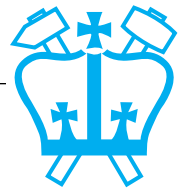
2 Spectral and cepstral models

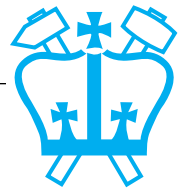
- **Spectrogram seems like a good representation**
  - long history
  - satisfying in use
  - experts can ‘read’ the speech
- **What is the information?**
  - intensity in time-frequency cells

→ **Discarded information:**
  - phase
  - fine-scale timing
- **The starting point for other representations**



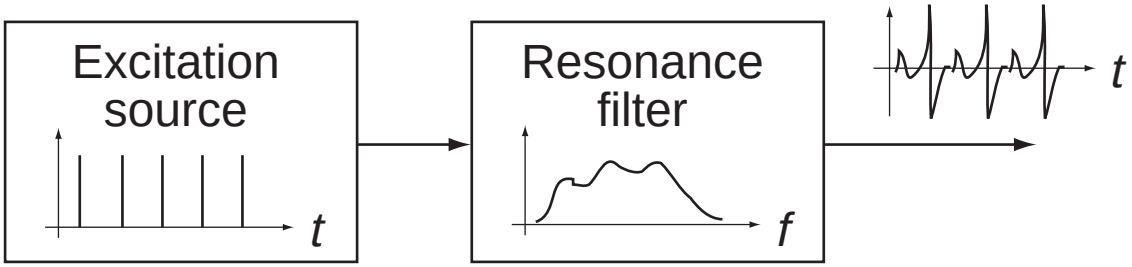
Limitations of spectral models

- **Not much data thrown away**
  - just fine phase/time structure (smoothing)
  - little actual 'modeling'
  - still a large representation!
- **Little separation of features**
  - e.g. formants and pitch
- **Highly correlated features**
  - modifications affect multiple parameters
- **But, quite easy to reconstruct**



The cepstrum

- **Original motivation: Assume a source-filter model:**



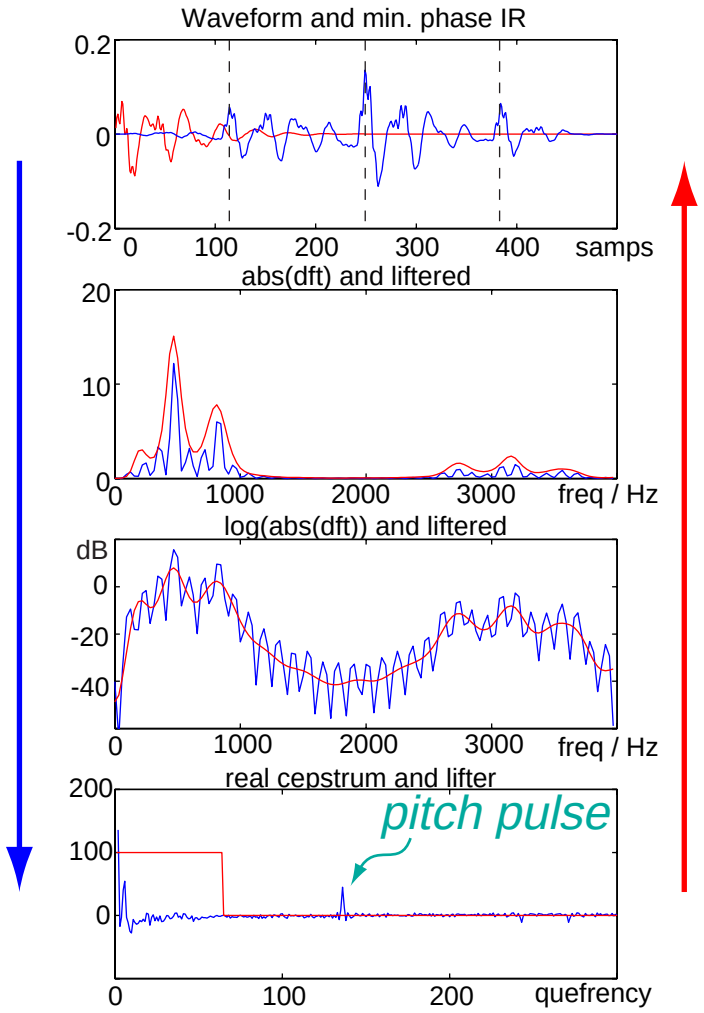
- **Define 'Homomorphic deconvolution':**
  - source-filter convolution: $g[n]*h[n]$
  - FT $\rightarrow$ product $G(e^{j\omega})\cdot H(e^{j\omega})$
  - log $\rightarrow$ sum: $\log G(e^{j\omega}) + \log H(e^{j\omega})$
  - IFT
    $\rightarrow$ separate fine structure: $c_g[n] + c_h[n]$
    = *deconvolution*

- **Definition:**

  Real cepstrum $c_n = \text{idft}(\log|\text{dft}(x[n])|)$

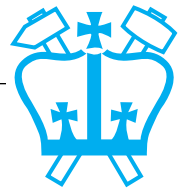
Stages in cepstral deconvolution

- Original waveform has excitation fine structure convolved with resonances
- DFT shows harmonics modulated by resonances
- Log DFT is *sum* of harmonic 'comb' and resonant bumps
- IDFT separates out resonant bumps (low quefrency) and regular, fine structure ('pitch pulse')
- Selecting low-n cepstrum separates resonance information (deconvolution / 'liftering')



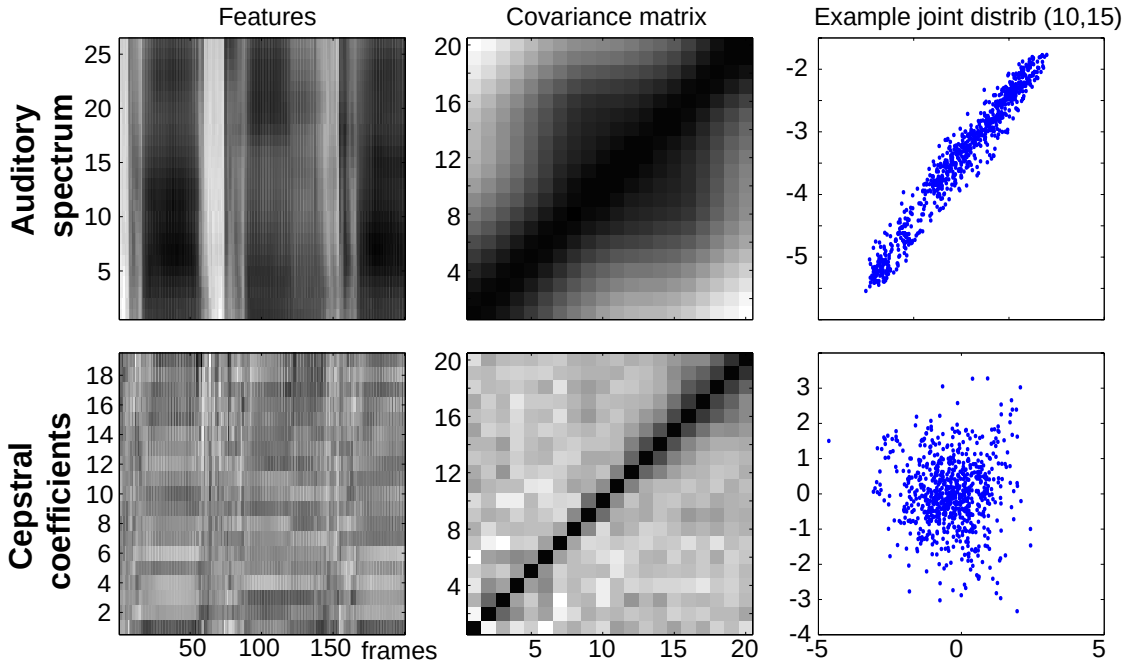
Properties of the cepstrum

- **Separate source (fine) & filter (broad structure)**
  - smooth the log mag spectrum to get resonances
- ***Smoothing* spectrum is *filtering* along freq.**
  - i.e. convolution applied in Fourier domain
    $\rightarrow$ *multiplication* in IFT ('liftering')
- **Periodicity in time $\rightarrow$ harmonics in spectrum**
  $\rightarrow$ 'pitch pulse' in high-n cepstrum



