EECS E6870
Speech Recognition

Michael Picheny, Stanley F. Chen, Bhuvana Ramabhadran
IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
{picheny,stanchen,bhuvana}@us.ibm.com

8 September 2009

What Is Speech Recognition?

- converting speech to text
  - automatic speech recognition (ASR), speech-to-text (STT)
- what it's not
  - speaker recognition — recognizing who is speaking
  - natural language understanding — understanding what is being said
  - speech synthesis — converting text to speech (TTS)

Why Is Speech Recognition Important?

Ways that people communicate

modality	*method*	*rate (words/min)*
sound	speech	150–200
sight	sign language; gestures	100–150
touch	typing; mousing	60
taste	covering self in food	<1
smell	not showering	<1

Why Is Speech Recognition Important?

- speech is potentially the fastest way people can communicate with machines
  - natural; requires no specialized training
  - can be used in parallel with other modalities
- remote speech access is ubiquitous
  - not everyone has Internet; everyone has a phone
- archiving/indexing/compressing/understanding human speech
  - *e.g.*, transcription: legal, medical, TV
  - *e.g.*, transaction: flight information, name dialing
  - *e.g.*, embedded: navigation from the car

This Course

- cover fundamentals of ASR in depth (weeks 1–9)
- survey state-of-the-art techniques (weeks 10–13)
- force you, the student, to implement key algorithms in C++
  - C++ is the international language of ASR

Speech Recognition Is Multidisciplinary

- too much knowledge to fit in one brain
  - signal processing, machine learning
  - linguistics
  - computational linguistics, natural language processing
  - pattern recognition, artificial intelligence, cognitive science
- three lecturers (no TA?)
  - Michael Picheny
  - Stanley F. Chen
  - Bhuvana Ramabhadran
- from IBM T.J. Watson Research Center, Yorktown Heights, NY
  - hotbed of speech recognition research

Meets Here and Now

- 1300 Mudd; 4:10-6:40pm Tuesday
  - 5 minute break at 5:25pm
- hardcopy of slides distributed at each lecture
  - 4 per page

Assignments

- four programming assignments (80% of grade)
  - implement key algorithms for ASR in C++ (best supported)
  - some short written questions
  - optional exercises for those with excessive leisure time
  - check, check-plus, check-minus grading
- final reading project (undecided; 20% of grade)
  - choose paper(s) about topic not covered in depth in course; give 15-minute presentation summarizing paper(s)
  - programming project
- weekly readings
  - journal/conference articles; book chapters

Course Outline

week	topic	assigned	due
1	Introduction;		
2	Signal processing; DTW	lab 1	
3	Gaussian mixture models; HMMs		
4	Hidden Markov Models	lab 2	lab 1
5	Language modeling		
6	Pronunciation modeling,Decision Trees	lab 3	lab 2
7	LVCSR and finite-state transducers		
8	Search	lab 4	lab 3
9	Robustness; Adaptation		
10	Advanced language modeling	project	lab 4
11	Discriminative training, ROVER		
12	Spoken Document Retrieval, S2S		
13	Project presentations		project

Programming Assignments

- C++ (g++ compiler) on x86 PC's running Linux
  - knowledge of C++ and Unix helpful
- extensive code infrastructure in C++ with SWIG to make it accessible from Java and Python (provided by IBM)
  - you, the student, only have to write the "fun" parts
  - by end of course, you will have written key parts of basic large vocabulary continuous speech recognition system
- get account on ILAB computer cluster
  - complete the survey
- labs due Wednesday at 6pm

Readings

- PDF versions of readings will be available on the web site

- recommended text (bookstore):
  - *Speech Synthesis and Recognition*, Holmes, 2nd edition (paperback, 256 pp., 2001, ISBN 0748408576) [**Holmes**]

- reference texts (library, online, bookstore, EE?):
  - *Fundmentals of Speech Recognition*, Rabiner, Juang (paperback, 496 pp., 1993, ISBN 0130151572) [**R+J**]
  - *Speech and Language Processing*, Jurafsky, Martin (2nd-Ed, hardcover, 1024 pp., 2008, ISBN 01318732210) [**J+M**]
  - *Statistical Methods for Speech Recognition*, Jelinek (hardcover, 305 pp., 1998, ISBN 0262100665) [**Jelinek**]
  - *Spoken Language Processing*, Huang, Acero, Hon (paperback, 1008 pp., 2001, ISBN 0130226165) [**HAH**]

How To Contact Us

- **in E-mail, prefix subject line with “EECS E6870:” !!!**
- Michael Picheny — `picheny@us.ibm.com`
- Stanley F. Chen — `stanchen@watson.ibm.com`
- Bhuvana Ramabhadran — `bhuvana@us.ibm.com`
  - phone: 914-945-2593,914-945-2976
- office hours: right after class; or before class by appointment
- Courseworks
  - for posting questions about labs

Web Site

`http://www.ee.columbia.edu/~stanchen/fall09/e6870/`

- syllabus
- slides from lectures (PDF)
  - online by 8pm the night before each lecture
- lab assignments (PDF)
- reading assignments (PDF)
  - online by lecture they are assigned
  - password-protected (not working right now)
    - username: *speech*, password: *pythonrules*

Help Us Help You

- feedback questionnaire after each lecture (2 questions)
  - feedback welcome any time
- EE’s may find CS parts challenging, and vice versa
- you, the student, are partially responsible for quality of course
- together, we can get through this
- let’s go!

Outline For Rest of Today

1. a brief history of speech recognition
2. speech recognition as pattern classification
   - why is speech recognition hard?
3. speech production and perception
4. introduction to signal processing

A Quick Historical Tour

1. the early years: 1920–1960's
   - *ad hoc* methods

2. the birth of modern ASR: 1970–1980's
   - maturation of statistical methods; basic HMM/GMM framework developed

3. the golden years: 1990's–now
   - more processing power, data
   - variations on a theme; tuning;
   - demand from downstream technologies (search, translation)

The Start of it All

Radio Rex (1920's)

- speaker-independent single-word recognizer ("Rex")
  - triggered if sufficient energy at 500Hz detected (from "e" in "Rex")

The Early Years: 1920–1960's

Ad hoc methods

- simple signal processing/feature extraction
  - detect energy at various frequency bands; or find dominant frequencies
- many ideas central to modern ASR introduced, but not used all together
  - *e.g.*, statistical training; language modeling
- small vocabulary
  - digits; yes/no; vowels
- not tested with many speakers (usually $<$10)
- error rates $< 10\%$

The Turning Point

Whither Speech Recognition? John Pierce, Bell Labs, 1969

> Speech recognition has glamour. Funds have been available. Results have been less glamorous . . .
>
> . . . General-purpose speech recognition seems far away. Special-purpose speech recognition is severely limited. It would seem appropriate for people to ask themselves why they are working in the field and what they can expect to accomplish . . .
>
> . . . These considerations lead us to believe that a general phonetic typewriter is simply impossible unless the typewriter has an intelligence and a knowledge of language comparable to those of a native speaker of English . . .

