How to Wreck A Nice Beach

Eric Fosler-Lussier

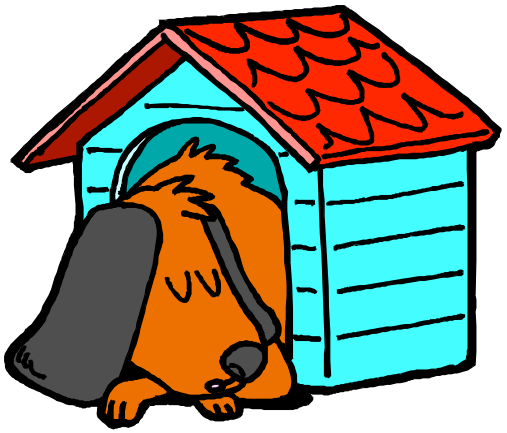
The Ohio State University

A brief history of speech recognition

- Radio Rex: The first commercial speech recognizer
  - What's going on in the speech signal?
  - Thinking about speech as a *sequence*
- A first model: template based speech recognition
- Enter the probability: HMMs for speech recognition
  - Parts of a standard speech recognizer
  - Gaussian vs neural-network acoustic models
- Recent techniques
  - Sequence-level training criteria
  - "End-to-end" speech recognition
  - Handling background noise

The First Commercial Speech Recognizer

- Radio Rex: Toy based on speech recognition (1922)



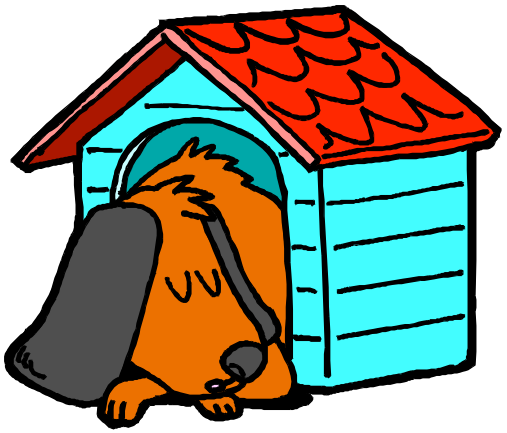
- Rex jumps out of his house when someone says “Rex!”
- How did this work? (DEMO)

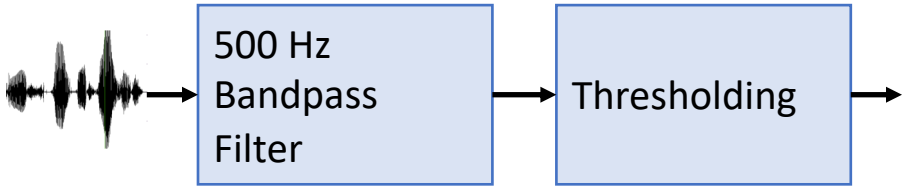
http://www.youtube.com/watch?v=AdUi_St-BdM&t=0m30s

Excerpt 1: 0:30
Excerpt 2: 2:55

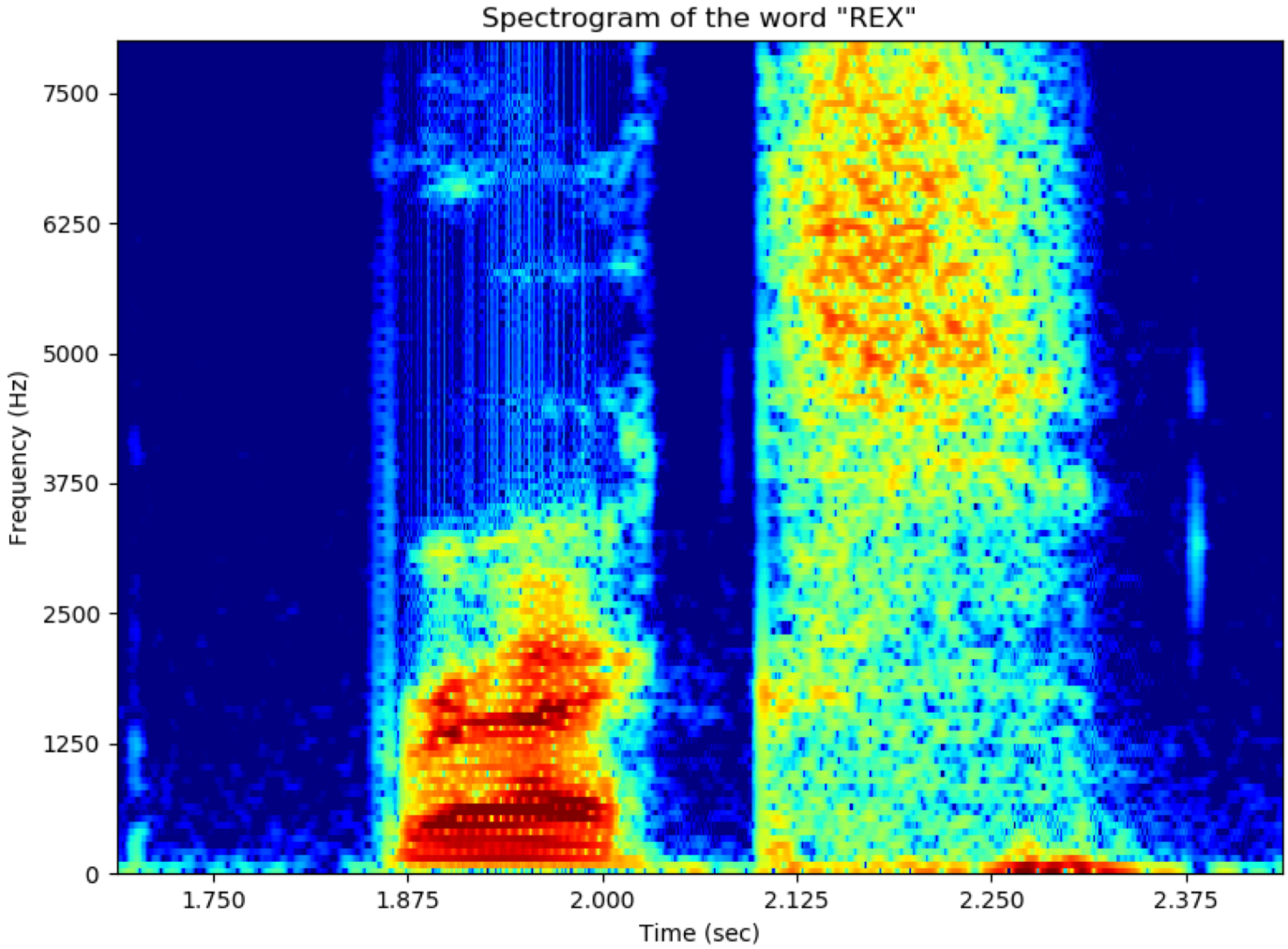
The First Commercial Speech Recognizer

- Radio Rex: Toy based on speech recognition (1922)