Aside: Correlation of elements

- **Cepstrum is a popular in speech recognition**
  - feature vector elements are *decorrelated*:

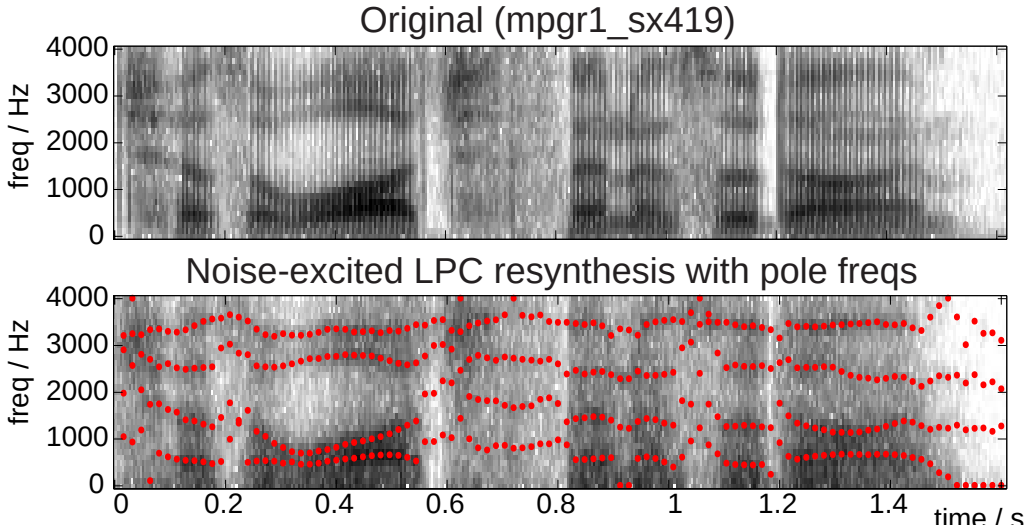


  - c_0 'normalizes out' average log energy

- **Decorrelated pdfs fit diagonal Gaussians**
  - simple correlation is a waste of parameters

Aside: Formant tracking

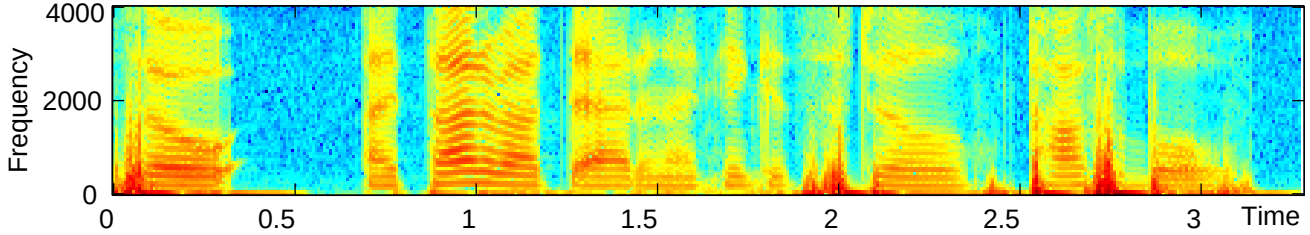
- **Formants carry (most?) linguistic information**
- **Why not classify → speech recognition ?**
  - e.g. local maxima in cepstral-liftered spectrum
- **But: recognition needs to work in *all* circumstances**
  - formants can be obscure or undefined



→ **Need more graceful, robust parameters**

1 Recognizing Speech

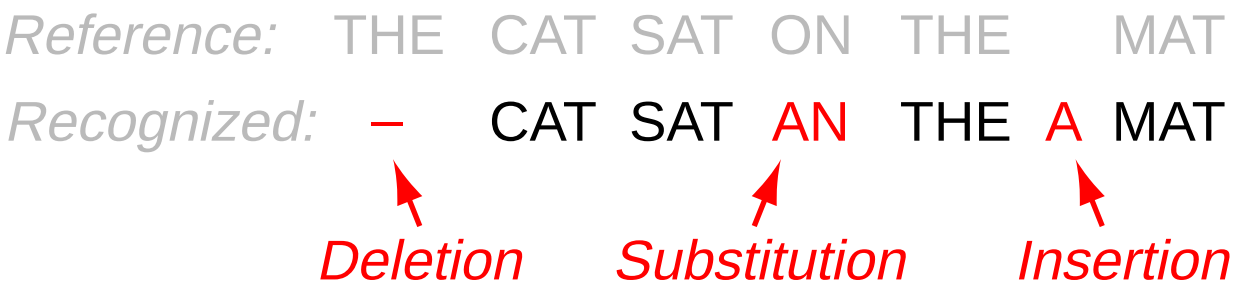
"So, I thought about that and I think it's still possible"



- **What kind of information might we want from the speech signal?**
  - words
  - phrasing, 'speech acts' (prosody)
  - mood / emotion
  - speaker identity

Speech recognition as Transcription

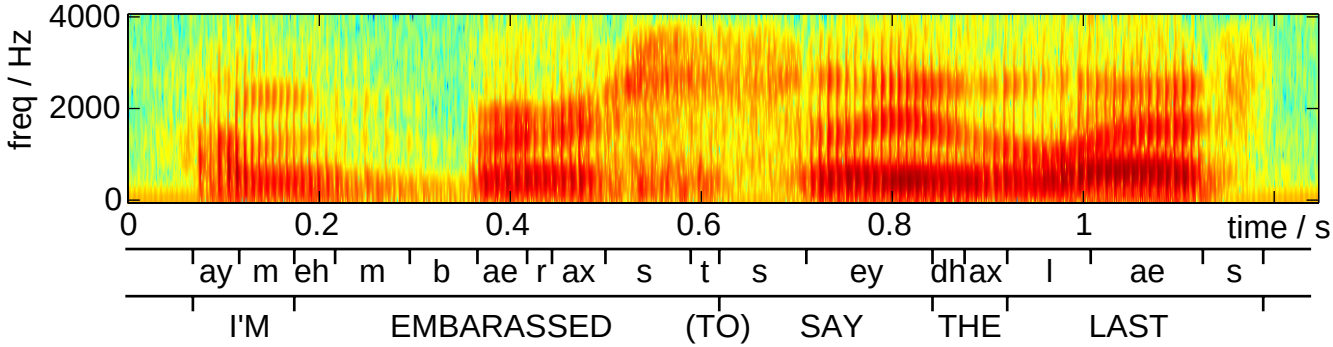
- **Transcription = "speech to text"**
  - find a word string to match the utterance
- **Best suited to small vocabulary tasks**
  - voice dialing, command & control etc.
- **Gives neat objective measure: word error rate (WER) %**
  - can be a sensitive measure of performance
- **Three kinds of errors:**



  - WER = (S + D + I) / N

Limitations of the Transcription paradigm

- **Starts to fall down with 'natural' speech**
  - some "words" may not even exist



- **Word transcripts do not capture everything**
  - speaker changes, intonation, phrasing
- **Word error rate treats all errors as equal**
  - small words ("of") counted as big words
  - small differences ("company's" → "companies") vs. larger ("held police" → "health plans")
- **Move towards other measures**
  - e.g. task-defined: was the *meaning* recognized?

Why is Speech Recognition hard?