The Turning Point

- killed ASR research at Bell Labs for many years
- partially served as impetus for first (D)ARPA program (1971–1976) funding ASR research
  - goal: integrate speech knowledge, linguistics, and AI to make a breakthrough in ASR
  - large vocabulary: 1000 words; artificial syntax
  - $<60\times$ “real time”

The Turning Point

- four competitors
  - three used hand-derived rules, scores based on “knowledge” of speech and language
  - HARPY (CMU): integrated all knowledge sources into finite-state network that was trained statistically
- HARPY won hands down

The Turning Point

Rise of probabilistic data-driven methods (1970's and on)

- view speech recognition as . . .
  - finding *most probable* word sequence given the audio signal
    - given some informative probability distribution
  - train probability distribution automatically from transcribed speech
  - minimal amount of explicit knowledge of speech and language used
- downfall of trying to manually encode intensive amounts of linguistic, phonetic knowledge

The Birth of Modern ASR: 1970–1980's

- basic paradigm/algorithms developed during this time still used today
  - expectation-maximization algorithm; n-gram models; Gaussian mixtures; Hidden Markov models; Viterbi decoding; etc.
- then, computer power still catching up to algorithms
  - first real-time dictation system built in 1984 (IBM)

The Golden Years: 1990's–now

- dramatic growth in available computing power
  - first demonstration of real-time large vocabulary ASR (1984)
    - specialized hardware $\approx$ 60 MHz Pentium
  - today: 3 GHz CPU's are cheap

- dramatic growth in transcribed data sets available
  - 1971 ARPA initiative: training on $<$ 1 hour of speech
  - today: systems trained on thousands of hours of speech

- basic algorithmic framework remains the same as in the 1980's
  - significant advances in adaptation; discriminative training
  - lots of tuning and twiddling improvements

Not All Recognizers Are Created Equal

More processing power and data lets us do more difficult things

- speaker dependent *vs.* speaker independent
  - recognize single speaker or many
- small *vs.* large vocabulary
  - recognize from list of digits or list of cities
- constrained *vs.* unconstrained domain
  - air travel reservation system *vs.* E-mail dictation
- isolated *vs.* continuous
  - pause between each word or speak naturally
- read *vs.* spontaneous
  - news broadcasts or telephone conversations

Commercial Speech Recognition

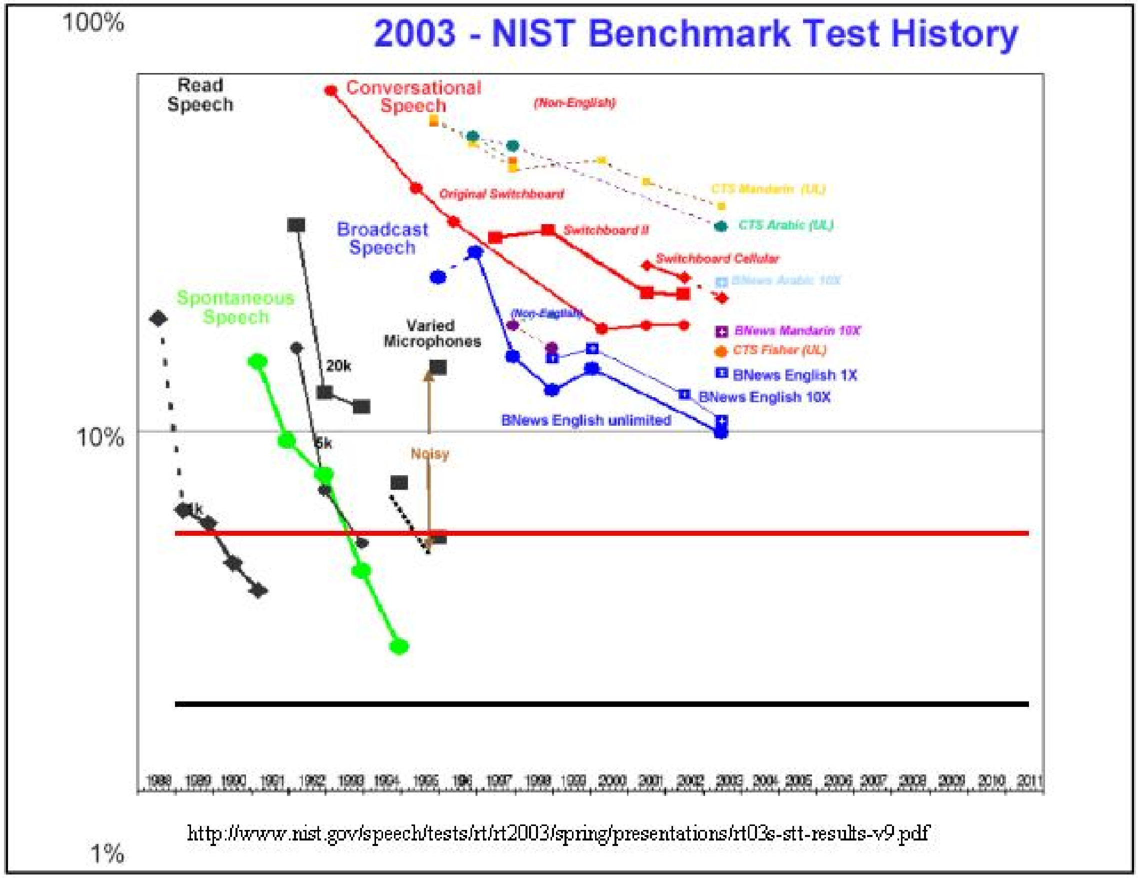
- 1995 — Dragon, IBM release speaker-dependent isolated word large-vocabulary dictation systems
- 1997 — Dragon, IBM release speaker-dependent continuous word large-vocabulary dictation systems
- late 1990's — speaker-independent continuous small-vocab ASR available over the phone
- late 1990's — limited-domain speaker-independent continuous large-vocabulary ASR available over the phone
- to get reasonable performance, must constrain something
  - speaker, vocabulary, domain
  - word error rates can be $<$ 5%, or not

Research Systems

Driven by government-funded evaluations (DARPA, NIST, etc.)

- different sites compete on a common test set
- harder and harder problems over time
  - read speech: TIMIT, resource management (1,000 word vocab), Wall Street Journal (5,000–20,000 word vocab), Broadcast News (partially spontaneous, background music)
  - spontaneous speech: air travel domain (ATIS), Switchboard (telephone), Call Home (accented)
  - Mandarin, Arabic (GALE)
  - Many more languages...

Research Systems



http://www.nist.gov/speech/tests/rt/rt2003/spring/presentations/rt03s-stt-results-v9.pdf

Where Are We Now?

Task	Word error rate
Broadcast News	$<$10%
conversational telephone (Switchboard)	$<$15%
meeting transcription (close-talking mike)	$<$25%
meeting transcription (far-field mike)	$\sim$50%
accented elderly speech (Malach)	$<$30%

- each system has been extensively tuned to that domain!
- still a ways to go until unconstrained large-vocabulary speaker-independent ASR is a reality

Where Are We Now?

