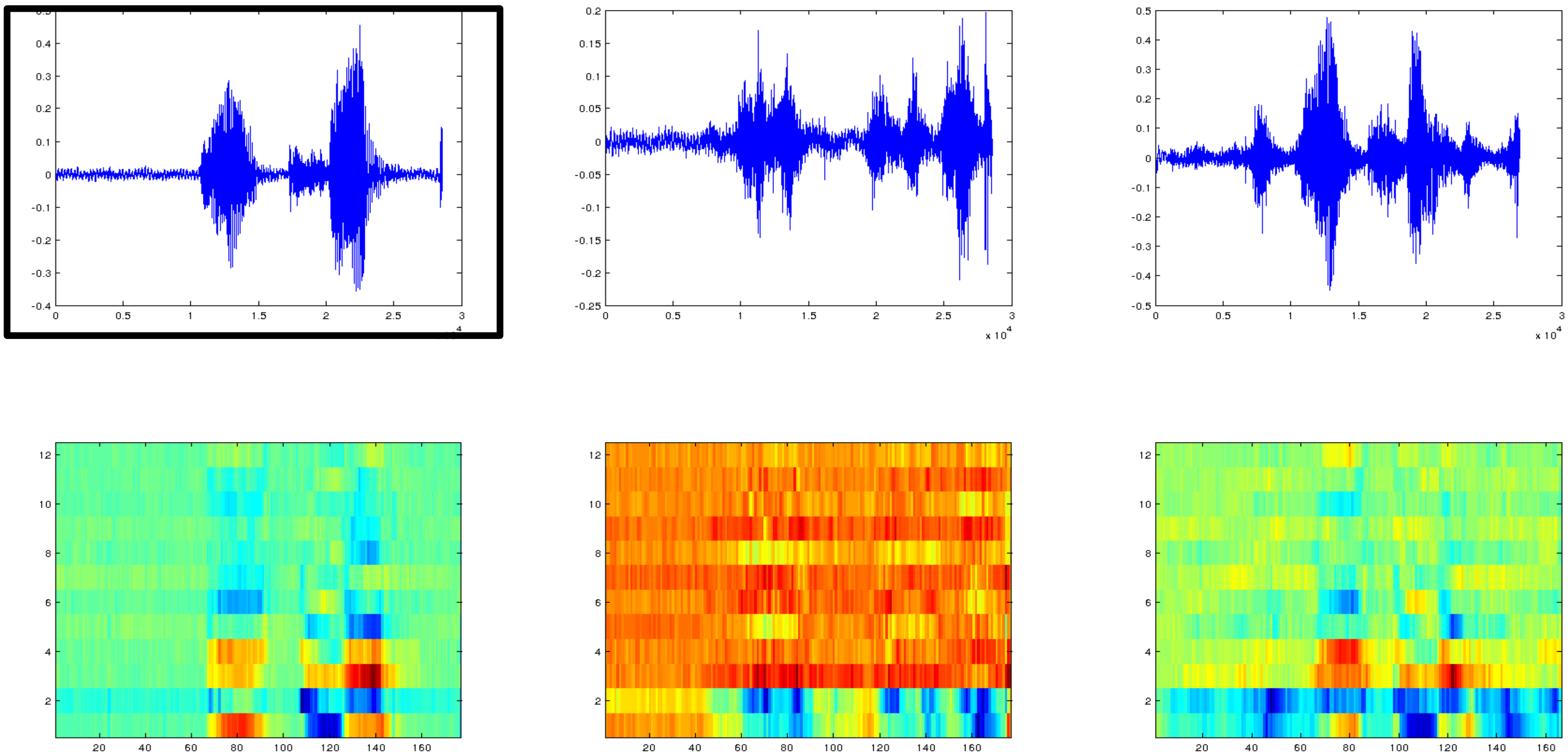




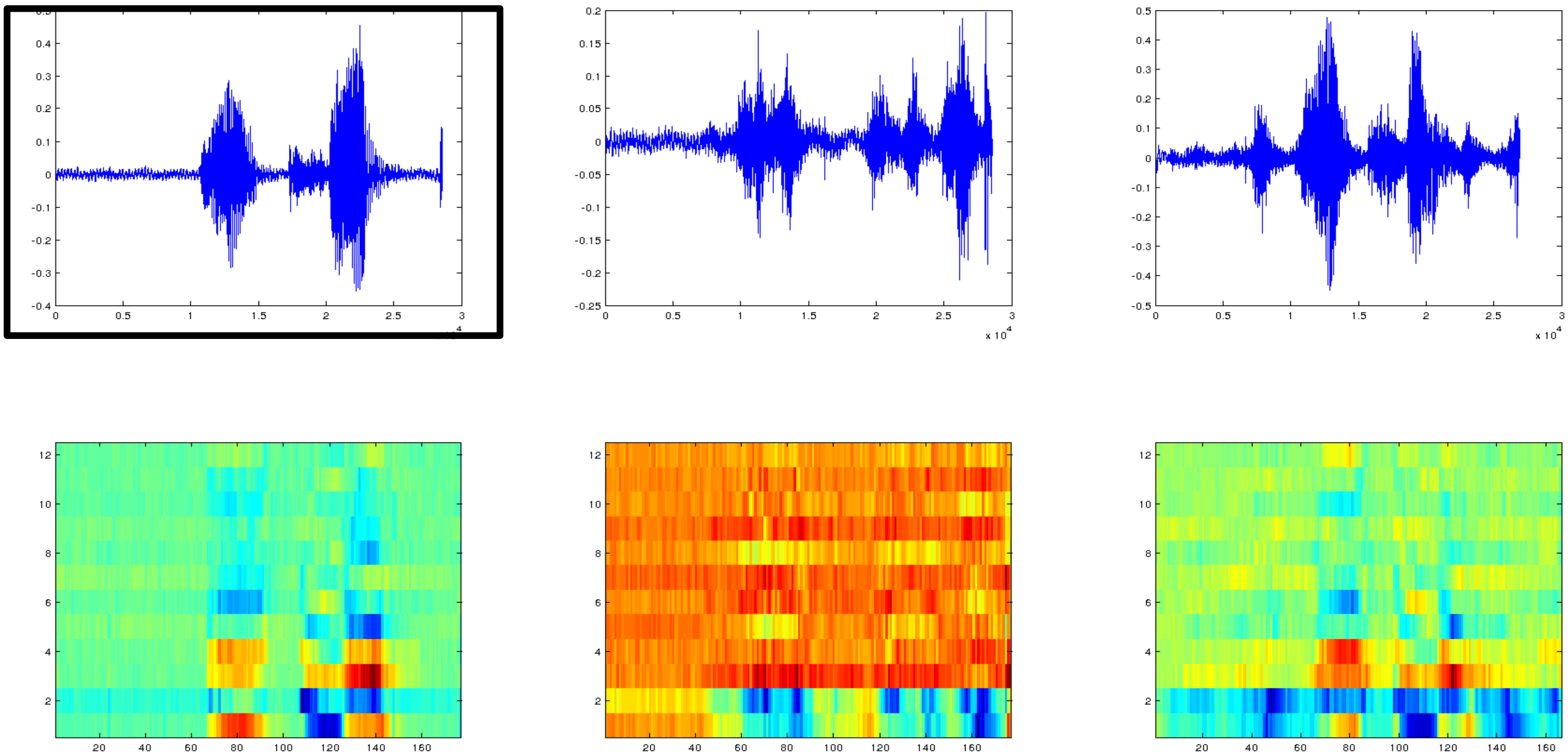