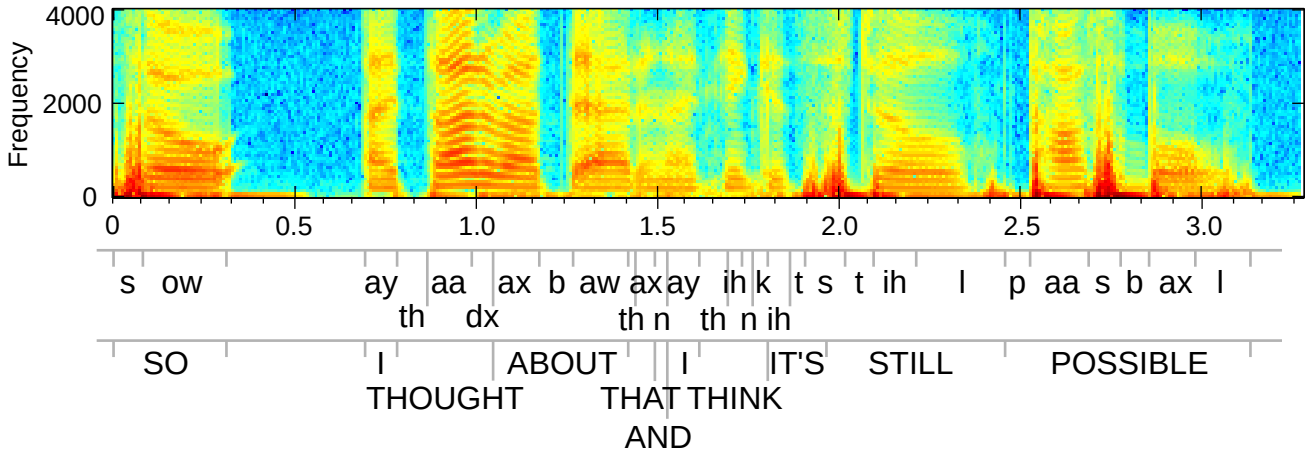
- **Why not just match against a set of waveforms?**
  - waveforms are never (nearly!) the same twice

- **Speech *variability* comes from various sources:**
  - speaker-dependent (SD) recognizers must handle within-speaker variability
  - speaker-independent (SI) recognizers must also deal with variation between speakers
  - all recognizers are afflicted by background noise, variable channels

→ **Need recognition models that:**
  - *generalize* i.e. accept variations in a range, and

Within-speaker variability

- **Timing variation:**
  - word duration varies enormously



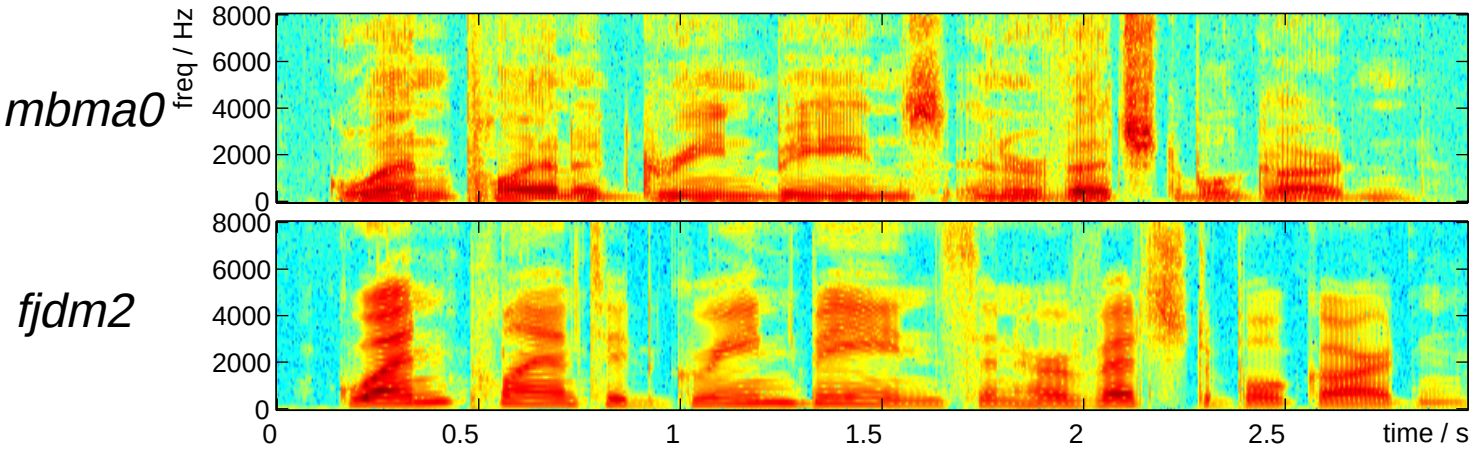
  - fast speech ‘reduces’ vowels

- **Speaking style variation:**
  - careful/casual articulation
  - soft/loud speech

- **Contextual effects:**
  - speech sounds vary with context, role:
    “How **do** you **do**?”

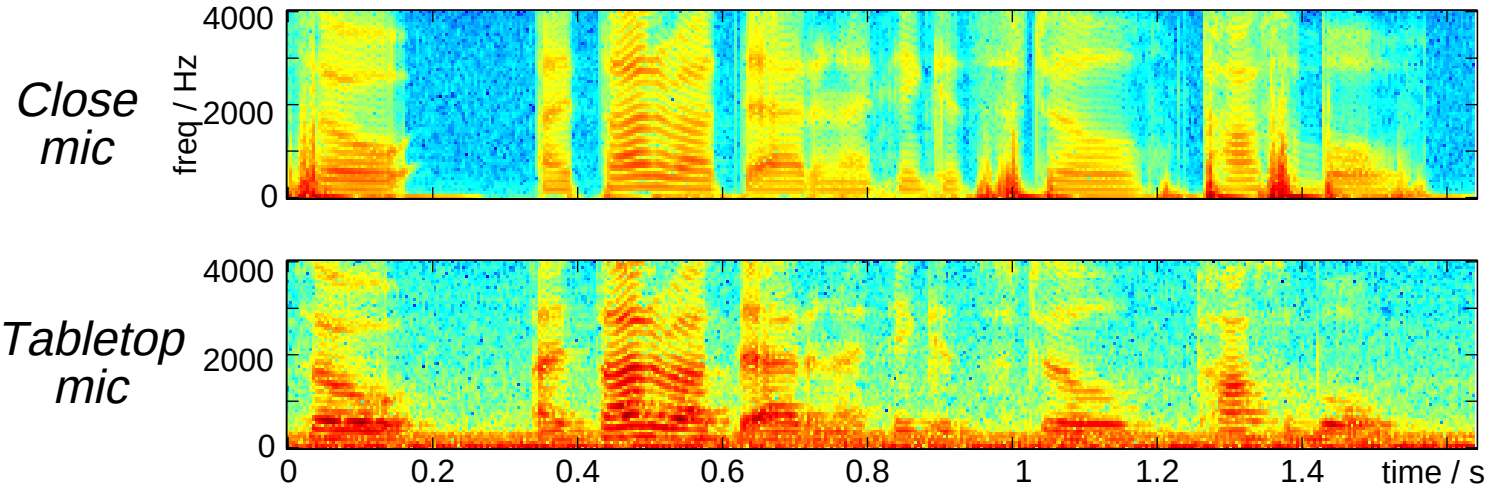
Between-speaker variability

- **Accent variation**
  - regional / mother tongue
- **Voice quality variation**
  - gender, age, huskiness, nasality
- **Individual characteristics**
  - mannerisms, speed, prosody



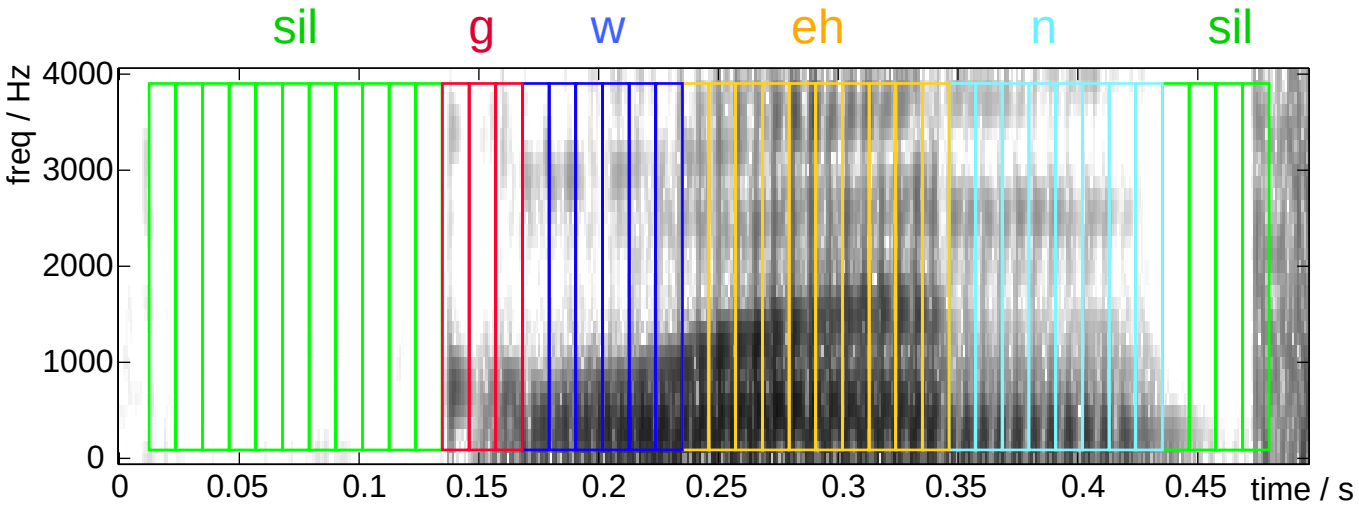
Environment variability

- **Background noise**
  - fans, cars, doors, papers
- **Reverberation**
  - 'boxiness' in recordings
- **Microphone channel**
  - huge effect on relative spectral gain



How to recognize speech?