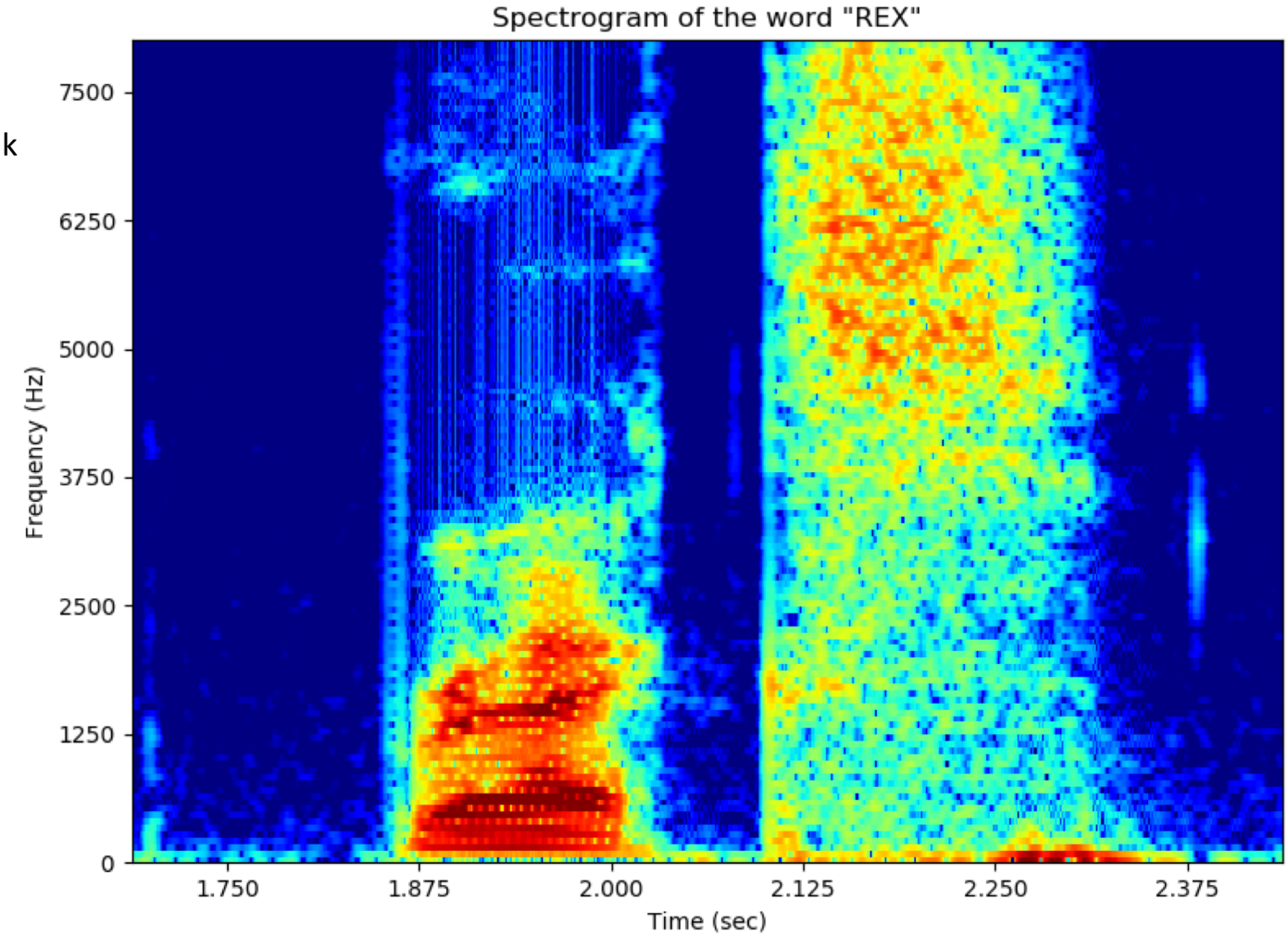


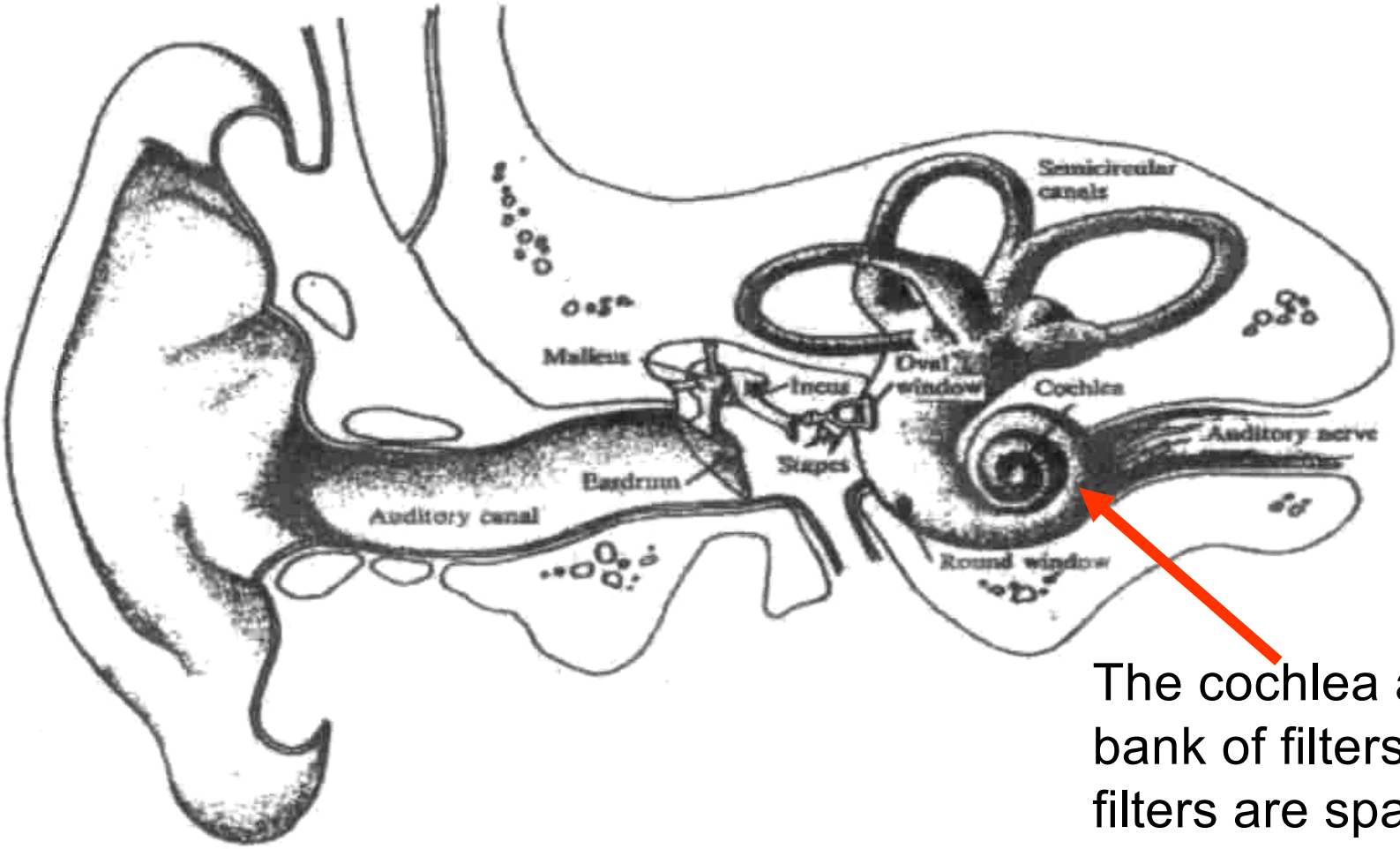
- Rex jumps out of his house when someone says “Rex!”
- Resonator picks up energy around 500Hz, releases electromagnet spring when there is enough vibration.


Spectrogram of the word "REX"
Frequency (Hz)
7500
6250
5000
3750
2500
1250
0
1.750
1.875
2.000
2.125
2.250
2.375
Time (sec)

Why do we break things down by frequency?

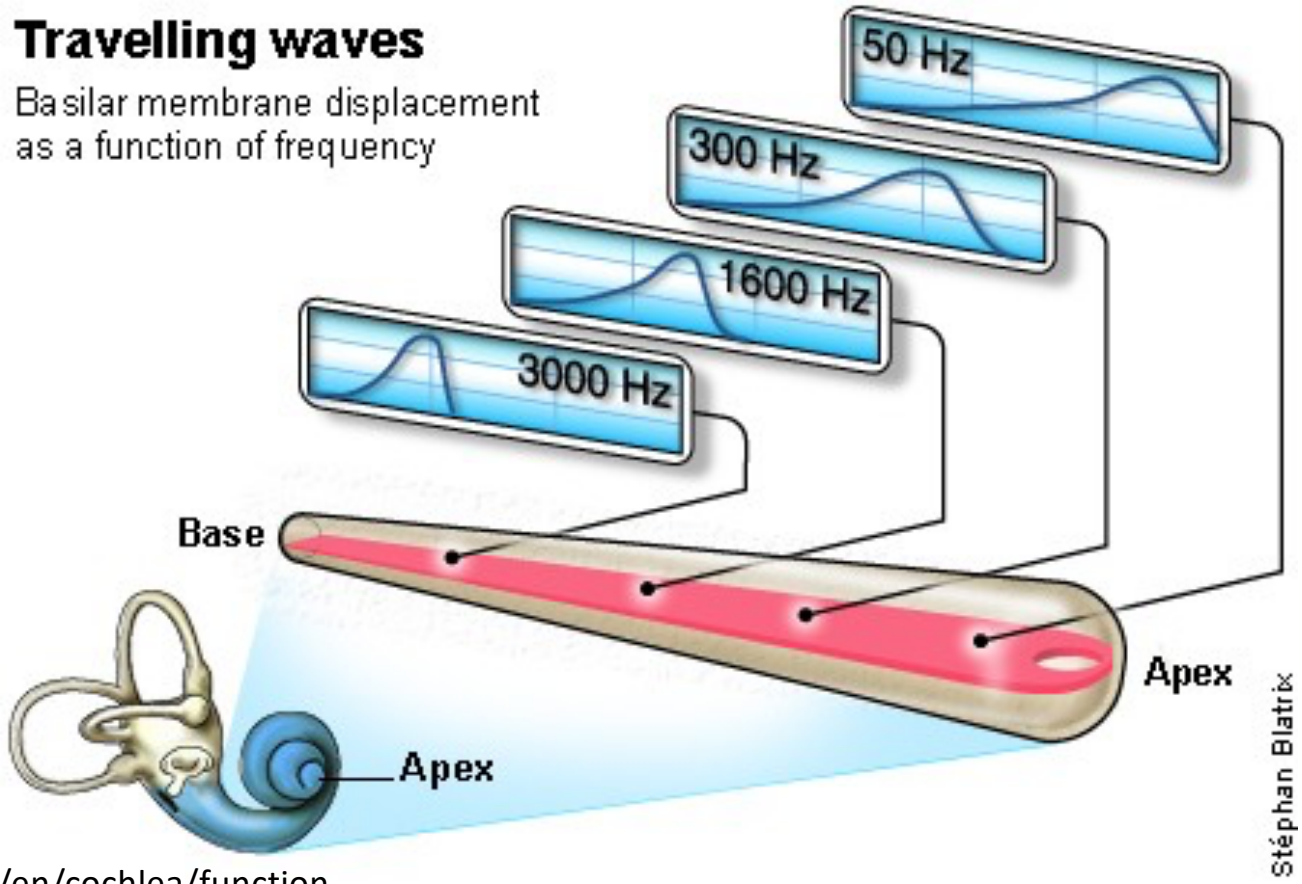
What do the patterns tell us?