Human word error rates an order of magnitude below that of machines (Lippmann, 1997)

- for humans, one system fits all

Task	Machine Performance	Human Performance
Connected Digits[1]	0.72%	0.009%
Letters[2]	5.0%	1.6%
Resource Management	3.6%	0.1%
WSJ	7.2%	0.9%
Timit[3]	20.0%	1.0%
SWITCHBOARD	30%	4.0%

[1] string error rates, [3] phone error rates

[2] isolated letters presented to humans, continuous for machine

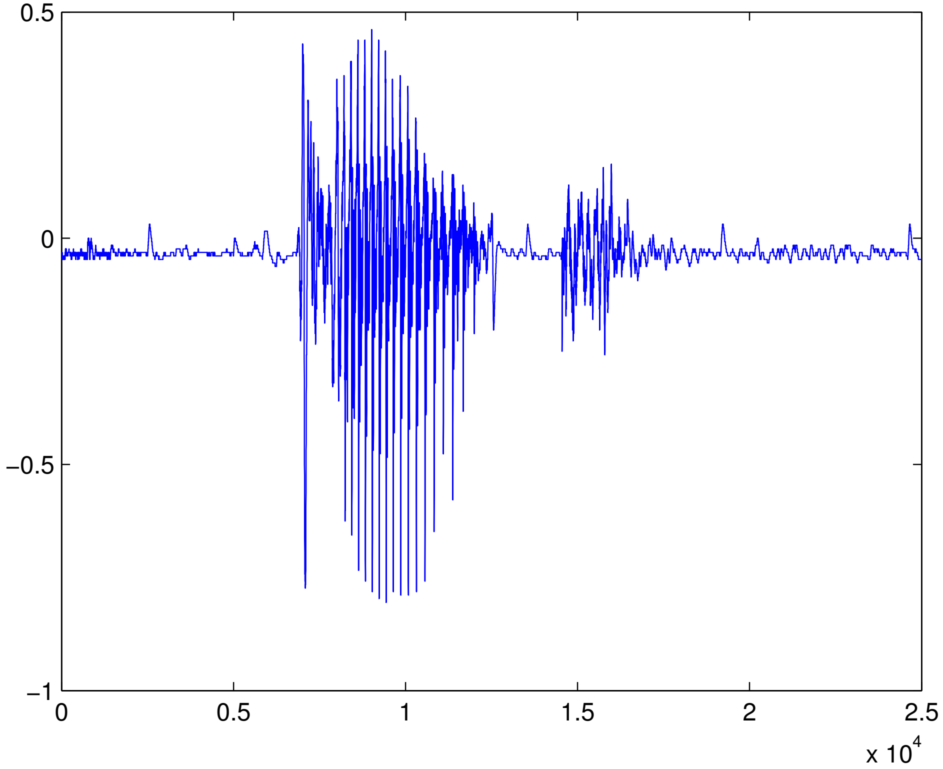
The Big Picture

- speech recognition as pattern classification
- why is speech recognition so difficult?
- key problems in speech recognition

Speech Recognition as Pattern Classification

- consider isolated digit recognition
  - person speaks a single digit $\in 0, \ldots, 9$
  - recognize which digit was spoken
- classification
  - which of ten classes does audio signal (A) belong to?

Speech Recognition as Pattern Classification



- What does an audio signal look like?
  - *e.g.*, turn on microphone for exactly one second
  - microphone converts instantaneous air pressure into real value

Speech Recognition as Pattern Classification

Discretizing the audio signal

- discretizing in time
  - sampling rate, *e.g.*, 16000 samples/sec (Hz)
- discretizing in magnitude (A/D conversion)
  - *e.g.*, 16-bit A/D returns integer value $\in [-32768, +32767]$
- one second audio signal $A \in \mathcal{R}^{16000}$
  - vector of 16000 real values, *e.g.*, [0, -1, 4, 16, 23, 7, . . .]

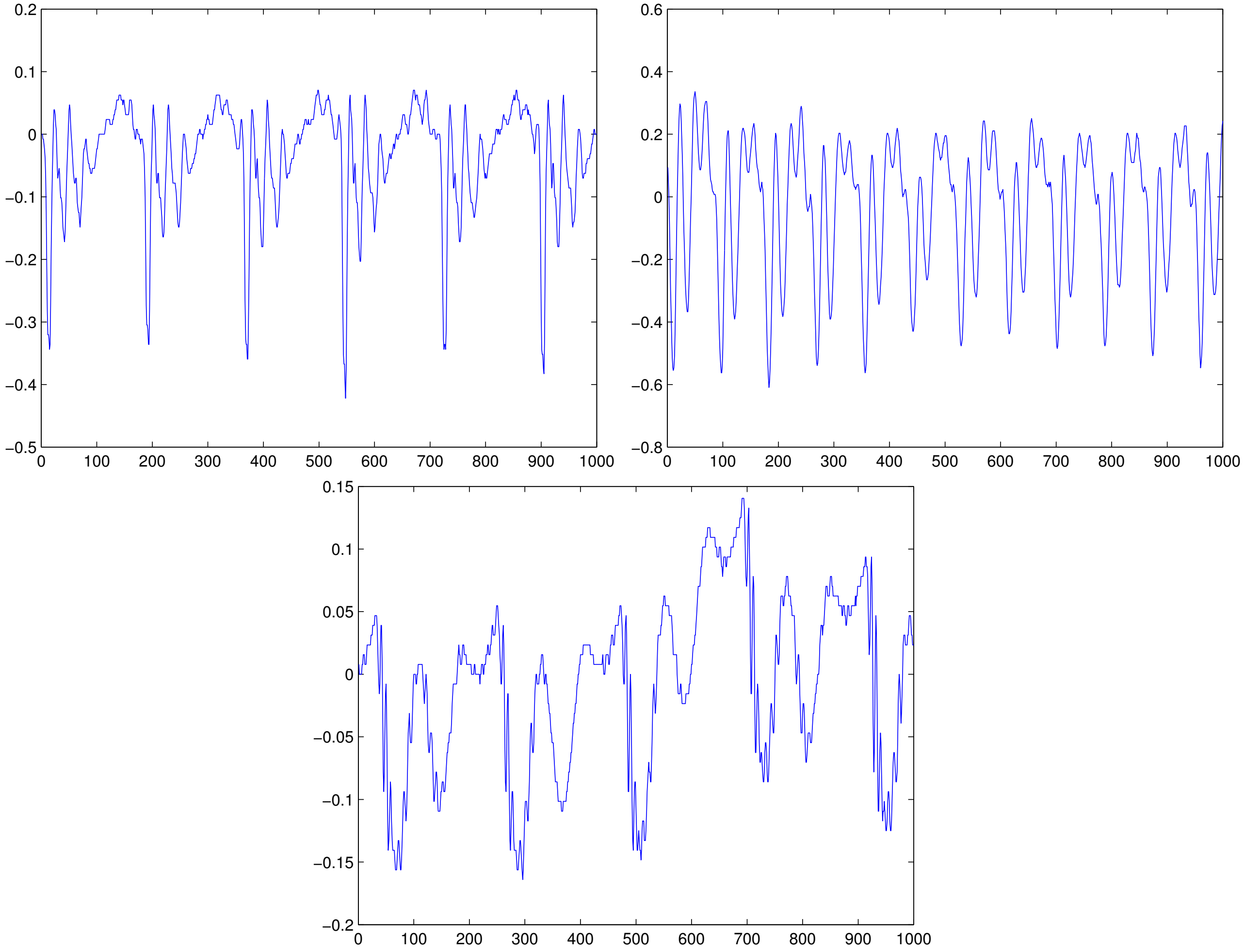
Speech Recognition as Pattern Classification