- **Cross correlate templates?**
  - waveform?
  - spectrogram?
  - *time-warp* problems
- **Match short-segments & handle time-warp later**
  - model with slices of ~ 10 ms
  - pseudo-stationary model of words:



  - other sources of variation...

Which segments to use?

- **Assume words can be broken down into pseudo-stationary segments**
  - not a perfect fit, but worth a try
- **Linguists offer phonemes or phones**
  - phonemes are the minimal set needed to disambiguate words
  - phones are realizations of phonemes
- **Other possibilities:**
  - data-clustering techniques to define segments 'intrinsically'
  - lesson from synthesis: transitions as important or more important than steady portions?

    ...but how to model?

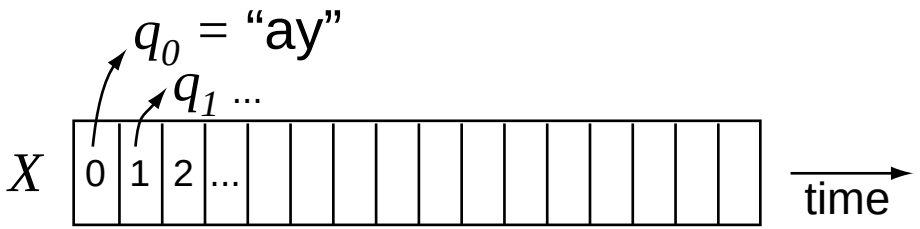
Probabilistic formulation

- ***Probability* that segment label is correct**
  - gives standard form of speech recognizers:

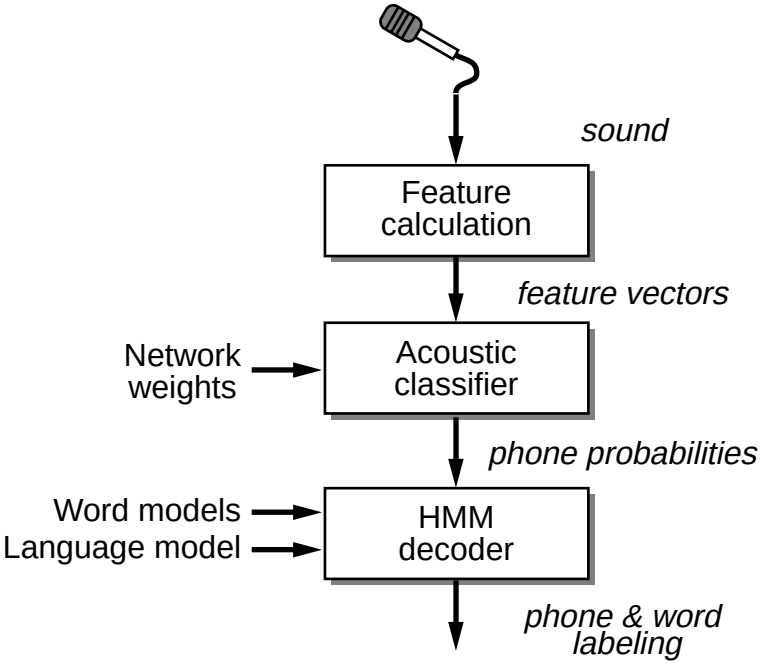
- **Feature calculation**
  transforms signal into easily-classified domain
  $$s[n] \rightarrow X_m \quad \left(m = \frac{n}{H}\right)$$

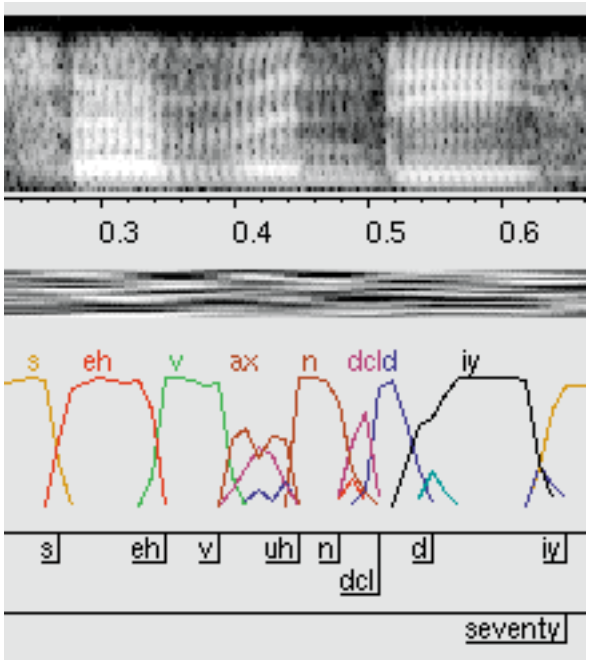
- **Acoustic classifier**
  calculates probabilities of each mutually-exclusive state q^i
  $$p(q^i | X)$$

- **'Finite state acceptor' (i.e. HMM)**
  $$\hat{Q} = \underset{\{q_0, q_1, \ldots q_L\}}{\operatorname{argmax}} p(q_0, q_1, \ldots q_L | X_0, X_1 \ldots X_L)$$
  MAP match of allowable sequence to probabilities:



Standard speech recognizer structure





- **Questions:**
  - what are the best features?
  - how do we do the acoustic classification?
  - how do we find/match the state sequence?

2 Feature Calculation

- **Goal: Find a representational space within which to apply classification**
  - waveform: voluminous, redundant, variable
  - spectrogram: better, still quite variable
  - ...?
- **Pattern Recognition: Representation is upper bound on performance**
  - maybe we *should* use the waveform...
  - or, maybe the representation can do *all* the work
- **Feature calculation is intimately bound to classifier**
  - pragmatic strengths and weaknesses
- **Features develop by slow evolution**
  - current choices more historical than principled

Desired characteristics for features

- **Provide the 'right' information**
  - extract signal information for classification task
  - suppress irrelevant information
- **Be compatible with acoustic classifier**
  - relatively low dimensionality
  - uncorrelated dimensions?
- **Be practical**
  - applicable in 'all' circumstances
  - relatively inexpensive to compute
- **Be robust**
  - so far as possible, exclude nonspeech information