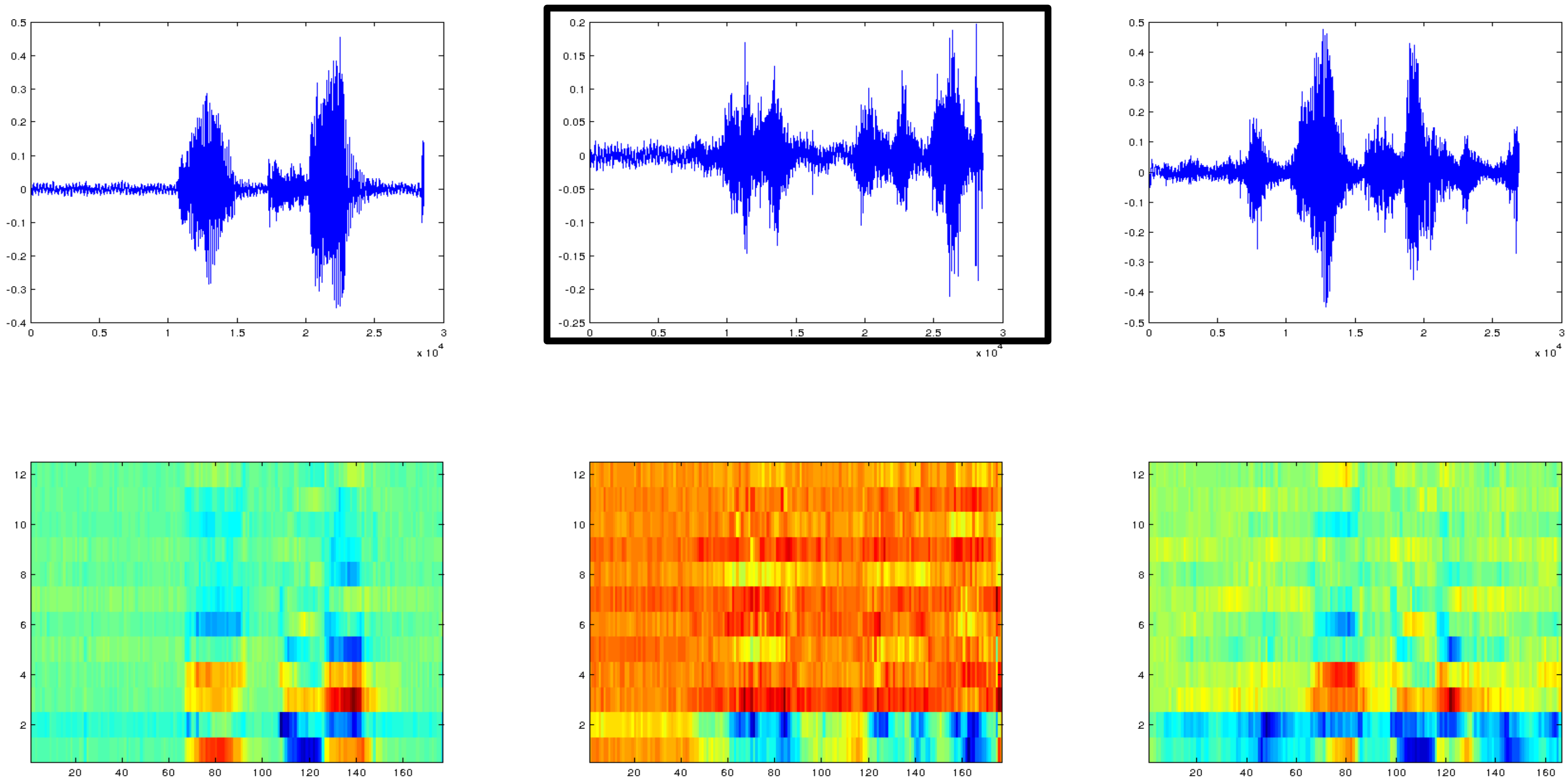
Background noise?