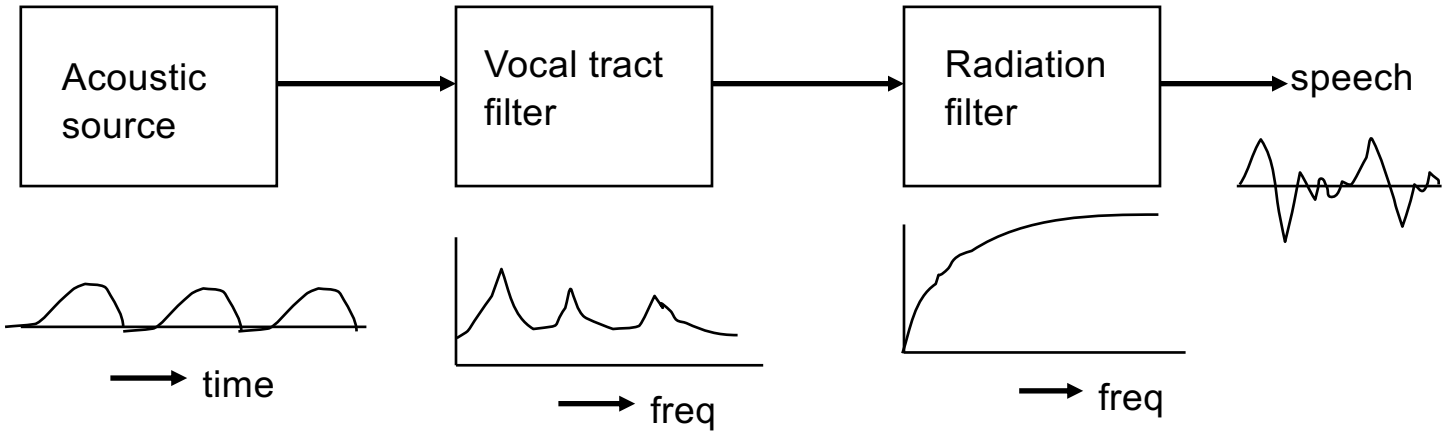
The cochlea acts as a bank of filters. The filters are spaced on a non-linear scale.

The cochlea separates out sound frequencies



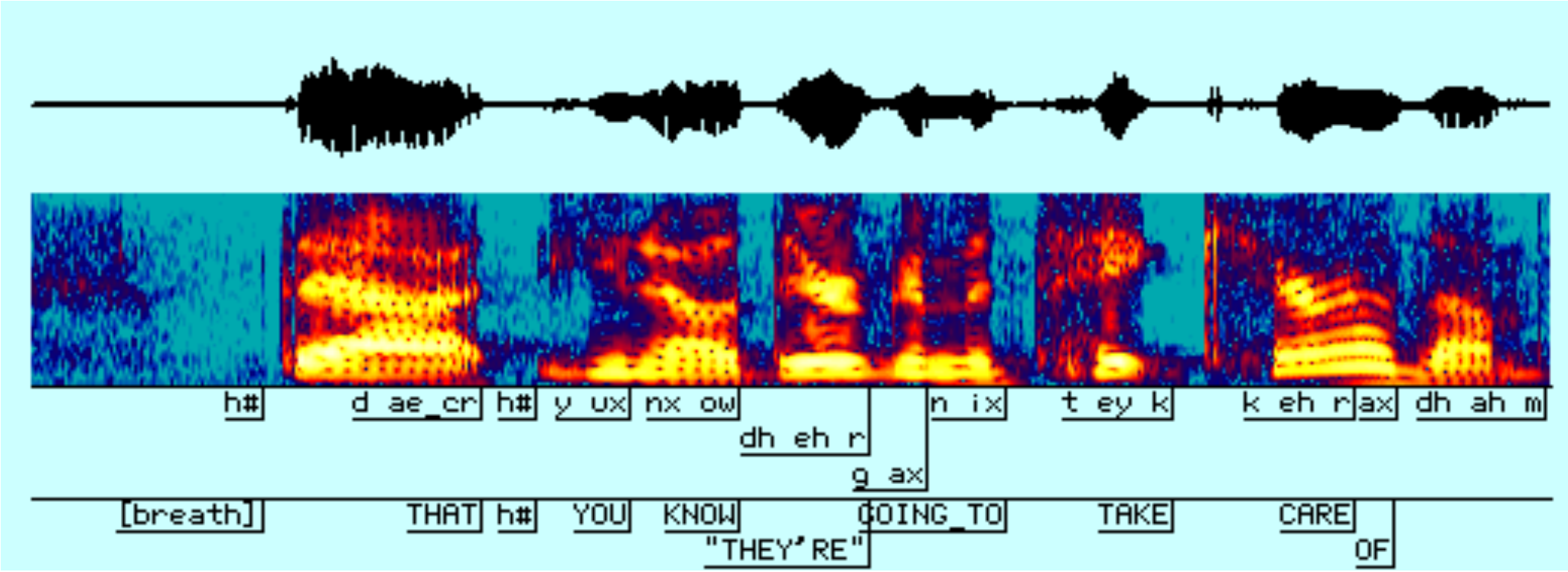
http://www.cochlea.eu/en/cochlea/function

Source Filter Model for Speech



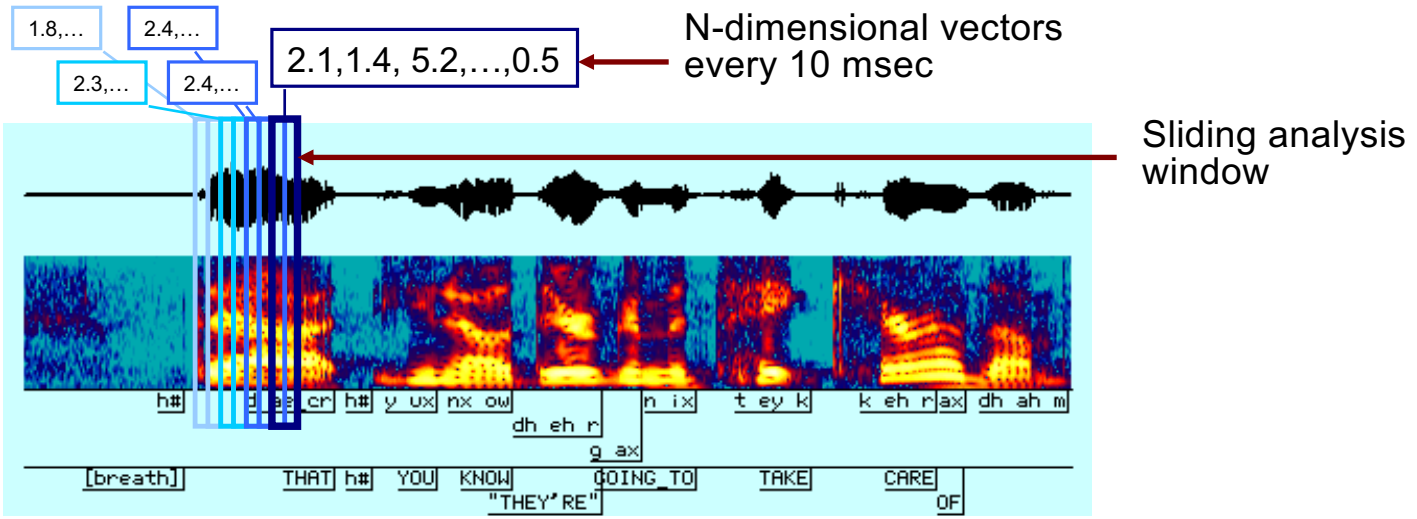
- Source:
  - Vocal cord vibration
  - Noise due to turbulence
- VT Filter:
  - Vocal tract from glottis to mouth
  - Nasal tract
- Radiation:
  - Filter converting volume velocity to pressure at lips

Can you spot the different sounds of English?



Sounds of a language can be either *phones* or *phonemes* – distinction in linguistics, often used interchangeably in speech tech.

Speech input: frequency vectors over time



- How big is N? (Depends on representation, but usually 30-40.)
- Typically use information about how the ear works to bin frequencies together.
- Sometimes decorrelate frequencies by taking cepstral transform.

Speech as a **sequence**

- What we have: input acoustic sequence **X**
  - 40-dimensional vectors, 100 frames/second = 4000 floats per second
  - (Could go from waveform directly: 44100 ints/sec at CD quality.)
- What we want: output word sequence **W**
  - Typical speaking rate: 4-5 syllables/sec == 2-3 words/sec

- In general, want a function ASR(**X**) → **W**
  - ASR = Automatic Speech Recognition
  - Notice the mismatch in number of frames to words: can't treat this as a straight labeling problem
  - Different approaches have evolved through the years

Template matching (early cell phones)

- Look at two spectral slices as vectors that can be compared
  - We can calculate a distance between any two slices (Euclidean, cosine…)
  - Similar phones from the same speaker in the same context often look the same → distances between them should be small
    - Also: assume that most sounds are steady-state across frames (not always true!)
- If two words are the same they *should* have the same sequence of sounds. (Also not true all of the time!)
- Dynamic Time Warping is a way to compute the best alignment between any two sequences.
  - For any input sequence **X**, match against template sequences $\mathbf{T_1 \ldots T_k}$
  - Return the best scoring template: ASR(**X**) = argmin$\mathbf{T_i}$ (DTW(**X**, $\mathbf{T_i}$))

Dynamic Time Warping

- Basic ideas of alignment of signal X to template T:
  - Every point in X must align to some point in T
  - Every point in T must align to some point in X
  - You need to align in signal order

- Pseudocode

```
for i ranging from 1 to m=size(X)
  for j ranging from 1 to n=size(T)
        B(i,j)=argmin(D(i-1,j),D(i-1,j-1),D(i,j-1))
    D(i,j)=min(D(i-1,j),D(i-1,j-1),D(i,j-1))+dist(X(i),Y(j))
return B,D
```