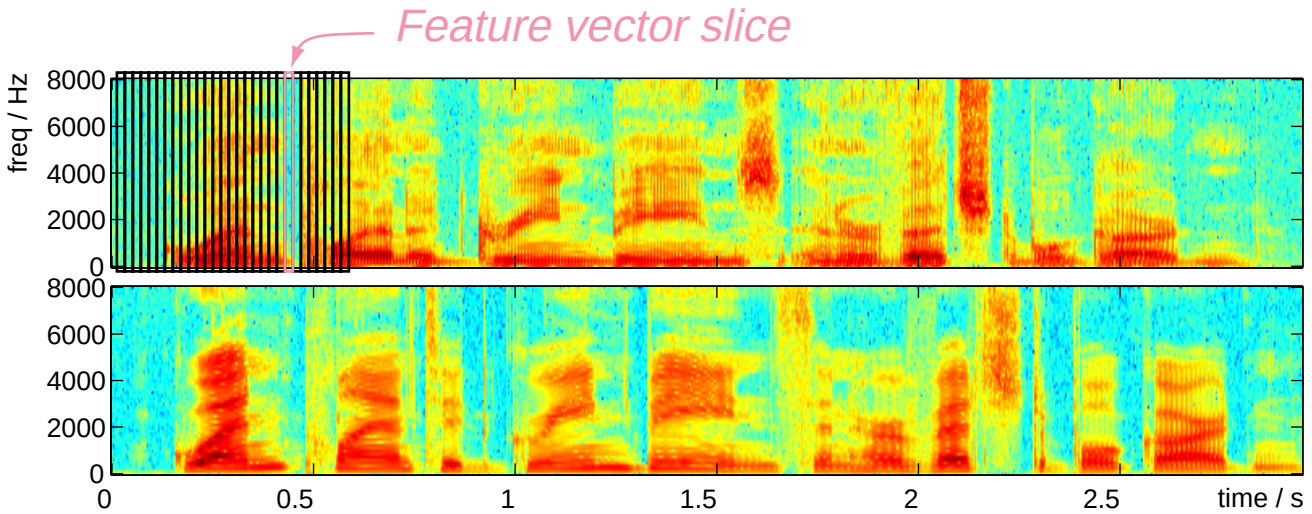
→ **How to evaluate features?**
  - normally: just put them in a recognizer

Features (1): Spectrogram

- **Plain STFT as features e.g.**

$$X_m[k] = S[mH, k] = \sum_n s[n + mH] \cdot w[n] \cdot e^{-(j2\pi kn)/N}$$

- **Consider examples:**



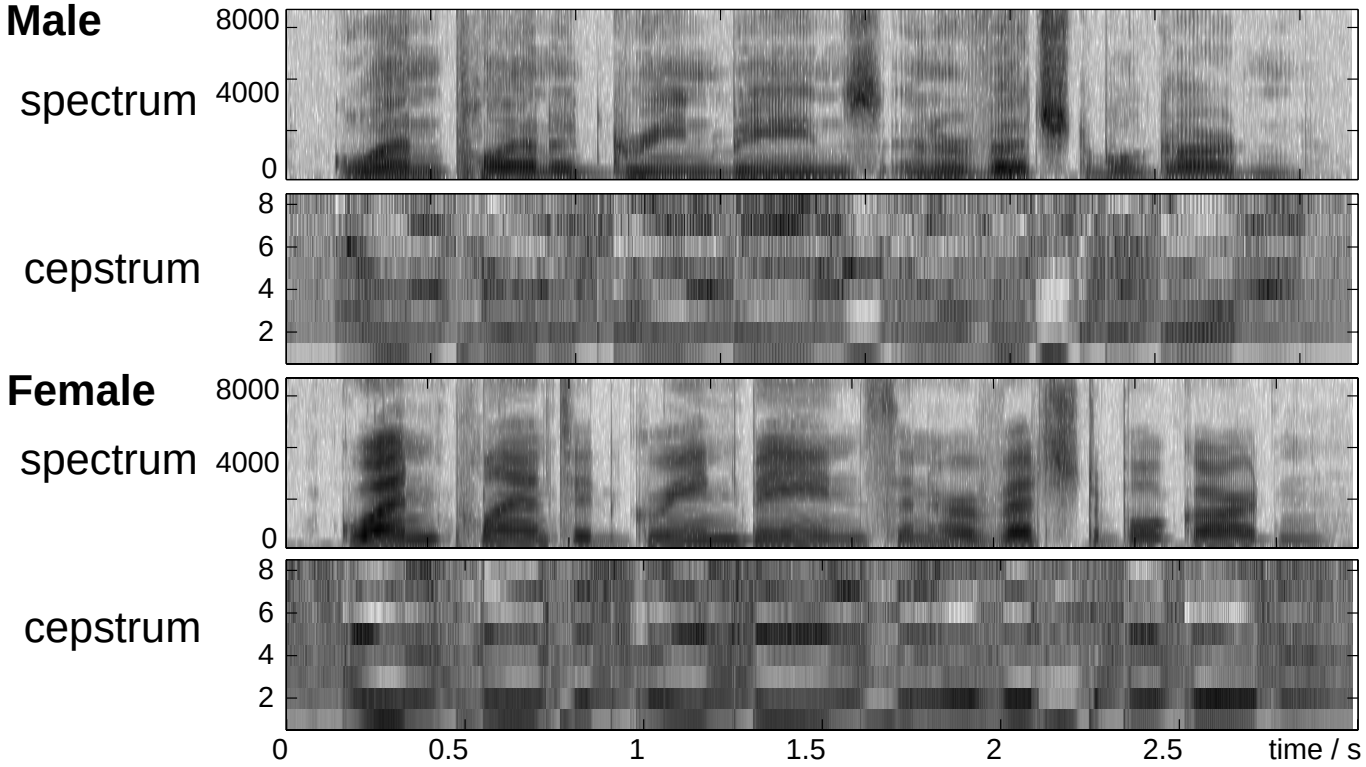
- **Similarities between corresponding segments**
  - but still large differences

Features (2): Cepstrum

- **Idea: Decorrelate, summarize spectral slices:**

$$X_m[l] = IDFT\{\log|S[mH, k]|\}$$

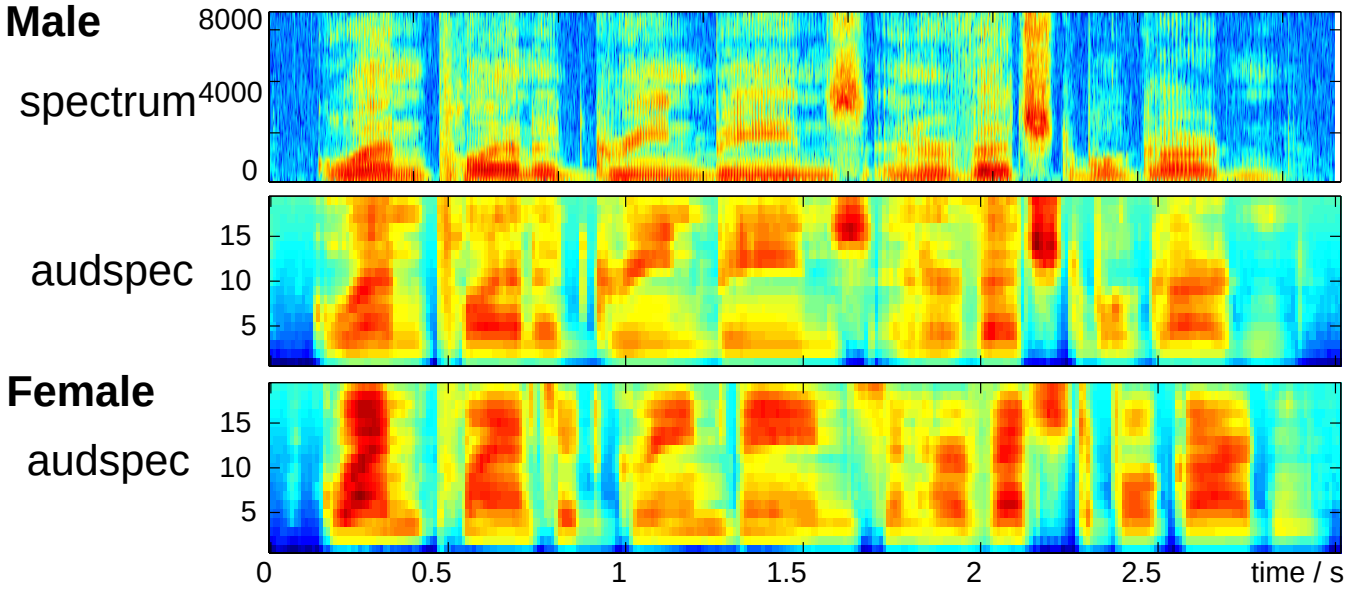
  - good for Gaussian models
  - greatly reduce feature dimension