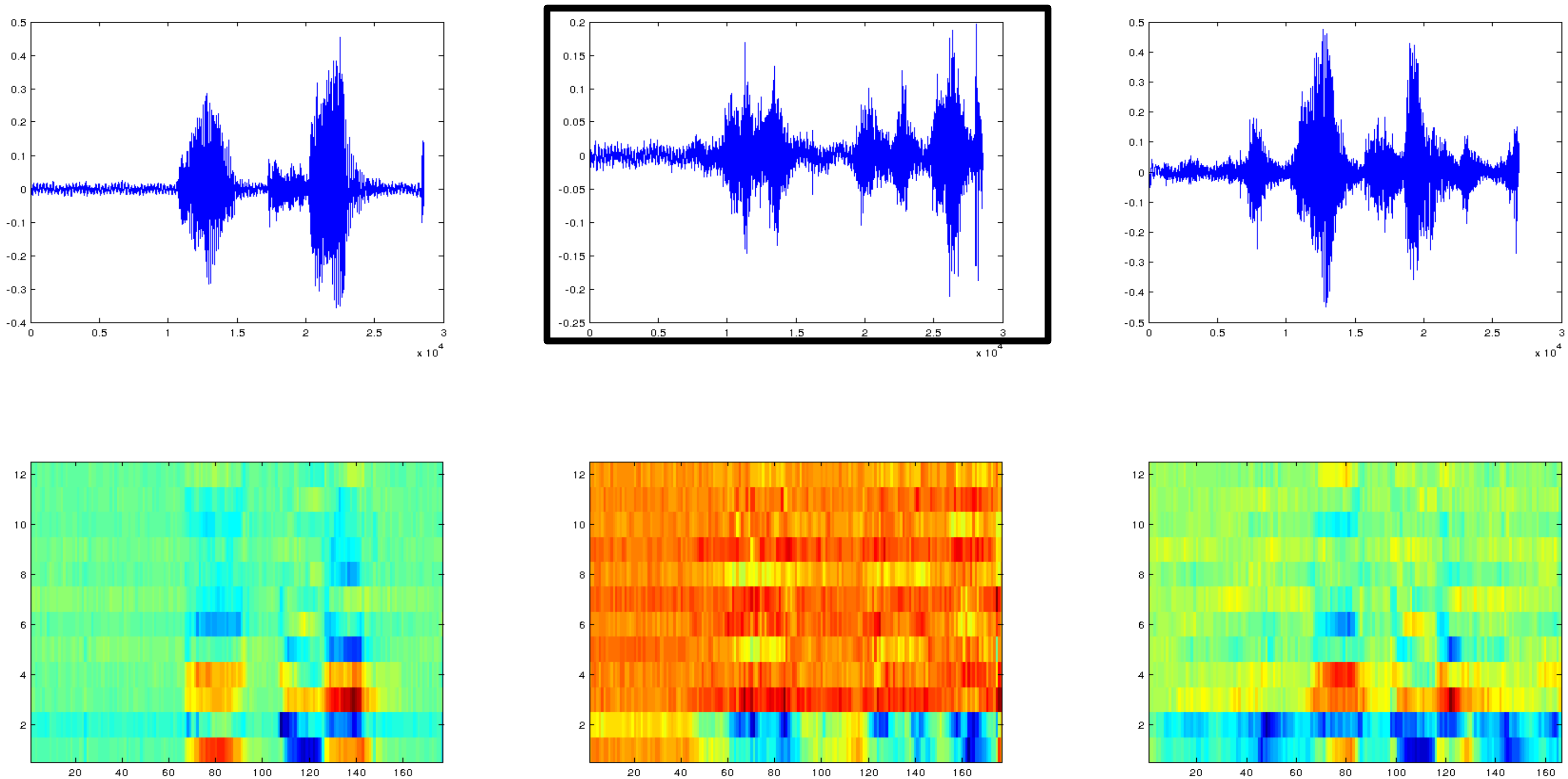
Background noise?



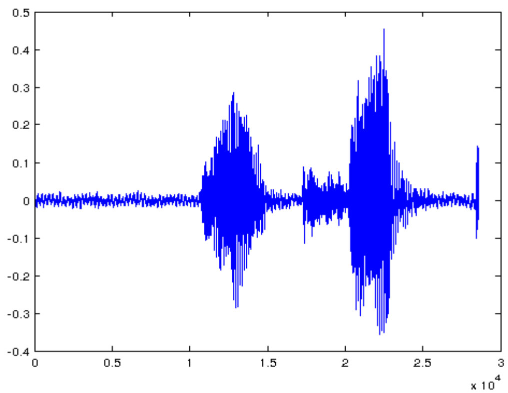
Background noise?

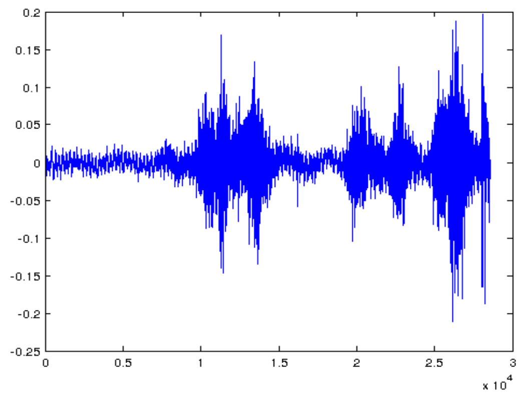


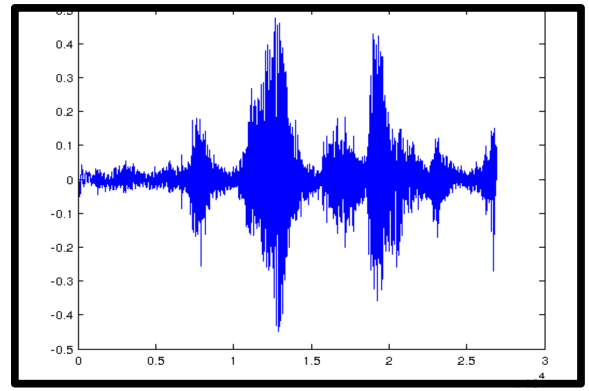
Background noise?

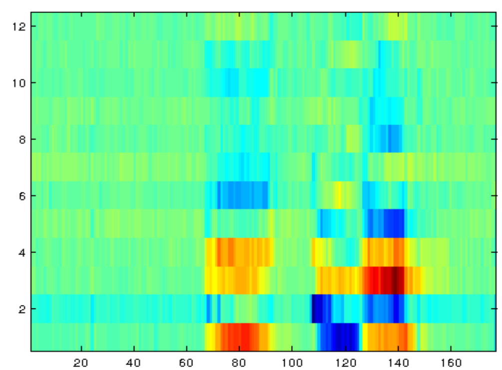


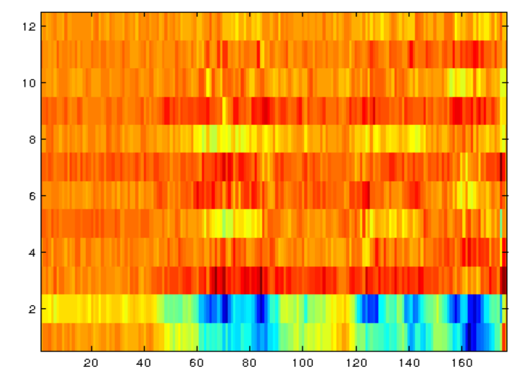
Background noise?