(often just care about the score, not the backtrace array)

Example

- Align signal X to template T

  X: 1 2 3 4

  T: 1 2 2 4 2 3 3

- Distance: $|x_i - t_j|$

Option 1:

X	T
1	1
2	2
3	2
4	4233

Cost: 1+2+1+1 = 7

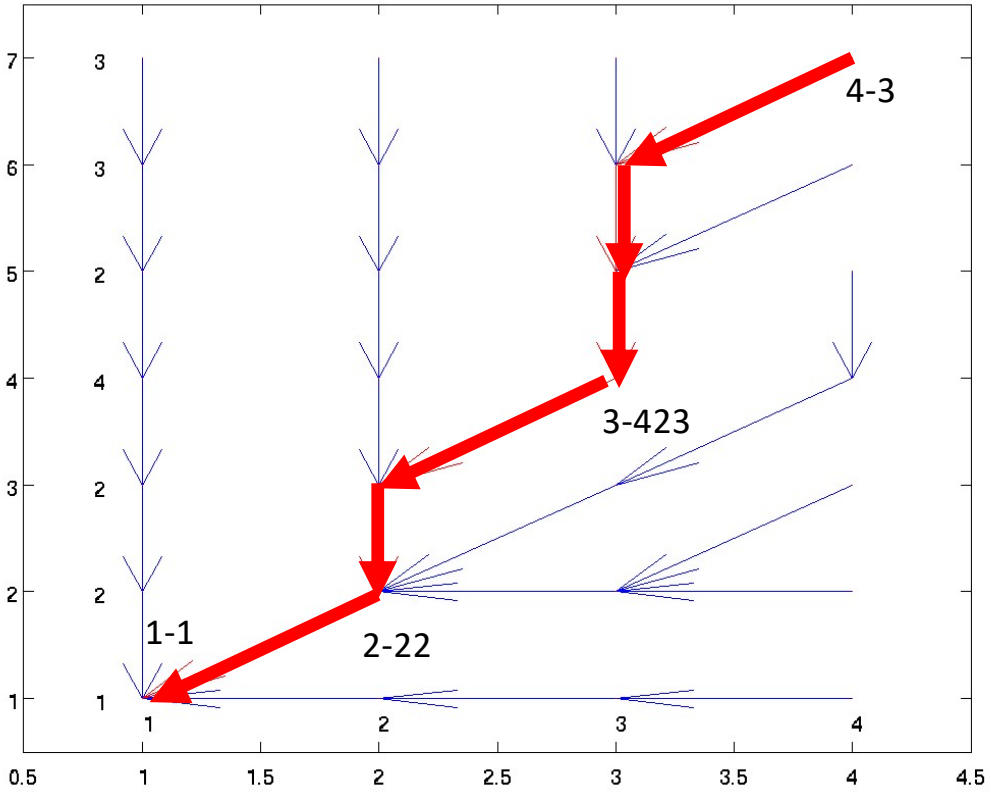
Option 2:

X	T
1	1
2	22
3	423
4	3

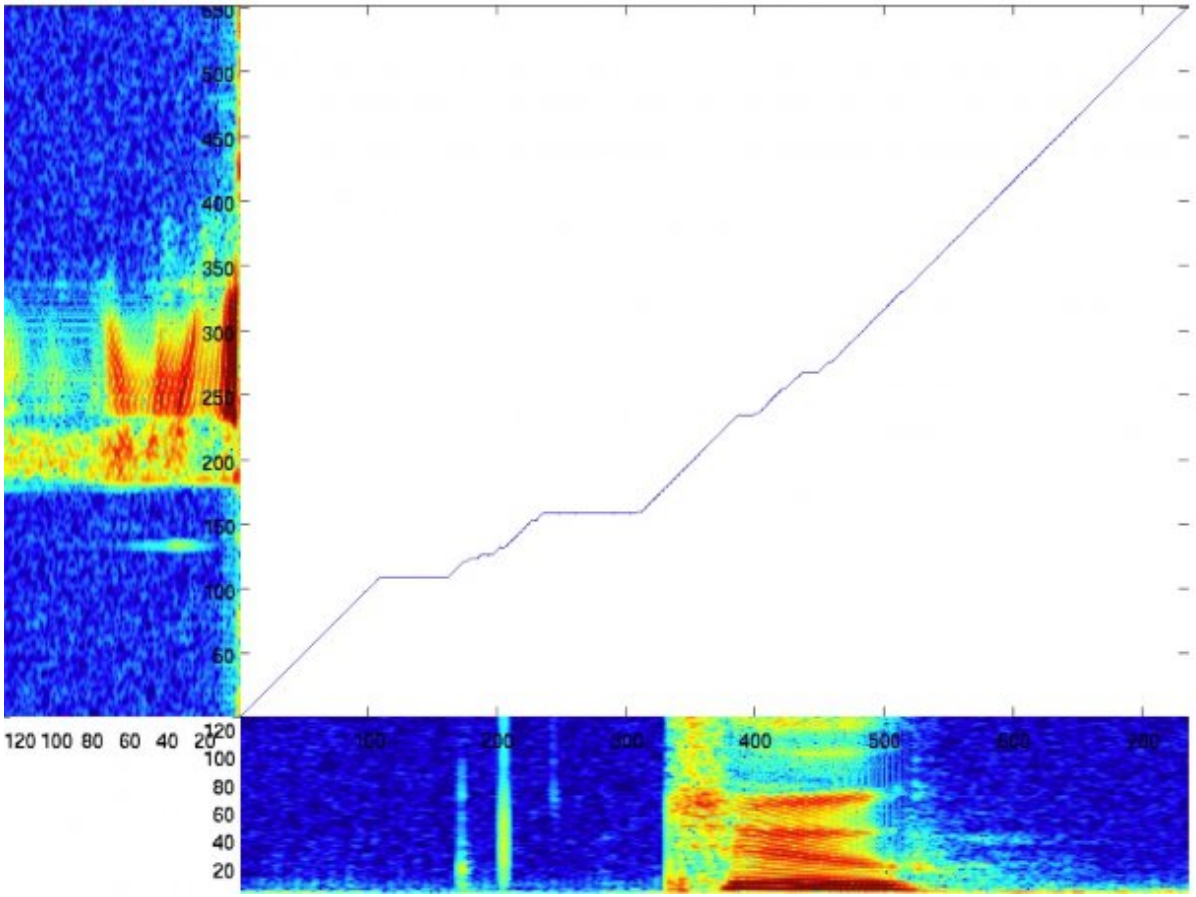
Cost: 1+1+1 = 3

```
for i ranging from 1 to m=size(X)
  for j ranging from 1 to n=size(T)
       B(i,j)=argmin(D(i-1,j),D(i-1,j-1),D(i,j-1))
    D(i,j)=min(D(i-1,j),D(i-1,j-1),D(i,j-1))+dist(X(i),Y(j))
return B,D
```

Backtrace array



DTW for speech: over spectral vectors

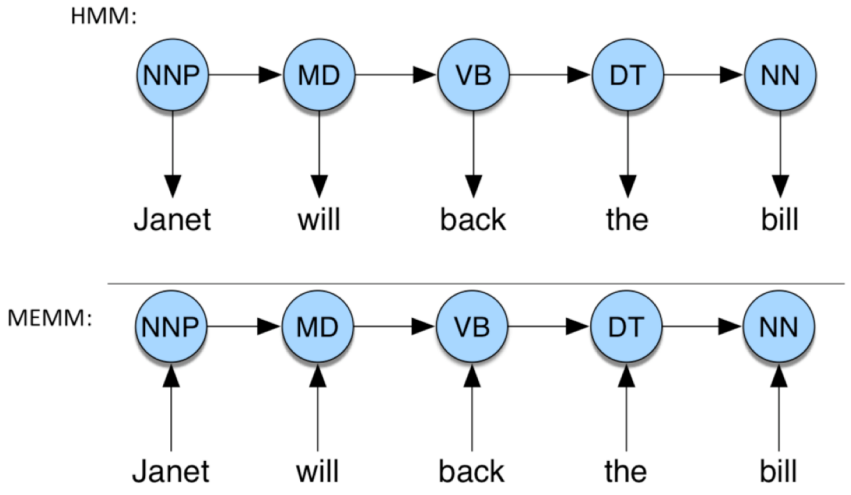


Next idea: probabilistic sequence matching

- Templates only work for utterances you've seen before
- Need a way to generalize to unseen utterances
- Start from probabilistic idea: want most likely **W** for P(**W**|**X**)

- Hidden Markov Model
  - Probabilistic model over sequences
  - You have seen this before in tagging!

$$\begin{aligned}\hat{T} &= \underset{T}{\operatorname{argmax}} P(T|W) \\ &= \underset{T}{\operatorname{argmax}} P(W|T)P(T) \\ &= \underset{T}{\operatorname{argmax}} \prod_i P(word_i|tag_i) \prod_i P(tag_i|tag_{i-1})\end{aligned}$$