Features (3): Frequency axis warp

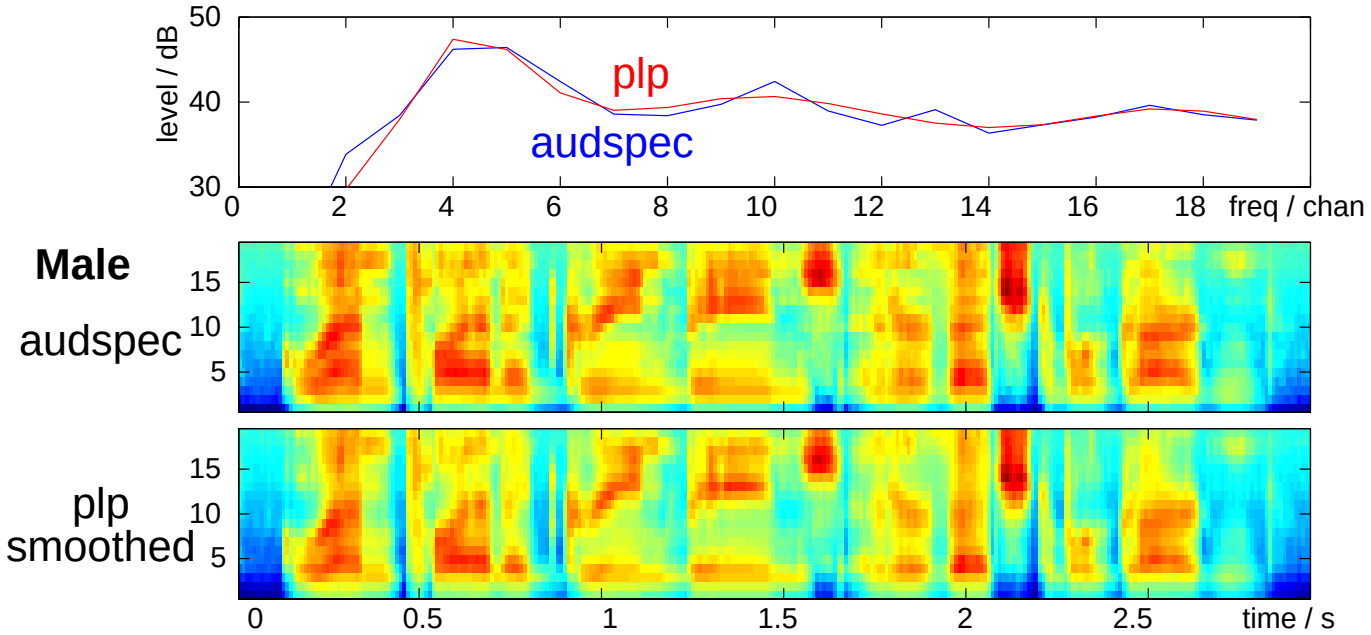
- **Linear frequency axis gives equal 'space' to 0-1 kHz and 3-4 kHz**
  - but relative perceptual importance is tiny
- **Warp frequency axis closer to perceptual axis:**
  - mel, Bark, constant-Q ...

$$X[c] = \sum_{k=l_c}^{u_c} |S[k]|^2$$



Features (4): Spectral smoothing

- **Generalizing across different speakers is helped by *smoothing* (blurring) spectrum**
- **Truncated cepstrum is one way:**
  - MSE approx to $\log|S[k]|$
- **LPC modeling is a little different:**
  - MSE approx to $|S[k]| \rightarrow$ prefers detail at peaks



Features (5): Normalization along time

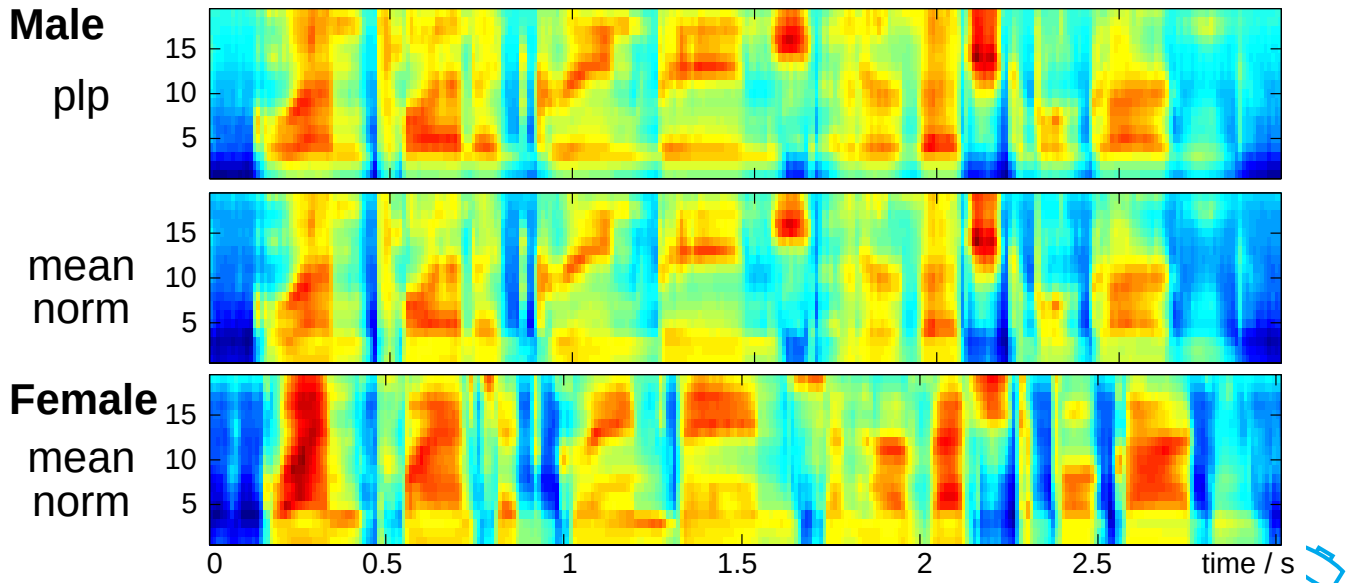
- **Idea: feature *variations*, not absolute level**
- **Hence: calculate average level & subtract it:**

$$X[k] = S[k] - \text{mean}\{S[k]\}$$

- **Factors out fixed channel frequency response:**

$$s[n] = h[n] * e[n]$$

$$\log|S[k]| = \log|H[k]| + \log|E[k]|$$

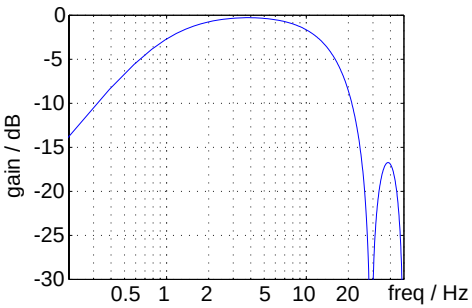


Features (6): RASTA filtering

- **Mean subtraction ≈ *high-pass filtering* along time in log-spectral domain**

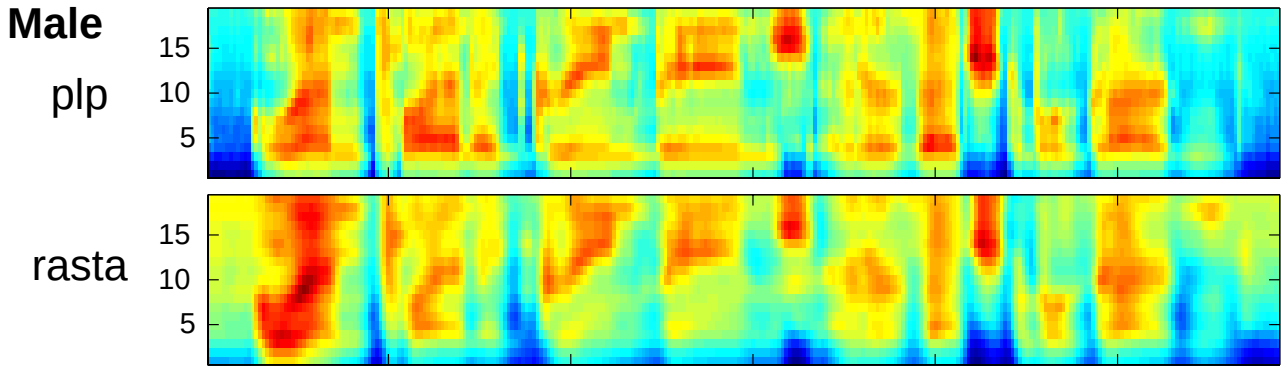
$$X[k] = S[k] - \mathbf{lpf}\{S[k]\}$$

- **+ *smooth* along time for more blurring**



→ **Bandpass filter in time**

- relates to 'modulation sensitivity' in hearing?