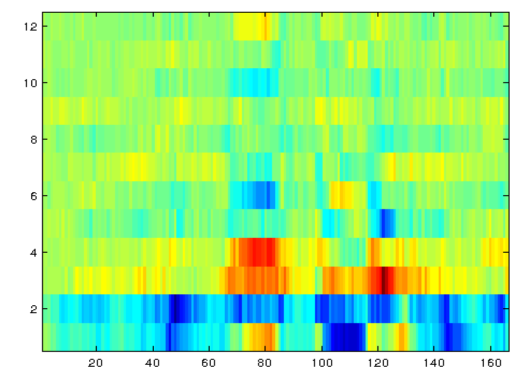


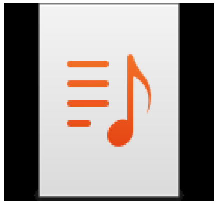




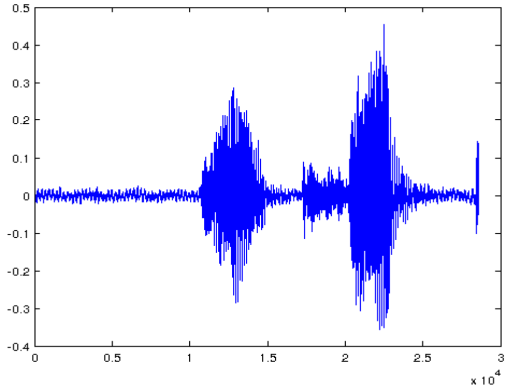


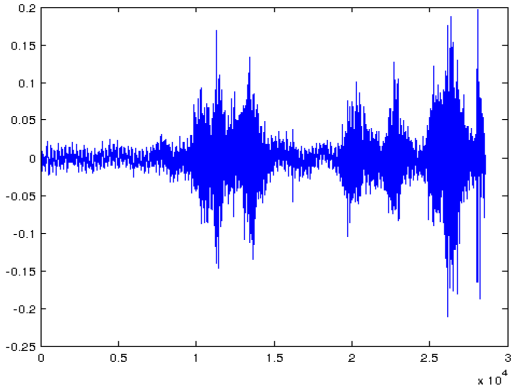


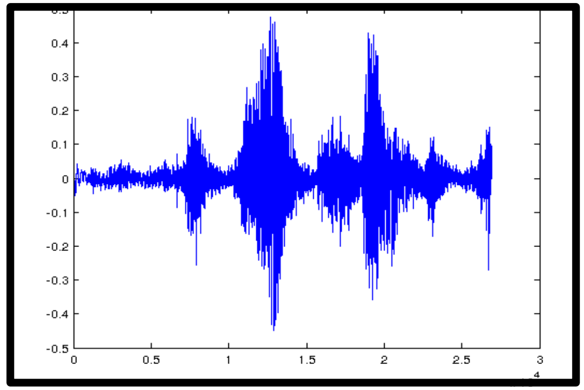


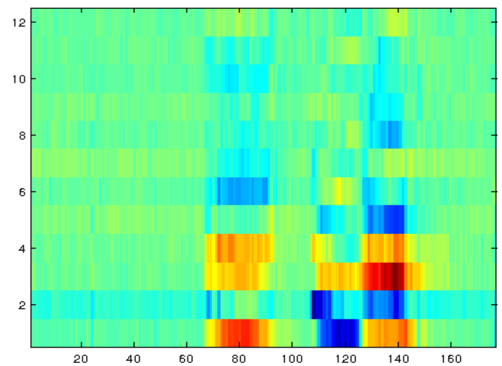


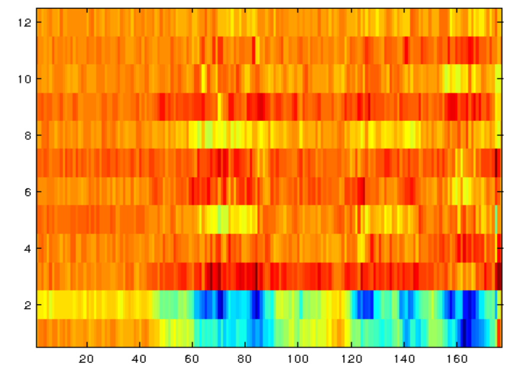
Background noise?

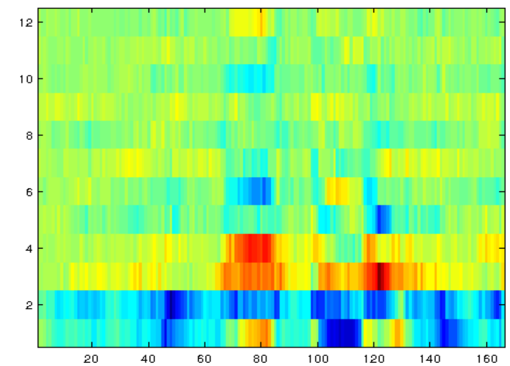












To classify speech sounds by features?

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

Real and complex cepstrum

- Classic: Real Cepstrum (RC)
  - symmetric
- Generalization: Complex Cepstrum (CC)
  - CC saves the *phase information* of the signal shape
  - Has also an anti-symmetric component
  - CC <u>coefficients are still always real</u>

Definitions

- **Real Cepstrum**: ($x[n]$ infinite sequence in time)

$$c[m] = \mathcal{F}^{-1}[\mathrm{Log}[\,|X[k]|\,]]\,[m] =$$

$$\mathcal{F}^{-1}[\mathrm{Log}[\,|\mathcal{F}[x[n]]|\,]]\,[m]$$

Note that we take the Magnitude spectrum!