- speech recognition $\Leftrightarrow$ building a classifier
  - discriminant function $\textsc{score}_c(A)$ for $c = 0, \ldots, 9$
    - *e.g.*, how much (little) signal A sounds like digit c
  - pick class c with highest (lowest) $\textsc{score}_c(A)$
- speech recognition $\Leftrightarrow$ design discriminant function $\textsc{score}_c(A)$
- can use concepts, tools from pattern classification

Speech Recognition as Pattern Classification

- a simple classifier
  - collect single example A_c of each digit $c = 0, \ldots, 9$
- discriminant function $\textsf{SCORE}_c(A) = \textsf{DISTANCE}(A, A_c)$
  - Euclidean distance? ($\sqrt{\sum_{i=1}^{16000} (a_i - a_{i,c})^2}$)
- pick class whose example is closest to A
- *e.g.*, scenario for cell phone name recognition

Why Is Speech Recognition Hard?



Why Is Speech Recognition Hard?

- wait, taking Euclidean distance in the time domain is dumb!
- what about the frequency domain?
  - a waveform can be decomposed into its energy at each frequency
  - spectrogram is graph of energy at each frequency over time

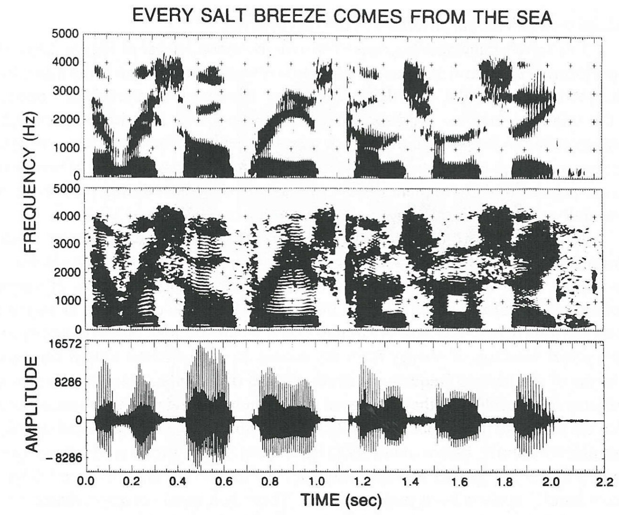
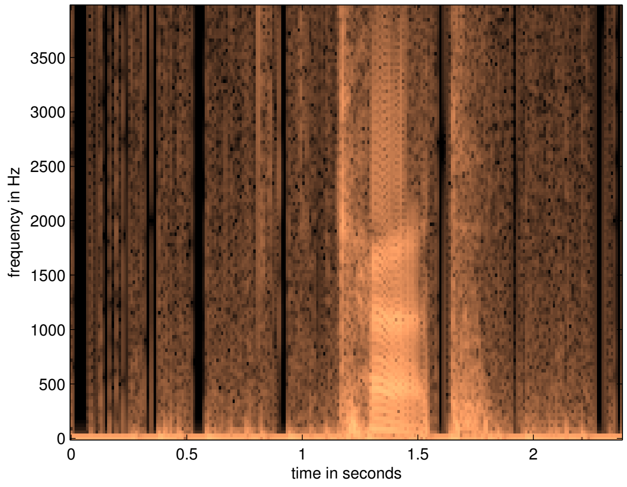
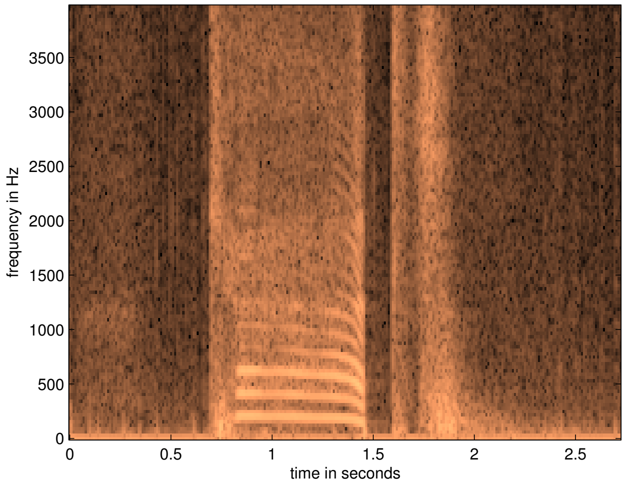
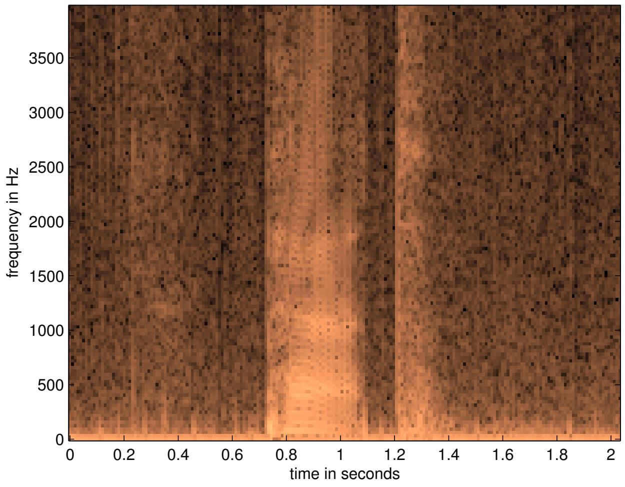


Figure 2.8 Wideband and narrowband spectrograms and speech amplitude for the utterance "Every salt breeze comes from the sea."

Why Is Speech Recognition Hard?







Why Is Speech Recognition Hard?

- taking Euclidean distance in the frequency domain doesn't work well either

- can we extract cogent features $A \Rightarrow (f_1, \ldots, f_k)$
  - such that can use simple distance measure between feature vectors to do accurate classification

- this turns out to be remarkably difficult!

Why Is Speech Recognition Hard?