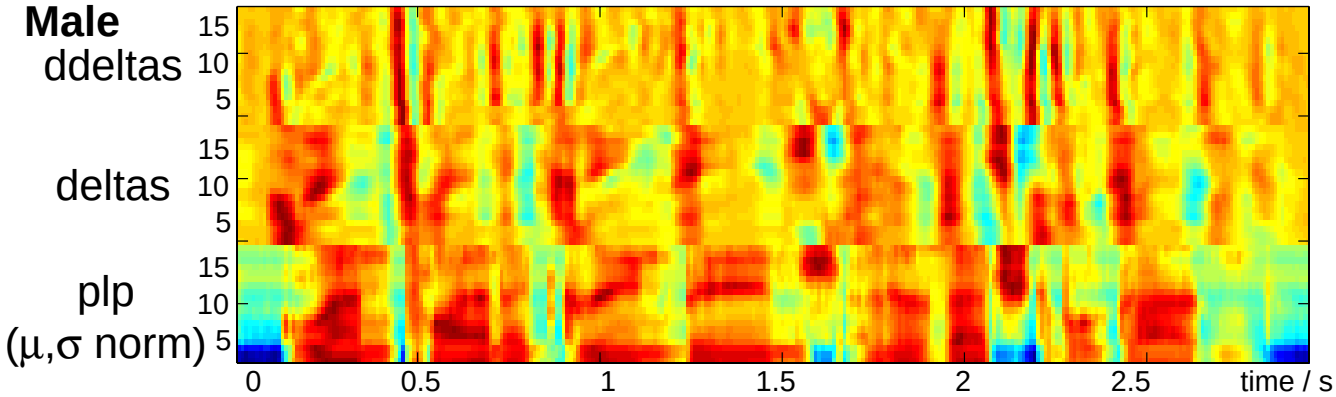


Delta features

- **Want each segment to have 'static' feature vals**
  - but some segments intrinsically dynamic!

  →calculate their derivatives - maybe steadier?

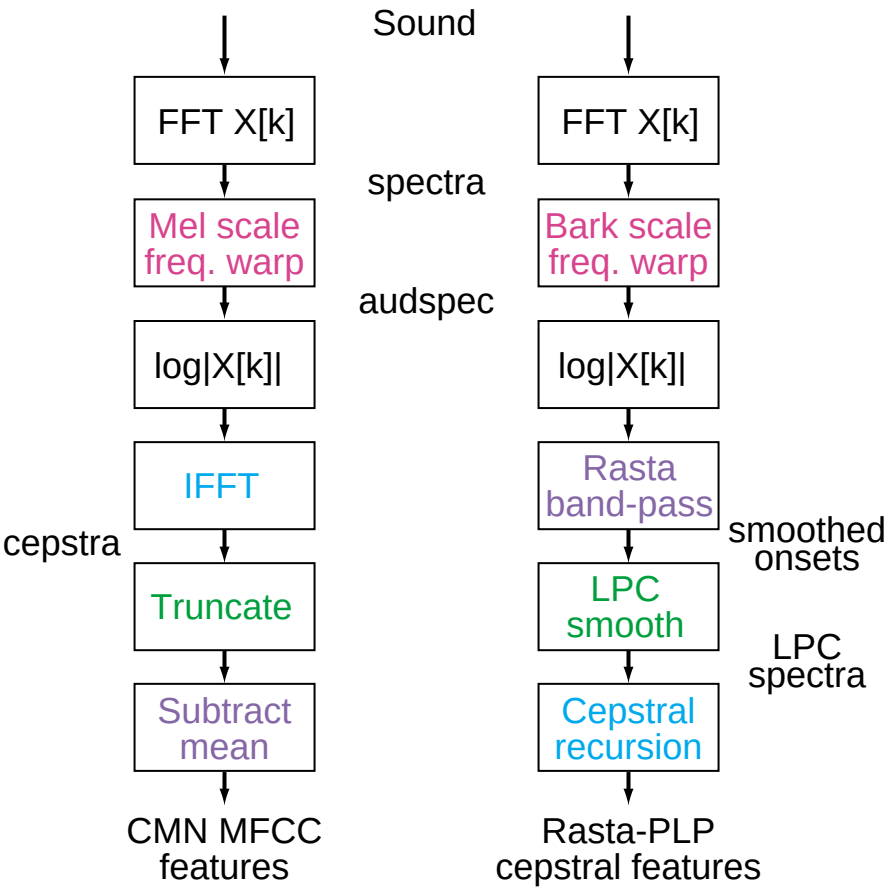
- **Append dX/dt (+ d^2X/dt^2) to feature vectors**



- **Relates to onset sensitivity in humans?**

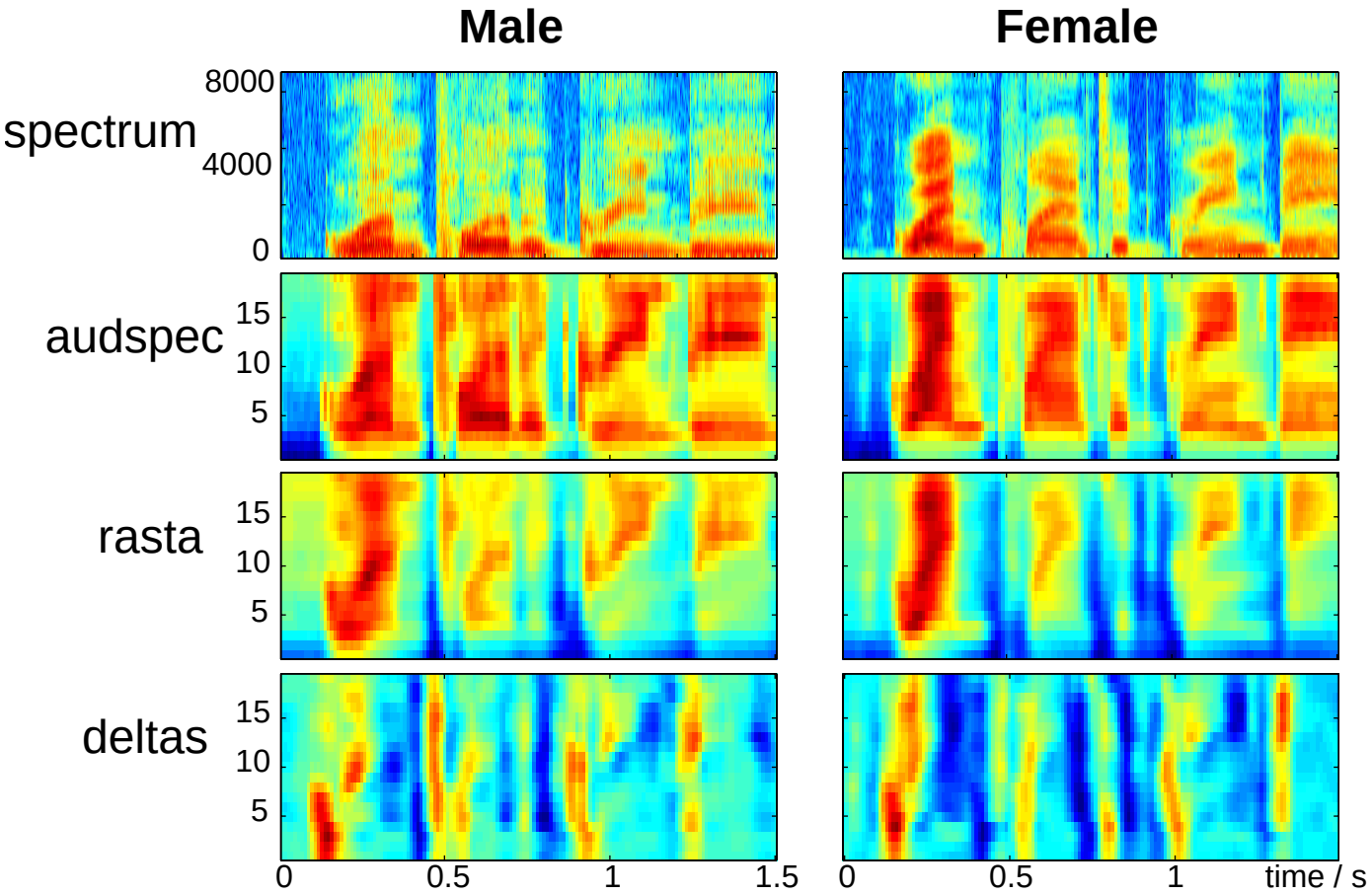
Overall feature calculation

- **MFCCs and/or RASTA-PLP**



- **Key attributes:**
  - spectral, auditory scale
  - decorrelation
  - smoothed (spectral) detail
  - normalization of levels