- **Complex Cepstrum**:

$$y[m] = \mathcal{F}^{-1}[\mathrm{Log}[X[k]]]\,[m] =$$

$$\mathcal{F}^{-1}[\mathrm{Log}[\,\mathcal{F}[x[n]]]]\,[m]$$

Linear prediction LP

LP-model: $G / (1 - a_1 z^{-1} - a_2 z^{-2} \dots - a_p z^{-p}) = H[z]$

- $x[n]$ *causal and minimum phase (impulse response)*

$y[0] = c[0] = \text{Log}[G]$ (Markel & Gray)

LP coefficients can be transformed **to cepstral coefficients** by:

$$\boldsymbol{y[0] = \text{Log}[G], \quad y[1] = a[1],}$$

$$\boldsymbol{y[m] = a[m] + \sum_{t=1, m-1} [(t/m)\, y[t]\, a[m-t]]}$$

$1 < m \le p$, where $a[m]$ is m's LP coefficient

Real cepstrum c[m] can be computed from $y[m]$:

$c[0] = y[0]$, $c[m] = y[m]/2$, $0 < m \le p$

Intuition

- Source-Filter Theory: $X(w) = S(w)\,H(w)$
- Real cepstrum: $\text{Log}[|X(w)|] = \text{Log}[|S(w)|] + \text{Log}[|H(w)|]$
- The effects of source and filter in logarithmic spectrum are additive => can be separated by linear transformation, if they occur at different bands
- Voiced source produces a comb structure (fast variation in frequency), filter adjusts its envelope (slow variation in frequency)
- Fast and slow variations in frequency can be separated by a new Fourier transform (IFT)!

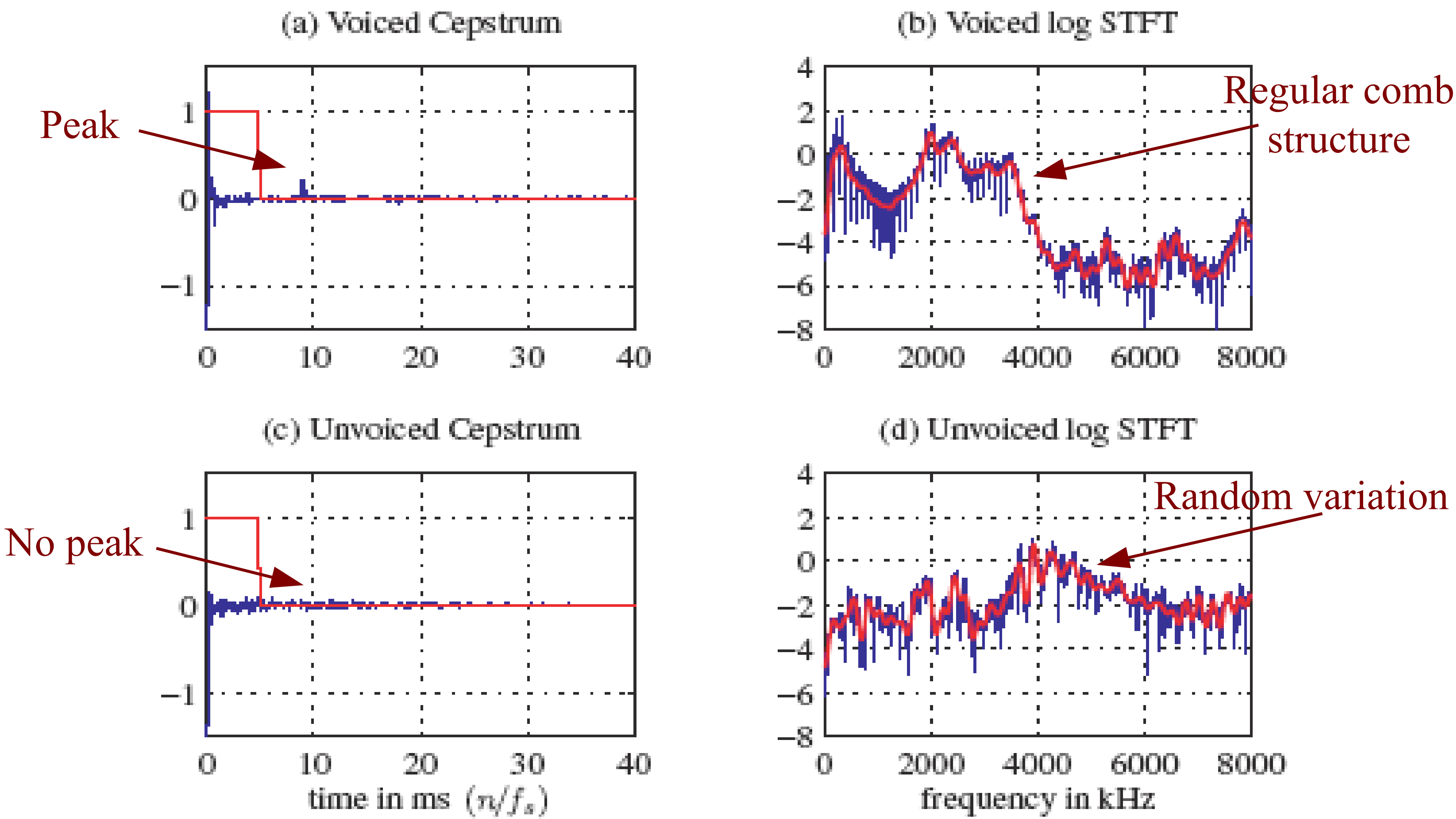
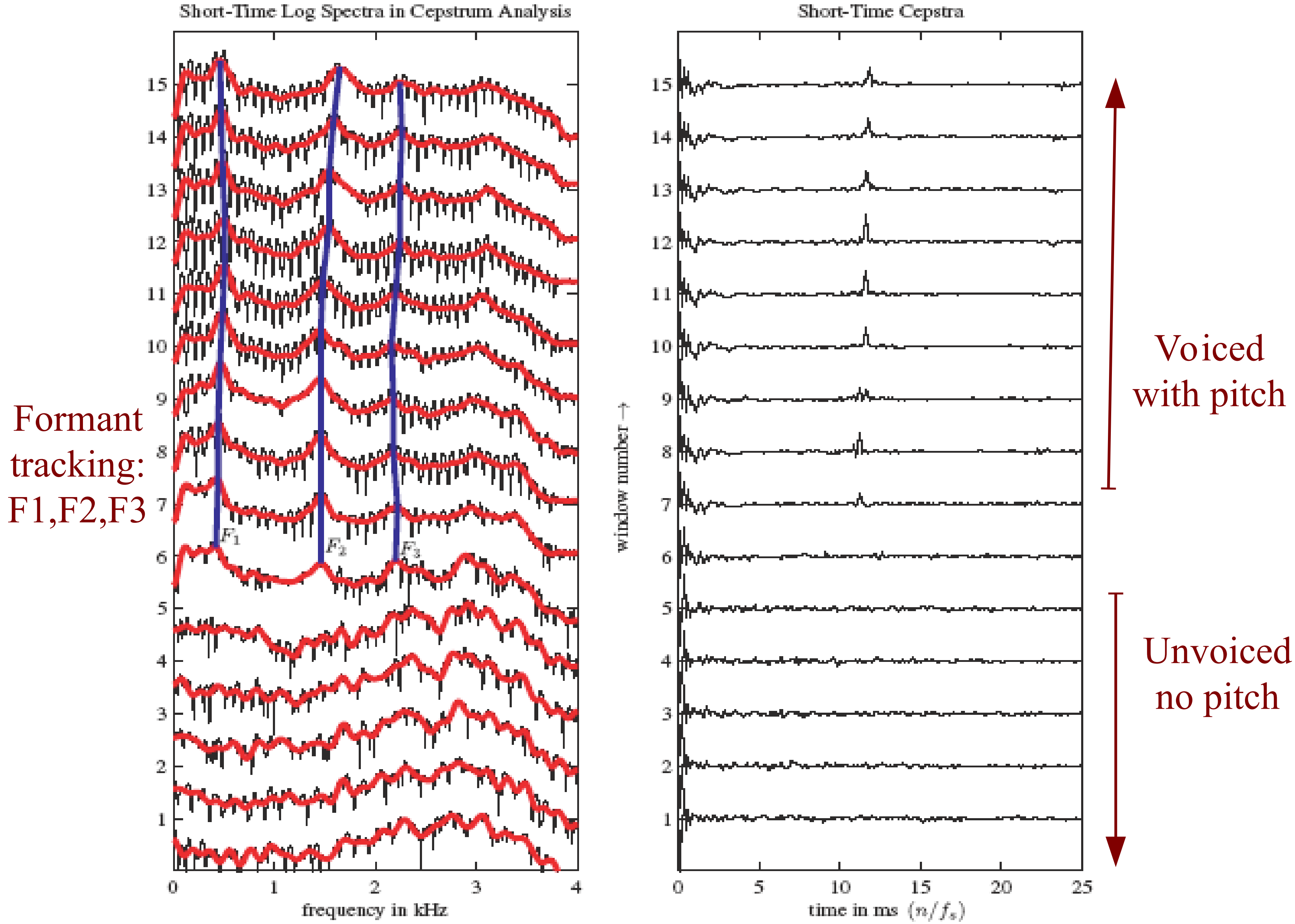