- there is a enormous range of ways a particular word can be realized
- sources of variability
  - source variation
    - volume; rate; pitch; accent; dialect; voice quality (*e.g.*, gender); coarticulation; context
  - channel variation
    - type of microphone; position relative to microphone (angle + distance); background noise
- screwing with any one of these factors can make ASR accuracy go to hell

Key Problems In Speech Recognition

At a high level, ASR systems are simple classifiers

- for each word w, collect many examples; summarize with set of canonical examples $A_{w,1}, A_{w,2}, \ldots$
- to recognize audio signal A, find word w that minimizes DISTANCE$(A, A_{w,i})$

Key Problems

- converting audio signals A into a set of cogent features values $(f_1, \ldots, f_k)$ so simple distance measures work well
  - signal processing; robustness; adaptation
- coming up with good distance measures DISTANCE$(\cdot, \cdot)$
  - dynamic time warping; hidden Markov models; GMM's

Key Problems In Speech Recognition (Cont'd)

- coming up with good canonical representatives $A_{w,i}$ for each class
  - Gaussian mixture models (GMM's); discriminative training
- what if don't have examples for each word? (sparse data)
  - pronunciation modeling
- efficiently finding the closest word
  - search; finite-state transducers
- using knowledge that not all words or word sequences are equally probable
  - language modeling

Finding Good Features

- find features of speech such that . . .
  - similar sounds have similar feature values
  - dissimilar sounds have dissimilar feature values
- discard stuff that doesn’t matter
  - *e.g.*, pitch (English)
- look at human production and perception for insight

Speech Production

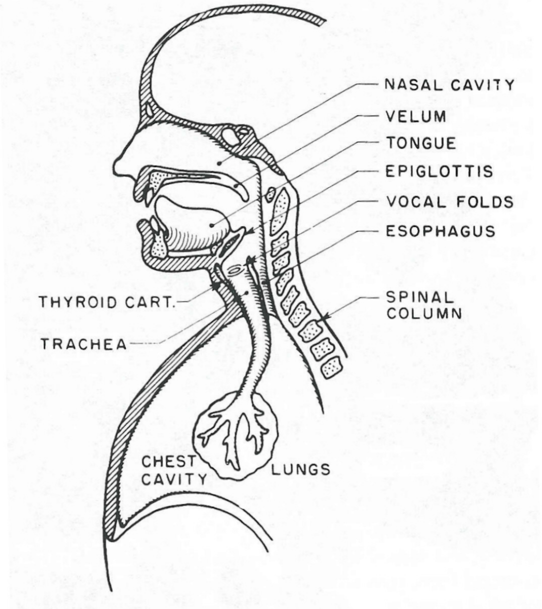


Figure 2.4 Schematic view of the human vocal mechanism (after Flanagan [3]).

- air comes out of lungs
- vocal cords tensed (vibrate ⇒ voicing) or relaxed (unvoiced)
- modulated by vocal tract (glottis → lips); resonates
  - articulators: jaw, tongue, velum, lips, mouth

Speech Is Made Up Of a Few Primitive Sounds?

- phonemes
  - 40 to 50 for English
  - speaker/dialect differences
    - are the vowels in MARY, MARRY, and MERRY different?
  - phone: acoustic realization of a phoneme

- may be realized differently based on context
  - *allophones*: different ways a phoneme can be realized
    - P in SPIN, PIN are two different allophones of P phoneme
    - T in BAT, BATTER; A in BAT, BAD

Classes of Speech Sounds

Can categorize phonemes by how they are produced

- voicing
  - *e.g.*, F (unvoiced), V (voiced)
  - all vowels are voiced
- stops/plosives
  - oral cavity blocked (*e.g.*, lips, velum); then opened
  - *e.g.*, P, B (lips)

Classes of Speech Sounds

- spectogram shows energy at each frequency over time
- voiced sounds have pitch (F0); formants (F1, F2, F3)
- trained humans can do recognition on spectrograms with high accuracy (*e.g.*, Victor Zue)

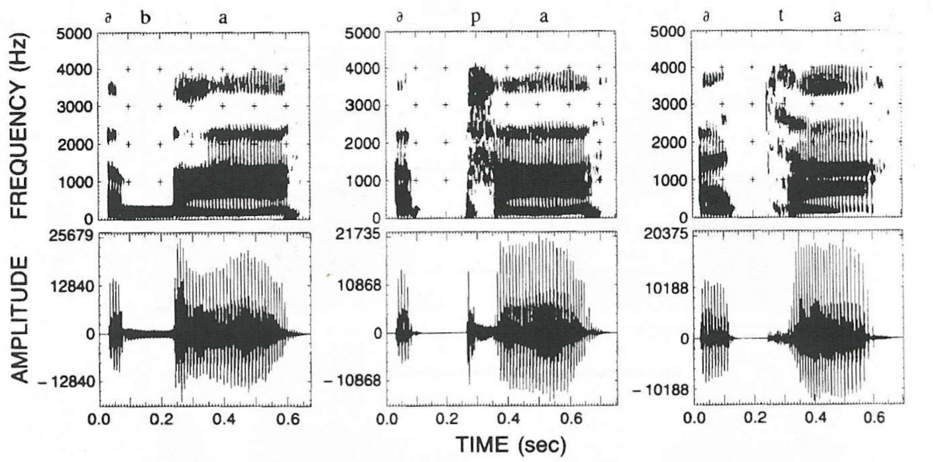
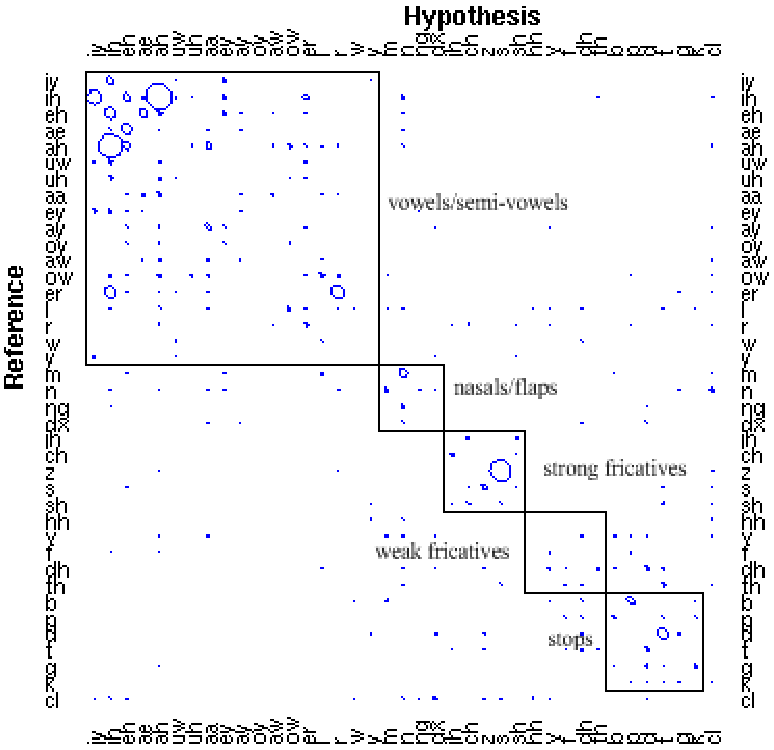


Figure 2.28 Spectrogram comparisons of the sequences of voiced (/ə-b-a/) and voiceless (/ə-p-a/ and /ə-t-a/) stop consonants.