Features summary



- **Normalize same phones**
- **Contrast different phones**

3 Acoustic Classification

- **Goal: Convert features into probabilities of particular labels:**

  i.e find $p(q_n^i | X_n)$ **over some state set {**q^i**}**

  - conventional statistical classification problem

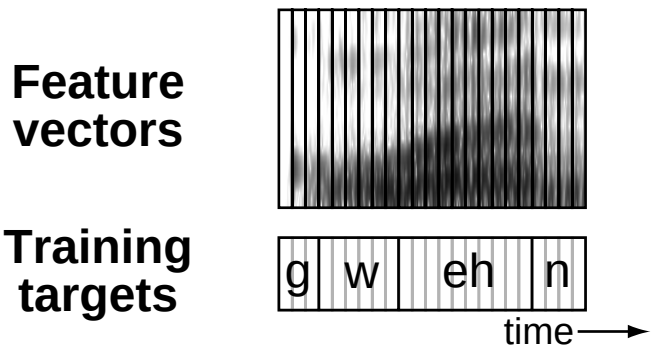
- **Classifier construction is *data-driven***
  - assume we can get examples of known good Xs for each of the q^is
  - calculate model parameters by standard training scheme

- **Various classifiers can be used**
  - GMMs model distribution under each state
  - Neural Nets directly estimate posteriors

- **Different classifiers have different properties**
  - features, labels limit ultimate performance

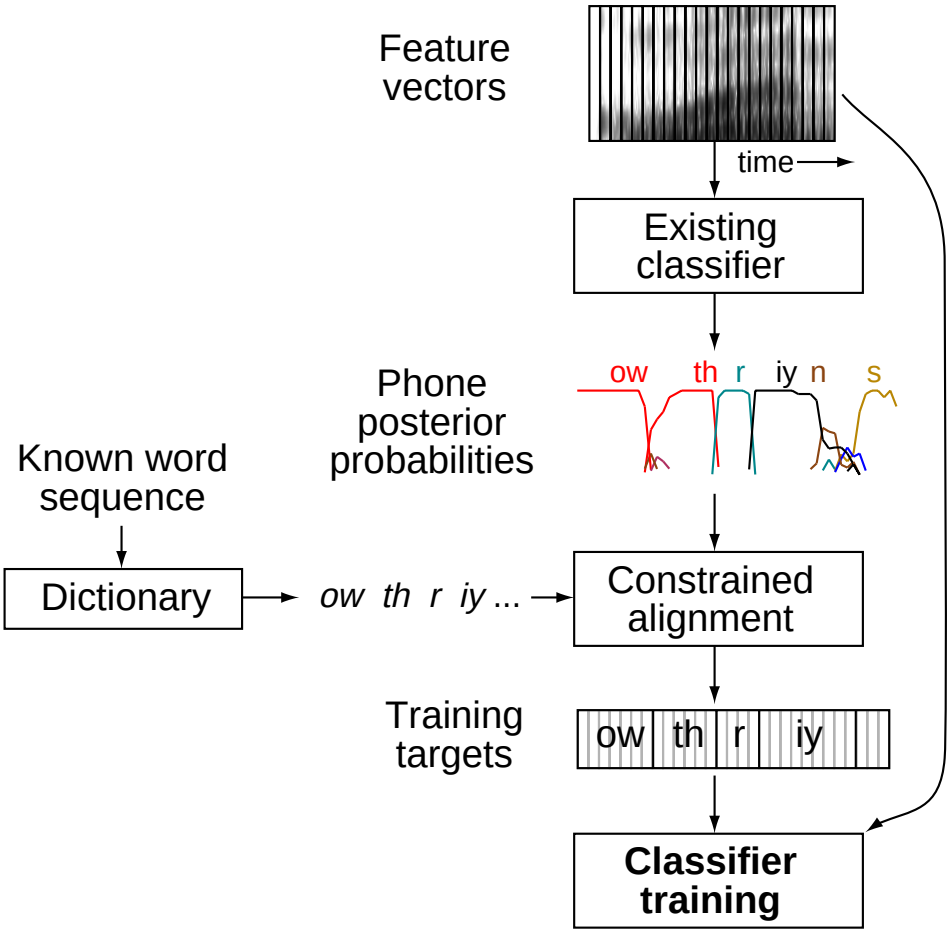
Defining classifier targets

- **Choice of {q^i} can make a big difference**
  - must support recognition task
  - must be a practical classification task
- **Hand-labeling is one source...**
  - 'experts' mark spectrogram boundaries
- **...Forced alignment is another**
  - 'best guess' with existing classifiers, given words
- **Result is *targets* for each training frame:**



Forced alignment

- **Best labeling given existing classifier constrained by known word sequence**



Gaussian Mixture Models: Principles

- **Fit the distribution of features under states:**
  - *separate* model for each state q^i

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

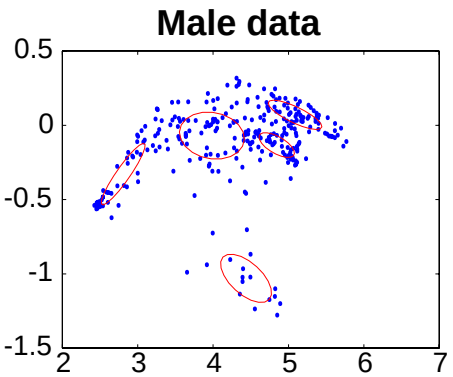
- **GMMs can match any distribution given enough data**
  - even if we assume diagonal covariance
  - but: curse of dimensionality
- **GMMs produce 'likelihoods'; can convert to posteriors via Bayes' rule:**

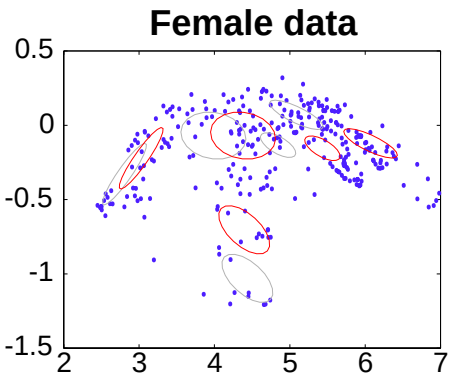
$$p(q^i|\mathbf{x}) = \frac{p(\mathbf{x}|q^i) \cdot Pr(q^i)}{\sum_j p(\mathbf{x}|q^j) \cdot Pr(q^j)}$$

GMMs: Practicalities

- **Practical GMMs:**
  - 9 to 39 dimensions
  - 2 to 64 Gaussians per mixture
    depending on number of training examples
- **Lots of data → can model more classes**
  - e.g context-independent (CI): q^i = ***ae*** ***aa*** ***ax*** ...

  →context-dependent (CD): q^i = *b*-***ae***-*b* *b*-***ae***-*k* ...
- **Explicit parameters**
  → opportunities for *adaptation*?





Summary

- **Speech recognition as *word transcription***
  - neat definition, but limited
  - hard because of variability
- **Feature calculation extracts information**
  - smoothed, decorrelated spectral parameters
  - long evolution to match classifiers
- **Acoustic classifier estimates phone class**
  - Training data + labels train classifier
  - GMMs model each class's distribution
  - Neural nets discriminatively estimate posteriors