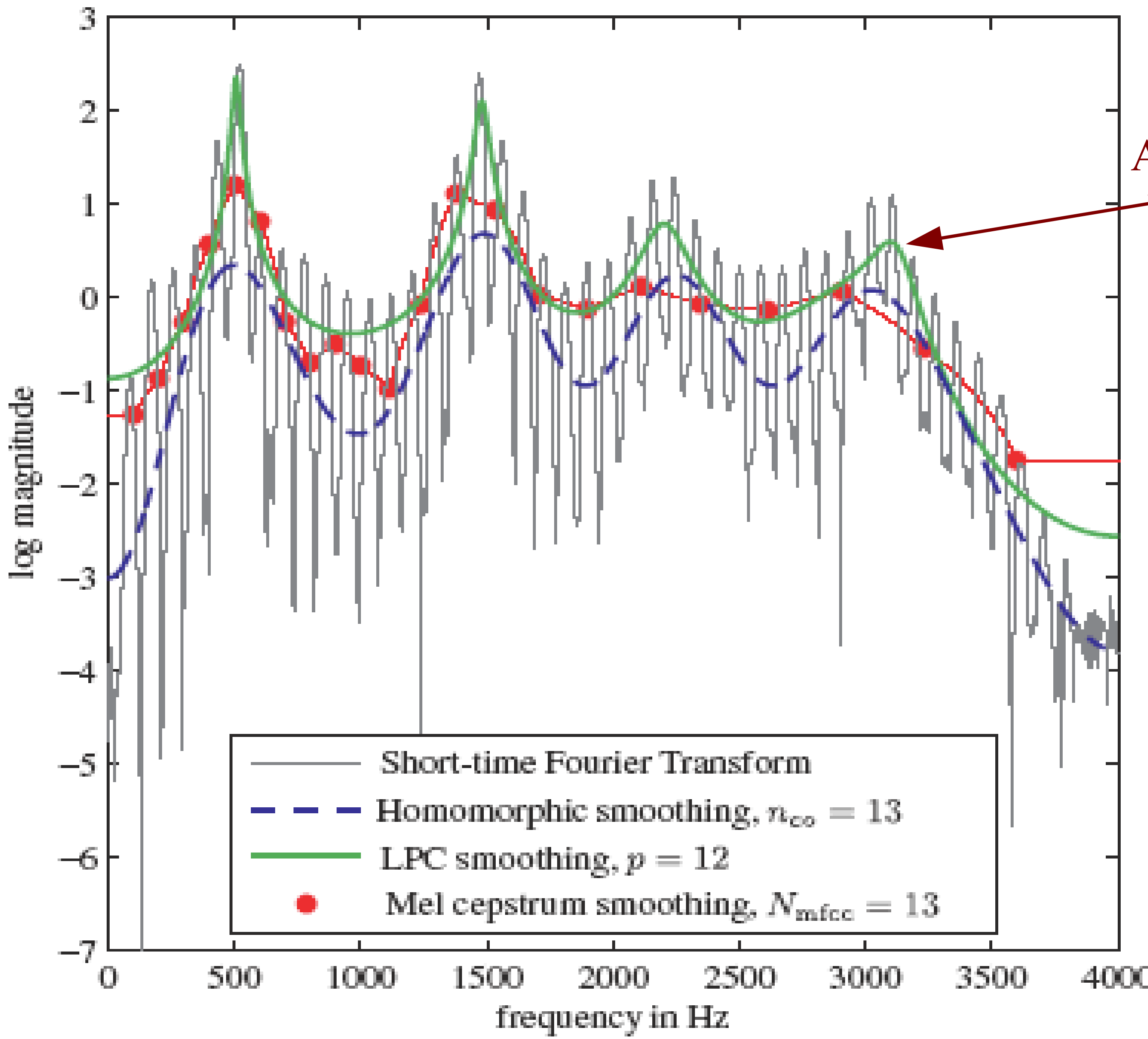
Fig. 5.5 Short-time cepstra and corresponding STFTs and homomorphically-smoothed spectra.