Classes of Speech Sounds

- What can the machine do? Here is a sample on TIMIT:



Classes of Speech Sounds

- vowels — EE, AH, etc.
  - differ in locations of formants
  - dipthongs — transition between two vowels (*e.g.*, COY, COW)
- consonants
  - fricatives — F, V, S, Z, SH, J
  - stops/plosives — P, T, B, D, G, K
  - nasals — N, M, NG
  - semivowels (liquids, glides) — W, L, R, Y

Coarticulation

- realization of a phoneme can differ very much depending on context (allophones)
- where articulators were for last phone affect how they transition to next

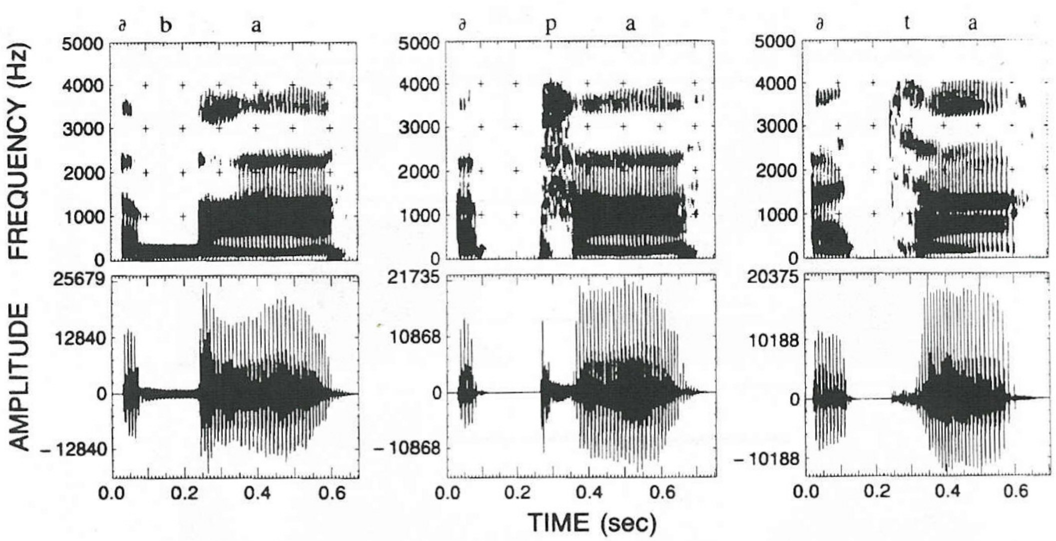


Figure 2.28 Spectrogram comparisons of the sequences of voiced (/ə-b-a/) and voiceless (/ə-p-a/ and /ə-t-a/) stop consonants.

Speech Production

Can we use knowledge of speech production to help speech recognition?

- insight into what features to use?
  - (inferred) location of articulators; voicing; formant frequencies
  - in practice, these features provide little or no improvement over features less directly based on acoustic phonetics
- influences how signal processing is done
  - source-filter model
  - separate excitation from modulation from vocal tract
  - *e.g.*, frequency of excitation can be ignored (English)

Speech Perception

- as it turns out, the features that work well . . .
  - motivated more by speech perception than production
- *e.g.*, Mel Frequency Cepstral Coefficients (MFCC)
  - motivated by how humans perceive pitches to be spaced
  - similarly for perceptual linear prediction (PLP)

Speech Perception — Physiology

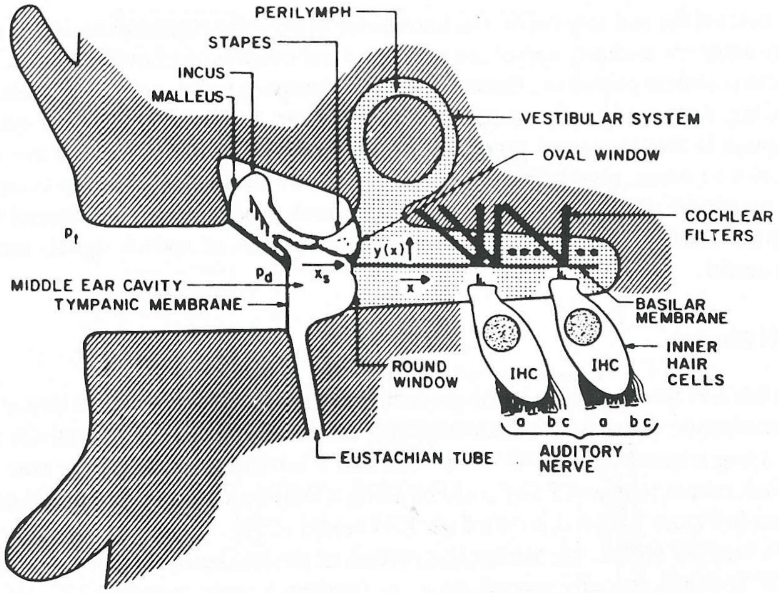


Figure 3.48 Expanded view of the middle and inner ear mechanics.

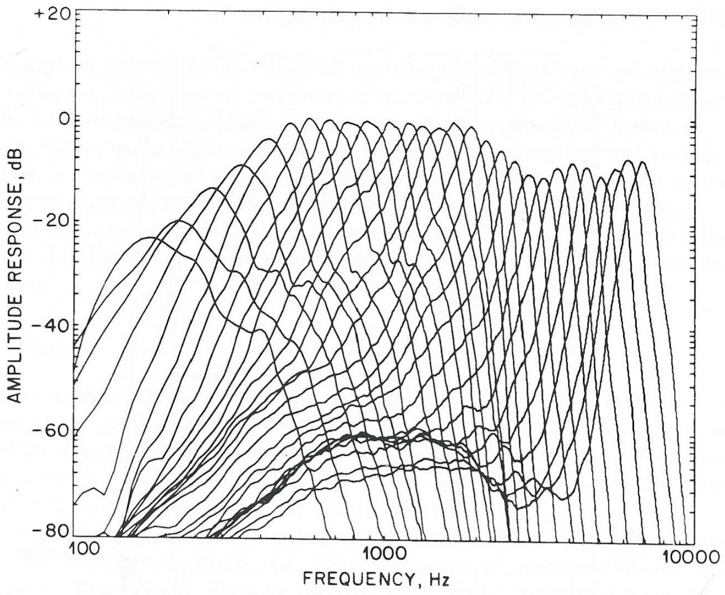


Figure 3.50 Frequency response curves of a cat's basilar membrane (after Ghitza [13]).