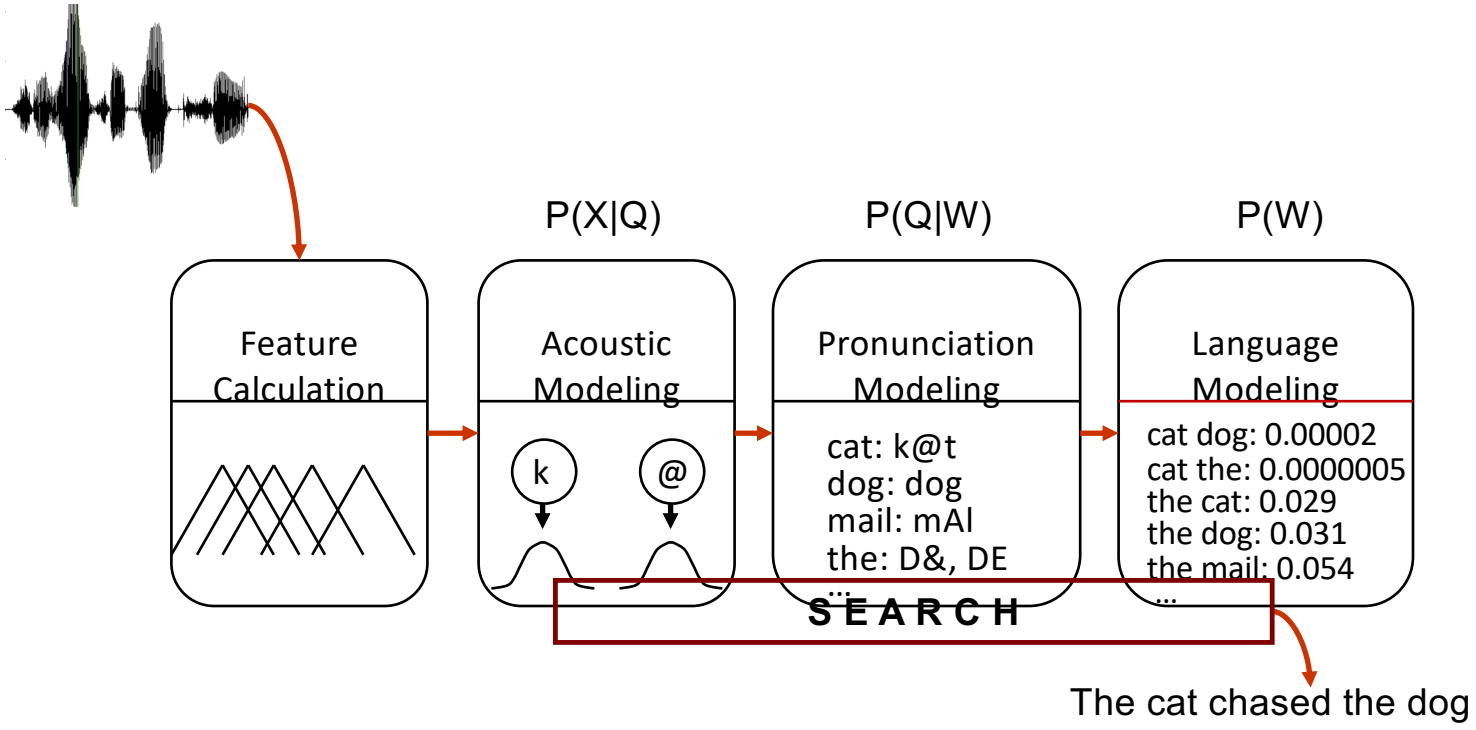
Next idea: probabilistic sequence matching

- Templates only work for utterances you've seen before
- Need a way to generalize to unseen utterances
- Start from probabilistic idea: want most likely **W** for P(**W**|**X**)

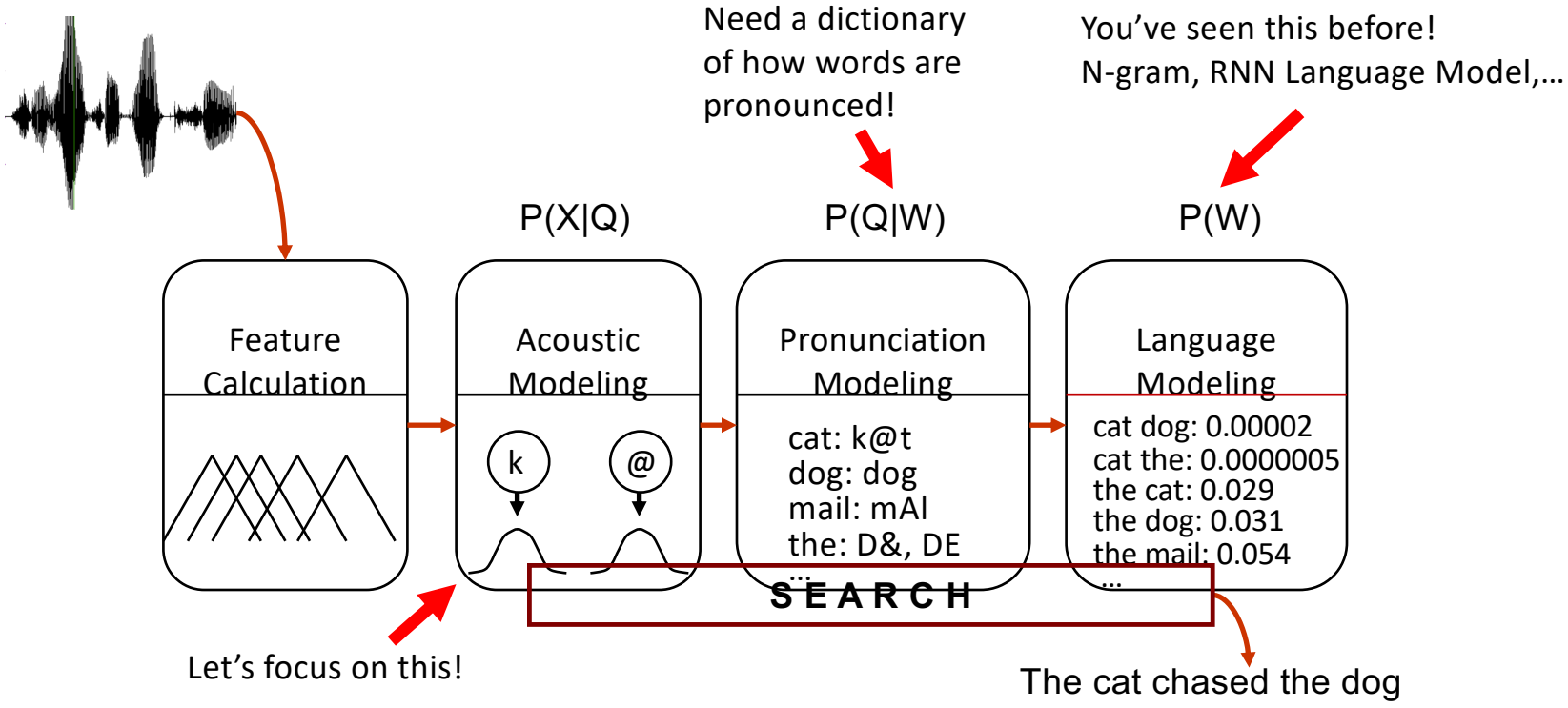
$$
\begin{aligned}
&\text{argmax}_W\ P(\mathbf{W}|\mathbf{X}) \\
&=\text{argmax}_W\ P(\mathbf{X}|\mathbf{W})P(\mathbf{W})/P(\mathbf{X}) \\
&=\text{argmax}_W\ P(\mathbf{X}|\mathbf{W})P(\mathbf{W}) \\
&=\text{argmax}_W\ \Sigma_Q\ P(\mathbf{X},\mathbf{Q}|\mathbf{W})P(\mathbf{W}) \\
&\approx\text{argmax}_W\ \text{max}_Q\ P(\mathbf{X},\mathbf{Q}|\mathbf{W})P(\mathbf{W}) \\
&\approx\text{argmax}_W\ \text{max}_Q\ P(\mathbf{X}|\mathbf{Q})\ P(\mathbf{Q}|\mathbf{W})\ P(\mathbf{W})
\end{aligned}
$$

- (Hey! What's this **Q** sequence?)