Picture by L.R.Rabiner



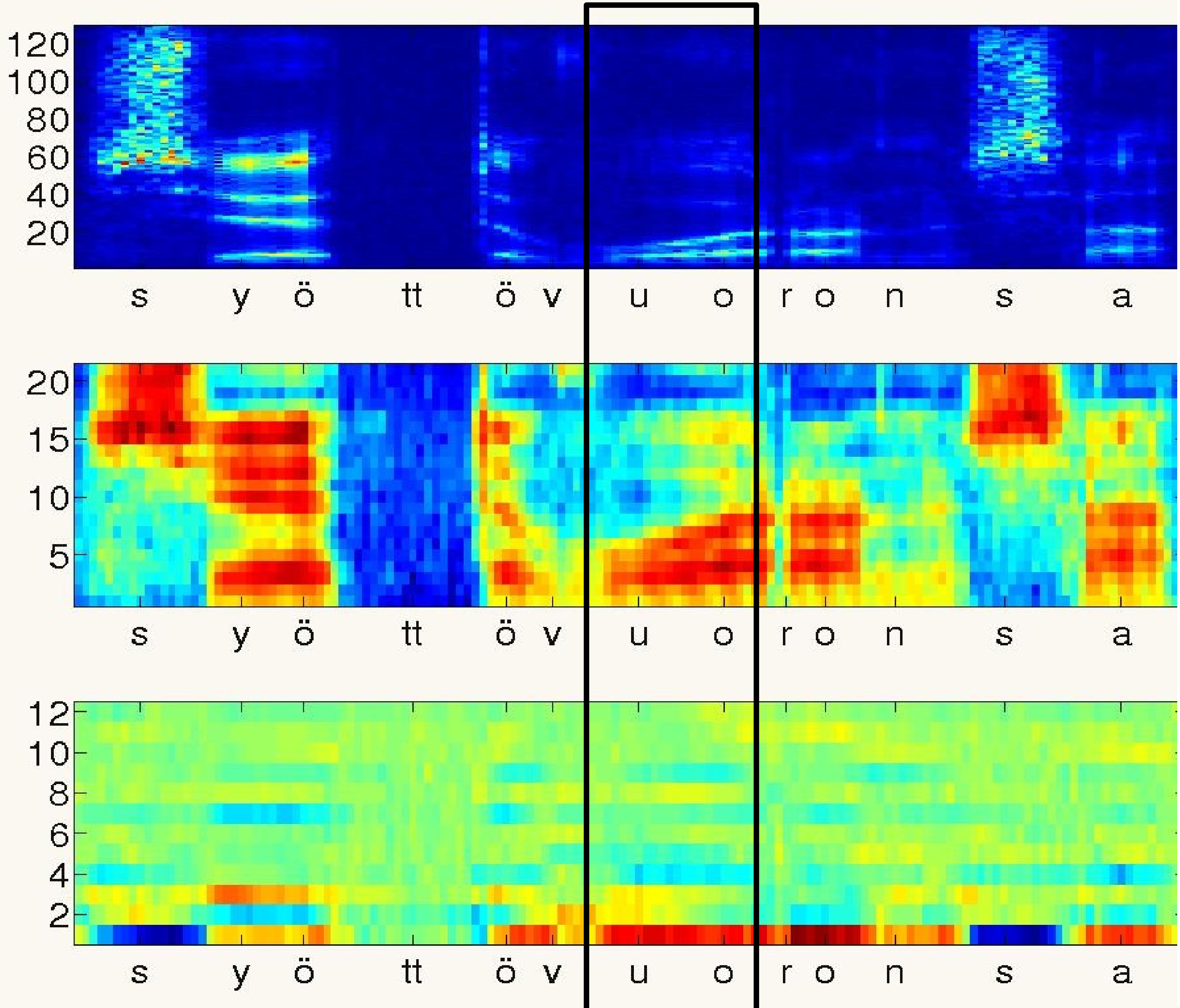
Picture by L.R.Rabiner

Fig. 5.6 Short-time cepstra and corresponding STFTs and homomorphically-smoothed spectra.



All have peaks at formant frequencies

Picture by L.R.Rabiner



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

Delta cepstrum

- Speech is dynamic, one way to capture that is taking the time derivatives of the short-time cepstrum
- First derivative = delta cepstrum
- Second derivative = delta-delta cepstrum
- The simplest way of computing the derivative is just the difference of two neighboring cepstral vectors: c[t] - c[t-1]
- The simple difference is very noisy, rather make a least-squares approximation to the local slope (smoothed difference including several neighbors with suitable weights)

Exercise, DL 22 April, 2022

1. Compute a cepstrum of a vowel segment and detect the formants
2. Compute the cepstrum from the LPC coefficients and compare it to the cepstral transformation
3. Compute the cepstrum of vowel /a/ and /i/ segments and classify the frames in the segments by using the distance between the cepstra
4. Compute the cepstrum of /a/, /m/, /k/ and /s/ segments and recognize the phonemes by using the distance to phoneme templates based on A. average of all frames and B. center frame.