- sound comes in ear, converted into vibrations in fluid in cochlea
- in fluid is basilar membrane, with ∼30,000 little hairs
  - hairs sensitive to different frequencies (band-pass filters)

Speech Perception — Physiology

- human physiology used as justification for frequency analysis ubiquitous in speech processing
- limited knowledge of higher-level processing
  - can glean insight from psychophysical experiments
  - relationship between physical stimuli and psychological effects

Speech Perception — Psychophysics

Threshold of hearing as a function of frequency

- 0 dB sound pressure level (SPL) ⇔ threshold of hearing
  - +20 decibels (dB) ⇔ 10× increase in pressure/loudness
- tells us what range of frequencies people can detect

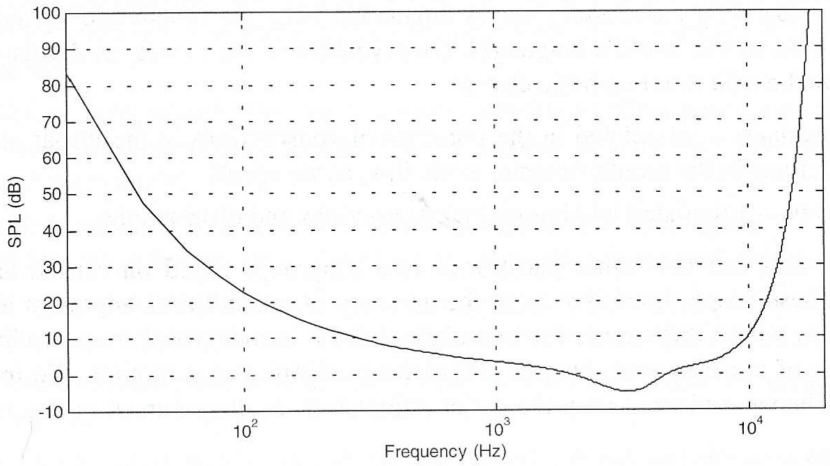


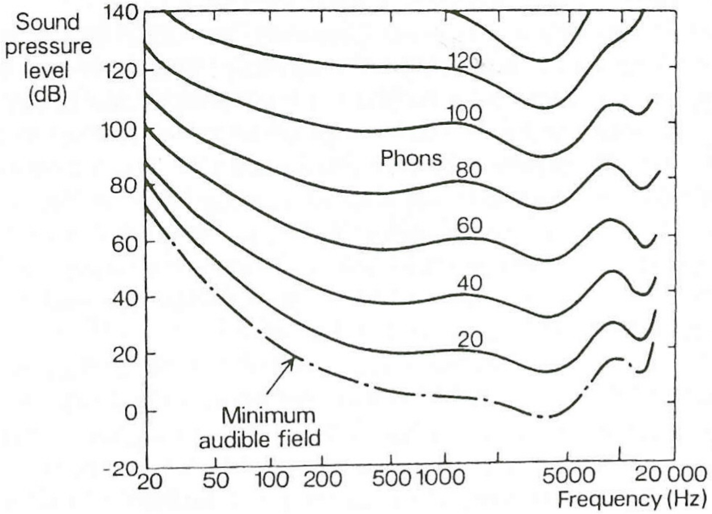
Figure 2.3 The sound pressure level (SPL) level in dB of the absolute threshold of hearing as a function of frequency. Sounds below this level are inaudible. Note that below 100 Hz and above 10 kHz this level rises very rapidly. Frequency goes from 20 Hz to 20 kHz and is plotted in a logarithmic scale from Eq. (2.3).

Speech Perception — Psychophysics

Sensitivity of humans to different frequencies

- equal loudness contours
  - subjects adjust volume of tone to match volume of another tone at different pitch

- tells us what range of frequencies might be good to focus on



Speech Perception — Psychophysics

Human perception of distance between frequencies

- adjust pitch of one tone until twice/half pitch of other tone
- Mel scale — frequencies equally spaced in Mel scale are equally spaced according to human perception

$$\text{Mel freq} = 2595 \log_{10}(1 + \text{freq}/700)$$

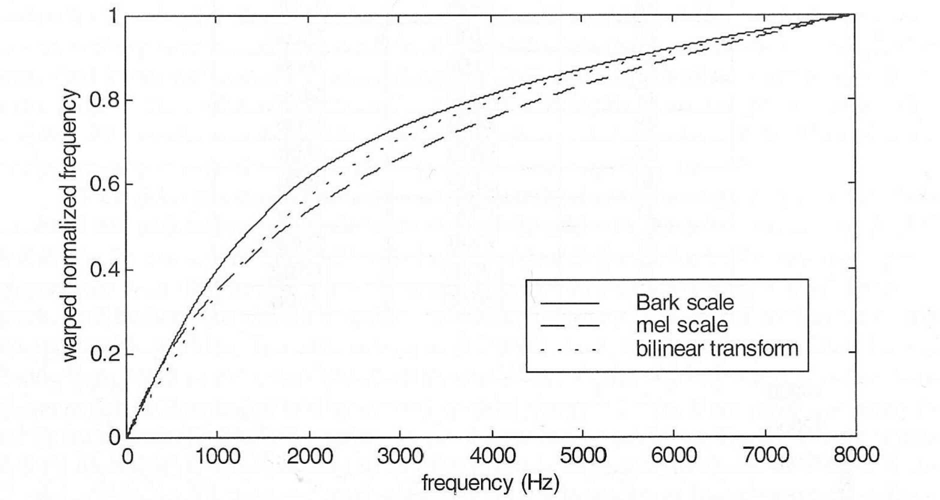


Figure 2.13 Frequency warping according to the Bark scale, ERB scale, mel-scale, and bilinear nsform for $\alpha = 0.6$: linear frequency in the x-axis and normalized frequency in the y-axis.

Speech Perception — Psychoacoustics

- use controlled stimuli to see what features humans use to distinguish sounds

- Haskins Laboratories (1940–1950's), Pattern Playback machine
  - synthesize sound from hand-painted spectrograms

- demonstrated importance of formants, formant transitions, trajectories in human perception
  - *e.g.*, varying second formant alone can distinguish between B, D, G

`http://www.haskins.yale.edu/haskins/MISC/PP/bdg/bdg.html`

Speech Perception — Machine

- just as human physiology has its quirks, so does machine “physiology”
- sources of distortion
  - microphone — different response based on direction and frequency of sound
  - sampling frequency
    - telephones — 8 kHz sampling; throw away all frequencies above 4 kHz (“low bandwith”)
  - analog/digital conversion — need to convert to digital with sufficient precision (8–16 bits)
  - lossy compression — *e.g.*, cellular telephones
  - voip (compressed audio over the internet)

Speech Perception — Machine

- input distortion can still be a significant problem
  - mismatched conditions — train/test in different conditions
  - low bandwidth — telephone, cellular
  - cheap equipment — *e.g.*, mikes in handheld devices
- enough said

Segue

- now that we see what humans do
- let’s discuss what signal processing has been found to work well empirically
  - has been tuned over decades
- goal: ignoring time alignment issues . . .
  - how to process signals to produce features . . .
  - so that alike sounds generate similar feature values
- start with some mathematical background

Signal Processing Basics — Motivation

Goal: Review some basics about signal processing to provide an appropriate context for the details and issues involved in feature extraction, which will be discussed next week.