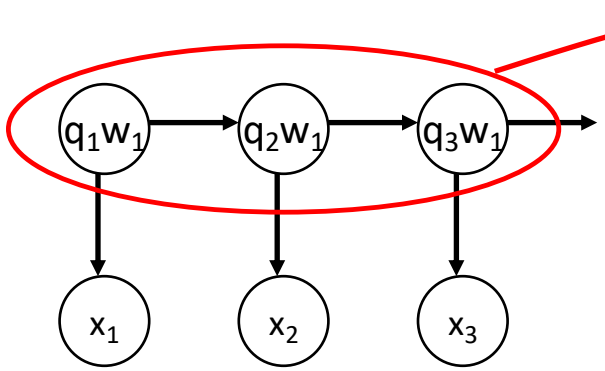
Parts of an HMM-based ASR System



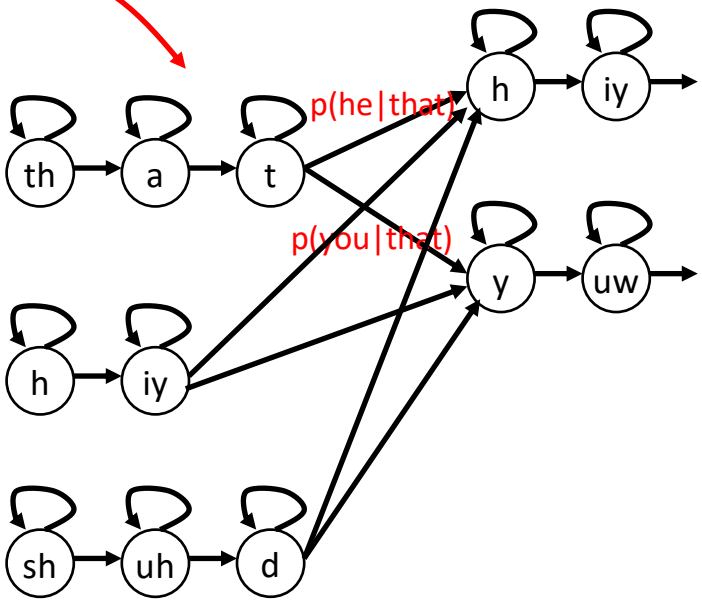
Parts of an HMM-based ASR System



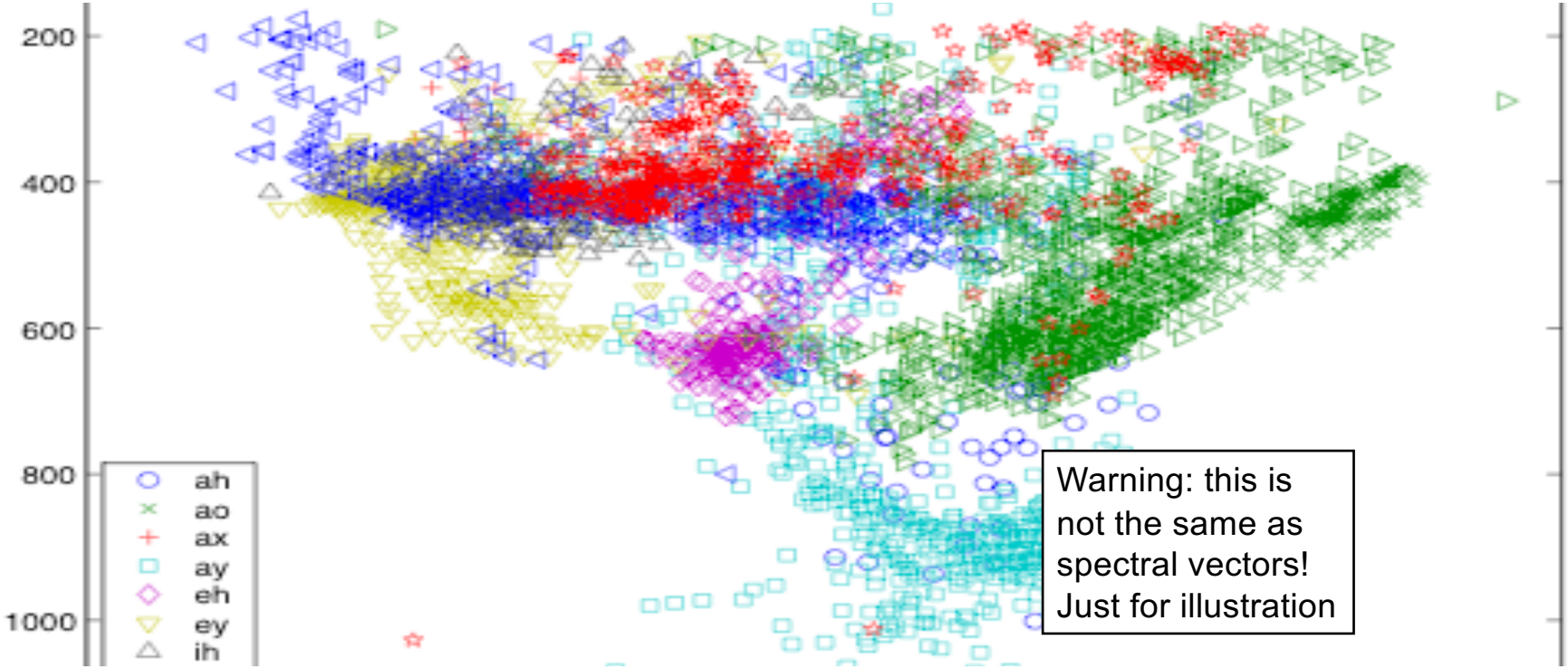
Hidden Markov Models: combining acoustics, phones and words



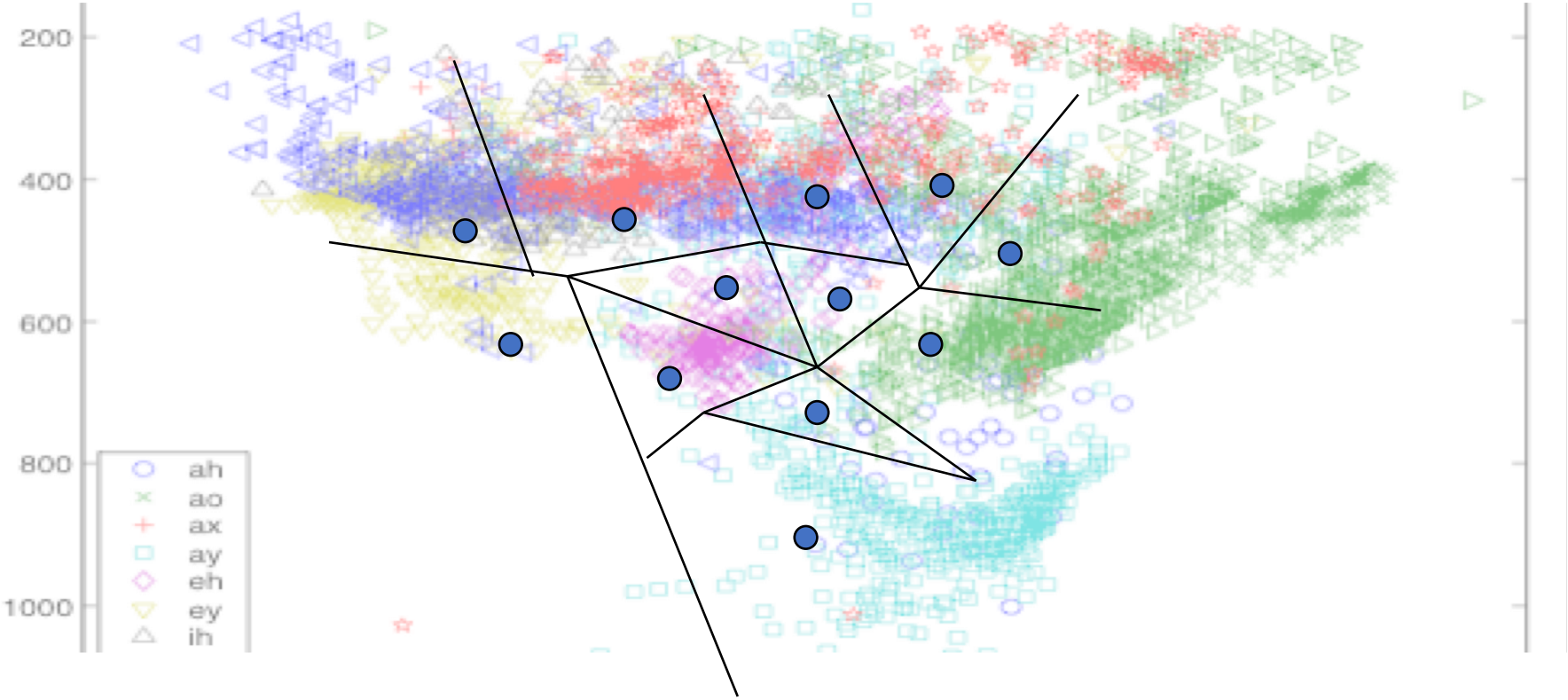
$\text{argmax}_W\ P(W|X)$
$=\text{argmax}_W\ P(X|W)P(W)/P(X)$
$=\text{argmax}_W\ P(X|W)P(W)$
$=\text{argmax}_W\ \Sigma_Q\ P(X,Q|W)P(W)$
$\approx\text{argmax}_W\ \max_Q\ P(X,Q|W)P(W)$
$\approx\text{argmax}_W\ \max_Q\ P(X|Q)\ P(Q|W)\ P(W)$