See MyCourses for details and guiding.

3. Vector space representation of words

- why to represent words as vectors?
- how to do it?

Contents

Cepstrum

- Literature and other material
- Idea and history of cepstrum
- Cepstrum and LP model
- Mel cepstrum
- Pitch detection, formant tracking
- Phoneme recognition
- Temporal (a.k.a. delta) features

Word2vec

- meaning of words
- statistical semantics
- word-document matrix
- word-word matrix
- distributed semantics

The meaning of words?

- “The complete meaning of a word is always contextual, and no study of meaning apart from context can be taken seriously.” (Firth, 1939)
- “If we consider words A and B to be more different than A and C, then we will often find that the distributions of A and B are more different than the distributions of A and C. In other words, difference in meaning correlates with difference in distribution.” (Harris, 1954)
- “**You shall know the word by the company it keeps**” (Firth, 1957)

Statistical semantics

Statistical semantics hypothesis: Statistical patterns of human word usage can be used to figure out what people mean (Weaver, 1955; Furnas et al., 1983).

Bag of words hypothesis: The frequencies of words in a document tend to indicate the relevance of the document to a query (Salton et al., 1975).

Distributional hypothesis: Words that occur in similar contexts tend to have similar meanings (Harris, 1954; Firth, 1957; Deerwester et al., 1990).

Latent relation hypothesis: Pairs of words that co-occur in similar patterns tend to have similar semantic relations (Turney et al., 2003).

Representing documents in a matrix

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10
against	0	0	0	1	0	0	3	2	3	0
age	0	0	0	1	0	3	1	0	4	0
agent	0	0	0	0	0	0	0	0	0	0
ages	0	0	0	0	0	2	0	0	0	0
ago	0	0	0	2	0	0	0	0	3	0
agree	0	1	0	0	0	0	0	0	0	0
ahead	0	0	0	1	0	0	0	0	0	0
ain't	0	0	0	0	0	0	0	0	0	0
air	0	0	0	0	0	0	0	0	0	0
aka	0	0	0	1	0	0	0	0	0	0

"Very sparse. Each column is a **bag-of-words representation** of a document. In Web search: after suitable re-weighting, the documents (columns) can be ranked according to their match for a given query (set of rows)."

An example (Potts, 2013)

Representing words in a matrix

	against	age	agent	ages	ago	agree	ahead	ain.t	air	aka	al
against	2003	90	39	20	88	57	33	15	58	22	24
age	90	1492	14	39	71	38	12	4	18	4	39
agent	39	14	507	2	21	5	10	3	9	8	25
ages	20	39	2	290	32	5	4	3	6	1	6
ago	88	71	21	32	1164	37	25	11	34	11	38
agree	57	38	5	5	37	627	12	2	16	19	14
ahead	33	12	10	4	25	12	429	4	12	10	7
ain't	15	4	3	3	11	2	4	166	0	3	3
air	58	18	9	6	34	16	12	0	746	5	11
aka	22	4	8	1	11	19	10	3	5	261	9
al	24	39	25	6	38	14	7	3	11	9	861

More dense. Co-occurrences of words in a specified window of text. For example, in the same document, in the same sentence, or next to each other (in any order)

An example (Potts, 2013)

Modifying the vector spaces

The basic matrix formulation offers lots of variations:

- window sizes
- word weighting, normalization, thresholding, removing stopwords
- stemming, lemmatizing, clustering, classification, sampling
- distance measures
- dimensionality reduction methods
- neural networks

Projecting words to vectors, in practise:

2 initialization methods:

<u>1. "one-hot" vectors:</u>

[0 … 0 0 1 0 0 … 0]

every word in the vocabulary has its own dimension

orthogonal mapping

very high dimensional

very sparse

<u>2. random vectors:</u>

several floats or binary

low dimensionality (e.g. 100)

approximately orthogonal

less sparse

Projecting words to vectors, in practise:

2 ways to define distributed semantical representations:

1. "context vectors"

First compute a word-word matrix from a large text corpus

Then compute new word vectors by summing the columns, i.e. those words that appeared near them, and normalizing again

2. "word2vec"

First train a deep neural network from a large training data to perform a specific task (e.g. to predict a word given its context)

Then map the words into the first hidden layer, so-called "projection layer" and use its outputs as the word vectors

Tools

Gensim, Matlab, R, Python NLTK, MALLET, FACTORIE, word2vec, torch, tensor flow, and many more…

References

Deerwester, S., S. T. Dumais, G. W. Furnas, T. K. Landauer & R. Harshman. 1990. Indexing by latent semantic analysis. Journal of the American Society for Information Science 41(6). 391–407.

Firth, John R. 1935. The technique of semantics. Transactions of the Philological Society 34(1). 36–73.

Firth, John R. 1957. A synopsis of linguistic theory 1930–1955. In Studies in linguistic analysis, 1–32. Oxford: Blackwell.

Furnas, G. W., Thomas K. Landauer, L. M Gomez & S. T. Dumais. 1983. Statistical semantics: Analysis of the potential performance of keyword information systems. Bell System Technical Journal 62(6). 1753–1806.

Harris, Zellig. 1954. Distributional structure. Word 10(23). 146–162.

Potts, Chris. 2013.Distributional approaches to word meanings, Stanford course slidesLing 236/Psych 236c: Representations of meaning, Spring 2013.

Salton, Gerald, Andrew Wong & Chung-Shu Yang. 1975. A vector space model for automatic indexing. Communications of ACM 18(11). 613–620.

Turney, Peter D. & Michael L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems (TOIS) 21. 315–346.

Weaver, Warren. 1955. Translation. In William N. Locke & A. Donald Booth (eds.), Machine translation of languages: Fourteen essays, 15–23. Cambridge, MA: MIT Press.

Exercise, DL 22 April, 2022

1. Prepare text data, define the vocabulary and list word pairs that occur near each other
2. Train a neural network to predict the first word in each word pair given the context (the other word)
3. Take the hidden layer weights as your word embedding and test that it maps related words near each other in this vector space
4. Modify your system to see if you can improve the system
5. Compare your system to a reference word2vec system (train it with the same data) and a bag-of-words model

See MyCourses for details and guiding.