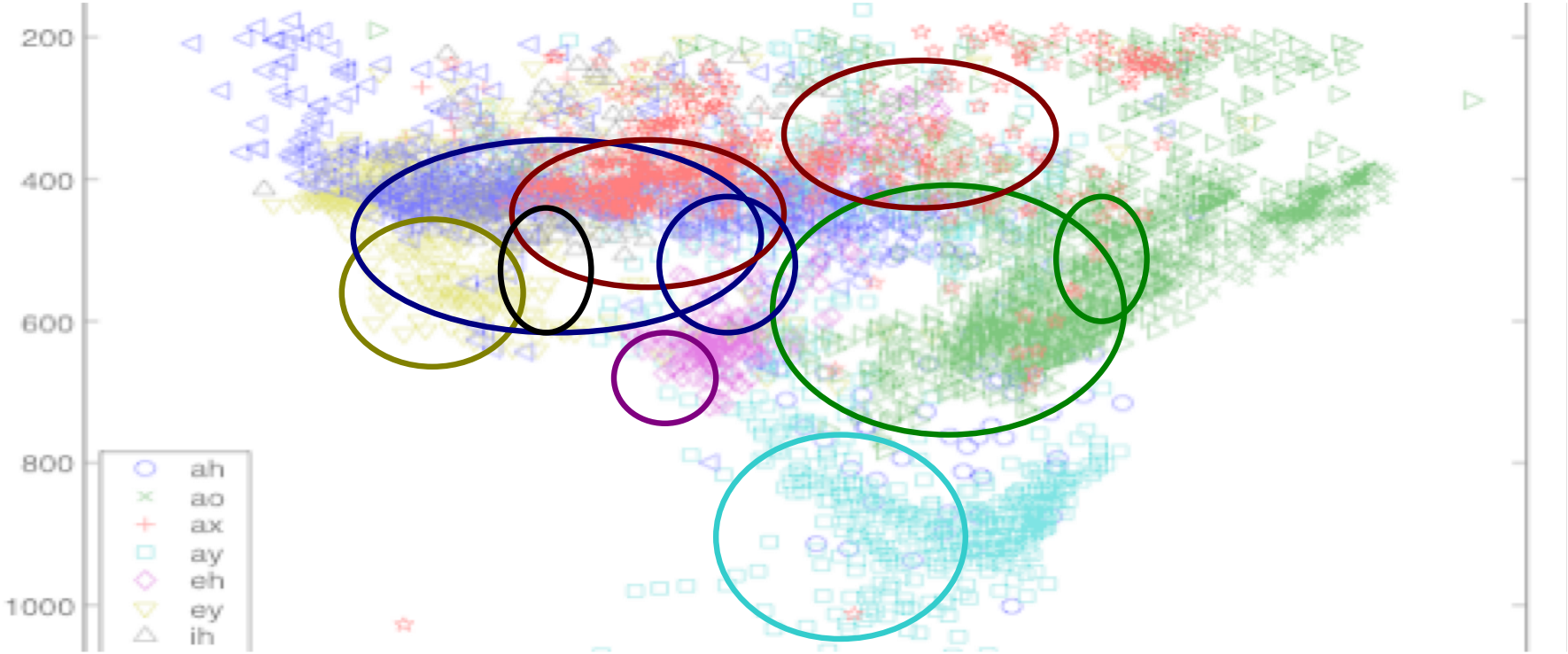
The vowels of digits: first formant vs. second formant



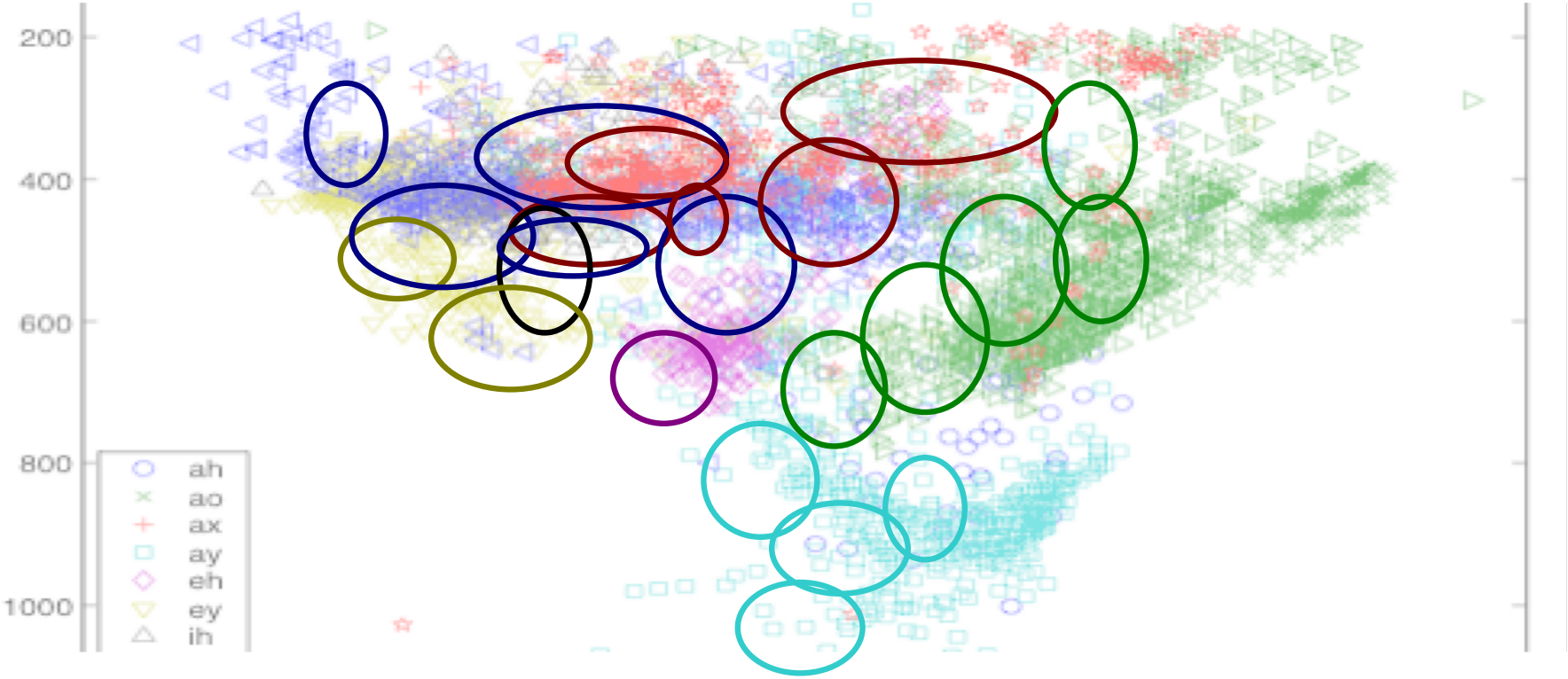
Acoustic Model Option 1: Discretization (Vector Quantization/K-means)



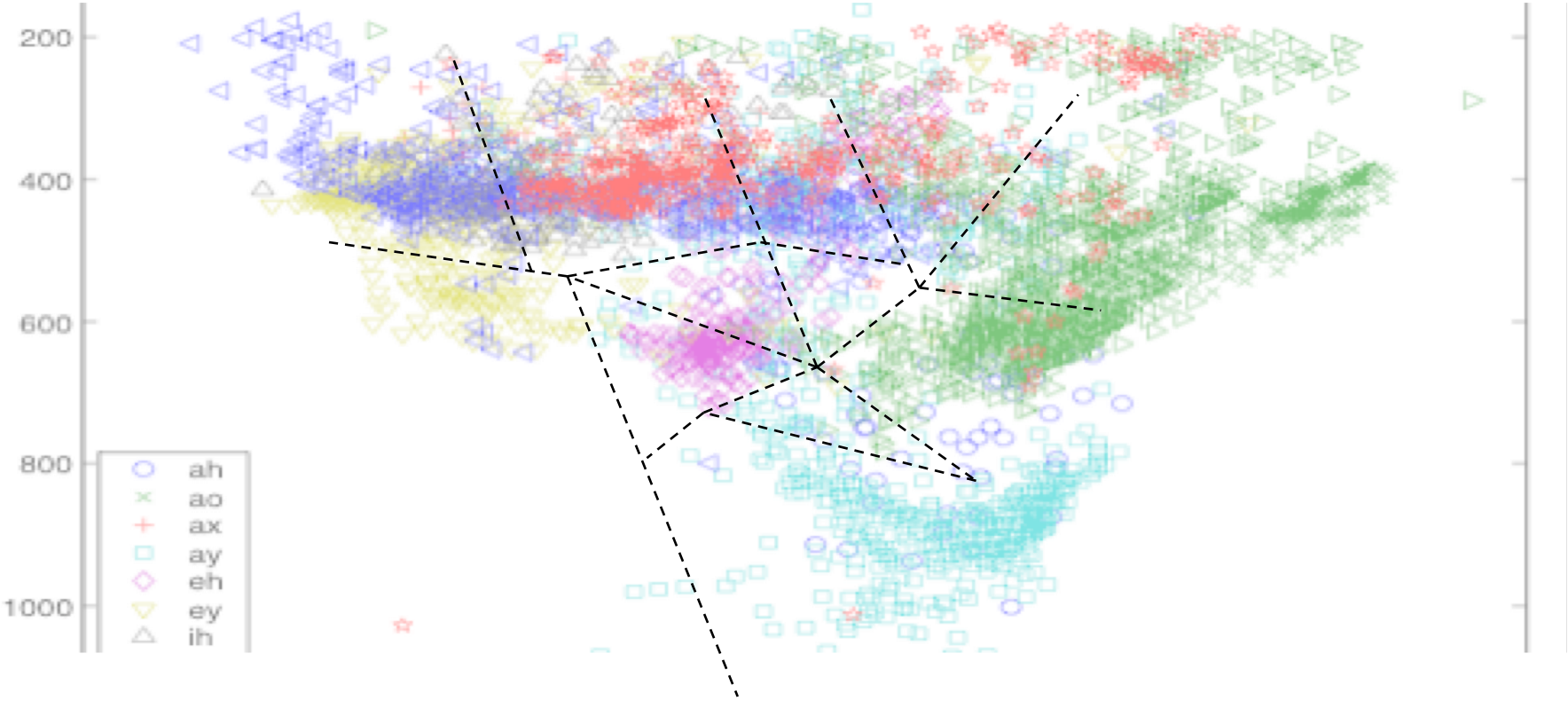
Acoustic Model Option 2: Gaussians



Option 2a: Mixtures of Gaussians



Acoustic Model Option 3: Neural networks



Normal (Gaussian) Distribution

- Also characterized by two parameters (for one-dimensional case)
  - Mean: μ
  - Standard deviaion: σ
- One-dimensional formula:

$$P(x) = N(\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Two (or higher) dimensional Gaussian

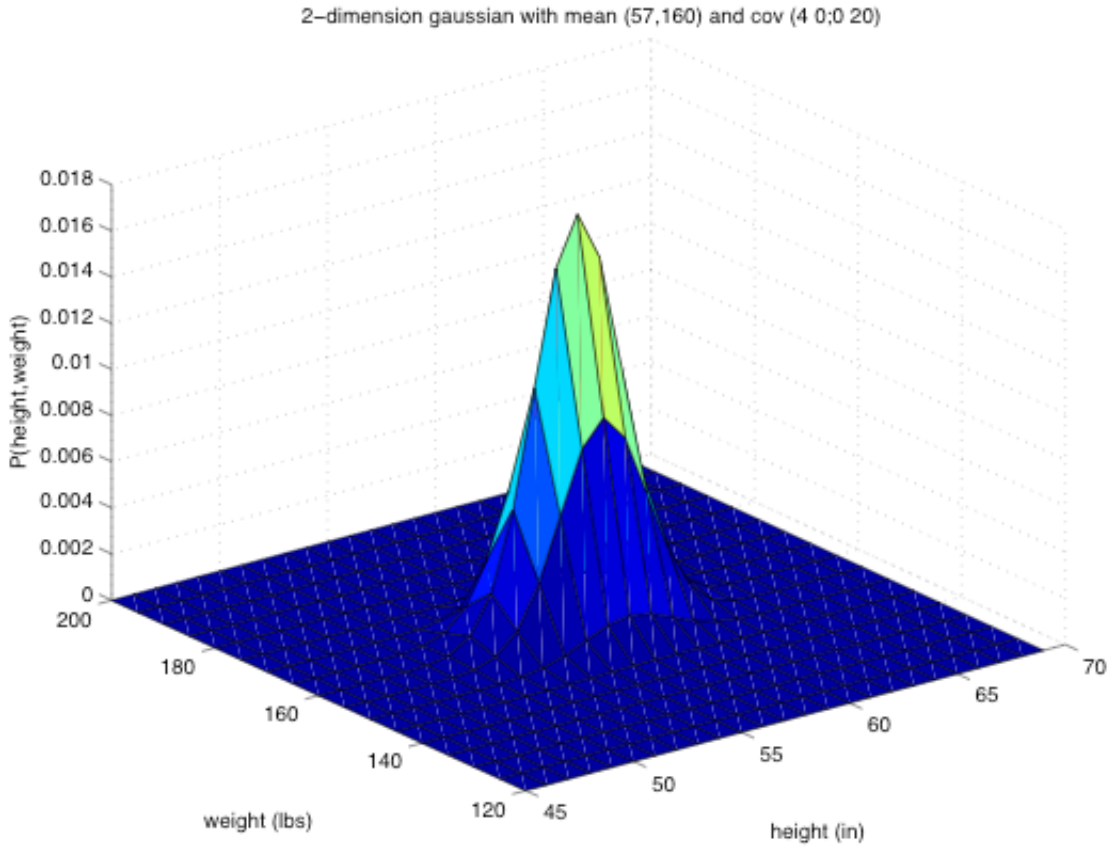
- We can have data points with more than one dimension
  - e.g. height and weight
- One dimensional gaussian:

$$P(x) = N(\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- Multidimensional gaussian

$$N(\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}\left((\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})\right)}$$

Two dimensional Gaussian



Two dimesional Gaussian

- Mean is just a vector
  - Height/weight: (57,160)
- Covariance is a matrix
  - If variables are independent, then covariance is diagonal
  - Diagonal: Non-diagonal:

$$\begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \qquad \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

Building an acoustic model

- If you know the labels Q_i for each frame, then compute $P(X_i|Q_i)$ using Gaussians (or Mixture of Gaussians)
  - This is similar to the Part of Speech Tagging problem: words == X, tags == Q, and you know both at training time
- If you *don't* know Q, then Q is a hidden variable
  - **Expectation-Maximization** algorithm (or **Forward-Backward** algorithm) needed to discover likely associations between data X and labels Q
    - Remember: Viterbi algorithm determines **best path** through an HMM
    - Forward-Backward: find probability of being in each HMM state Q at time i
  - For speech Q is semi-hidden: we know that "cat" must go through "k ae t" phones (Q) in order, but don't know which exact Q sequence matches X

If time… show how EM works for HMMs

Discriminative Acoustic Models

- Gaussians are a *generative* model for the acoustics (X): $P(x_i|q_i)$
  - There is nothing that tries to *discriminate* between different classes

- A discriminative model will try to distinguish classes: $P(q_i|x_i)$
  - Can incorporate more local context: $P(q_i|x_{i\pm c})$
  - Problem: no longer fits into basic HMM equation

$$\prod_{i=1}^{t} P(x_i|q_i)P(q_i|q_{i-1})$$

Scaled Likelihoods

- By Bayes' Rule, $P(x_i|q_i) = \frac{P(q_i|x_i)P(x_i)}{P(q_i)}$
- Scaled likelihood: $P(x_i|q_i) \propto \frac{P(q_i|x_i)}{P(q_i)}$

Scaled Likelihoods

- By Bayes' Rule, $P(x_i|q_i) = \frac{P(q_i|x_i)P(x_i)}{P(q_i)}$
- Scaled likelihood: $P(x_i|q_i) \propto \frac{P(q_i|x_i)}{P(q_i)}$

Posterior: what is the probability of this phone/state given the acoustics at this time?

Prior: what is the probability of this phone/state overall?

- Use the scaled likelihood in place of Gaussian

Neural networks: Discriminative Acoustic Models

- Dominant method of computing local posteriors
- Three basic models in play today
  - DNN (Deep Neural Network) – fully interconnected layers, predicting each time slice given a range of input
  - LSTM (Long-Short Term Memory network) – a recurrent model that allows the network to retain or forget information over time
  - CNN (Convolutional Neural Network) – allows the system to learn patterns over time-frequency patches

Example: (Bi-)LSTMs

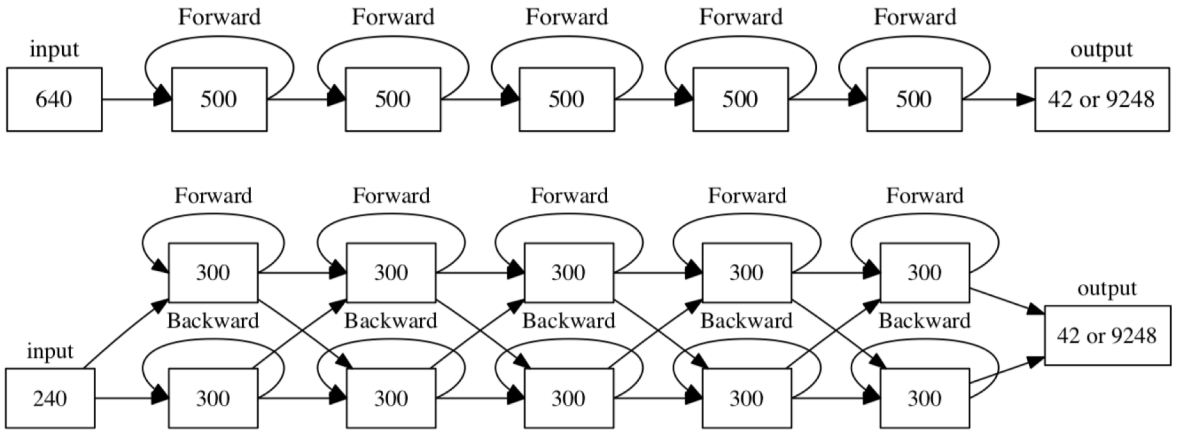


Figure 1: Layer connections in unidirectional (top) and bidirectional (bottom) 5-layer LSTM RNNs.

Image from Sak et al. 2015 arXiv:1507.06947

Changing the training criteria

- Traditional training criteria is a *local* criteria:
  - Maximize the probability of the "correct" class at each time step
  - Often called *cross-entropy* criteria
- Problems:
  - You really care about the sequence, not local prediction
    - k k ae ae ae t t is really same as k k k ae t t t
  - You're not really sure what the correct class should be at any time point

Sequence-level criteria

- sMBR (State-level Minimum Bayes Risk)
  - Rather than local prediction, make sure that the correct HMM states are produced
    - k k k ae ae t t t is ok for ”k ae t”
  - Minimize the risk of producing the wrong sequence of states
  - Requires integration with HMM decoder and NN trainer

- CTC (Connectionist Temporal Classification)
  - Uses a sequence-based NN (e.g. LSTM)
  - Train so network produces correct sequence of phones
    - Allowed to put out a “blank” phone
    - Cat: $\_^* k^+ \_^* ae^+ \_^* t^+$

End-to-end speech recognition

- Previous models have multiple constraints:
  - Need to have a pronunciation dictionary mapping phones to words
    - Grapheme-to-phoneme converters needed for new words
  - Language models are pretty separate from acoustic models
  - Sequence-level training requires integration of HMM decoder with Neural Network trainer (can be hard to modularize)

- Proposed solution:
  - Train NN to directly predict words from the acoustics
    - Build in language models as a layer in NN (eg RNNLM)
    - Ties to the Bayesian probabilistic framework are a bit dubious: direct prediction of outputs
  - Problem: too many words as output tokens
    - Solution: predict character sequences

End-to-end approaches to ASR

- Connectionist Temporal Classification (CTC) approach described earlier can just output characters instead of phones
- Sequence-to-sequence models treat the problem similarly to machine translation (translate acoustics to characters)
- Attention-based approaches learn to focus on particular subparts of the acoustics to produce characters (again like MT)

- Current end-to-end approaches require significantly more amounts of data to match performance of hybrid HMM-NN approaches, but are more flexible when adapting to new vocabularies