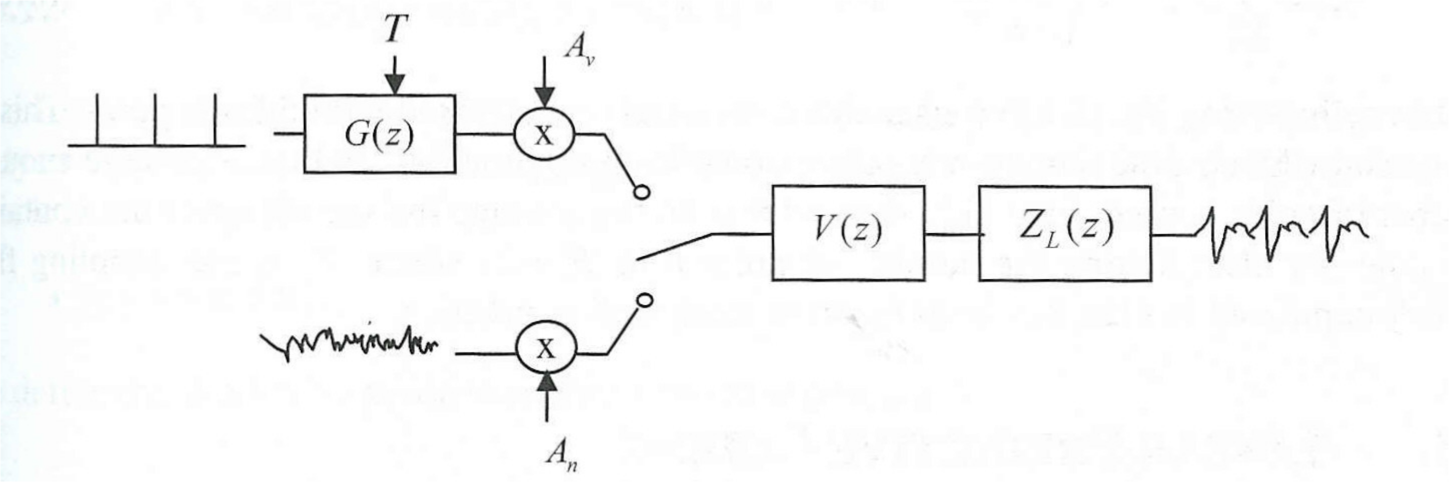
- Present enough about signal processing to allow you to understand how we can digitally simulate banks of filters, similar to those present in the human peripheral auditory system

- Describe some basic properties of linear systems, since linear channel variability is one of the main problems speech recognition systems need to be able to cope with to achieve robustness.

Recommended Readings: HAH pg. 201-223, 242-245. R+J pg. 69-91. All figures taken from these sources unless indicated otherwise.

Source-Filter Model

A simple popular model for the vocal tract is the *Source-Filter* model in which the vocal tract is modeled as a sequence of filters representing the various functions of the vocal tract.



The initial filter, $G(z)$, represents the effect of the glottis. Differences in the glottal waveform (essentially different amounts of low-frequency emphasis) are one of the main sources of interspeaker differences. $V(z)$ represents the effects of the vocal tract — a linear filter with time varying resonances. Note that the length of the vocal tract, which strongly determines the general positions of the resonances, is another major source of interspeaker differences. The last filter, $Z_L(z)$ represents the effects of radiation from the lips and is basically a simple high-frequency pre-emphasis.

Signal Processing Basics — Linear Time Invariant Systems

The output of our A/D converter is a signal $x[n]$.

A digital system T takes and input signal $x[n]$ and produces a signal $y[n]$:

$$y[n] = T(x[n])$$

Calculating the output of T to an input signal x becomes very simple if a digital system T satisfies two basic properties

T is *linear* if

$$T(a_1 x_1[n] + a_2 x_2[n]) = a_1 T(x_1[n]) + a_2 T(x_2[n])$$

T is *time-invariant* if

$$y[n - n_0] = T(x[n - n_0])$$

i.e., a shift in the time axis of x produces the same output, except for a time shift.

Therefore, if $h[n]$ is the response of an LTI system to an impulse $\delta[n]$ (a signal which is 1 at $n = 0$ and 0 otherwise) the response of the the system to an arbitrary signal, $x[n]$, because of linearity and time invariance, will just be the weighted superposition of the impulse responses:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{k=-\infty}^{\infty} x[n-k]h[k]$$

The above is also known as *Convolution* and is written as

$$y[n] = x[n] * h[n]$$

Linear Time Invariant Systems and Sinusoids

A sinusoid $\cos(\omega n + \phi)$ can also be written as $\Re(e^{j(\omega n+\phi)})$ — a complex exponential. It is more convenient to work directly with complex exponentials for ease of manipulation.

If $x[n] = e^{j\omega n}$ then

$$y[n] = \sum_{k=-\infty}^{\infty} e^{j\omega(n-k)}h[k] = e^{j\omega n} \sum_{k=-\infty}^{\infty} e^{-j\omega k}h[k] = H(e^{j\omega})e^{j\omega n}$$

Hence if the input to an LTI system is a complex exponential, the output is just a scaled and phase-adjusted version of the same complex exponential.

So if we can decompose $x[n] = \int X(e^{j\omega})e^{-j\omega n}d\omega$ by the LTI property

$$y[n] = \int H(e^{j\omega})X(e^{j\omega})e^{-j\omega n}d\omega$$

We will not try to prove this here, but the above decomposition can almost always be performed for most functions of interest.

Fourier Transforms and Z-Transforms

The Fourier Transform of a discrete signal is defined as

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h[n]e^{-j\omega n}$$

Note this is a complex quantity, with a magnitude $|H(e^{j\omega})|$ and a phase $e^{j\arg[H(e^{j\omega})]}$

The inverse Fourier Transform is defined as

$$h[n] = 1/(2\pi)\int_{-\pi}^{\pi} H(e^{j\omega})e^{j\omega n}d\omega$$

The Fourier transform is invertible, and exists as long as $\sum_{-\infty}^{\infty} |h[n]| < \infty$

One can generalize the Fourier Transform to

$$H(z) = \sum_{n=-\infty}^{\infty} h(n) z^{-n}$$

where z is any complex variable. The Fourier Transform is just the z-transform evaluated at $z = e^{-j\omega}$.

The z-transform concept allows DSPers to analyze a large range of signals, even those whose integrals are unbounded. We will primarily just use it as a notational convenience, though.

The main property we will use is the convolution property:

$$Y(z) \quad = \quad \sum_{n=-\infty}^{\infty} y[n] z^{-n} = \sum_{n=-\infty}^{\infty} \left(\sum_{k=-\infty}^{\infty} x[k] h[n-k] \right) z^{-n}$$

$$
\begin{aligned}
&= \sum_{k=-\infty}^{\infty} x[k] (\sum_{n=-\infty}^{\infty} h[n-k] z^{-n}) = \sum_{k=-\infty}^{\infty} x[k] (\sum_{n=-\infty}^{\infty} h[n] z^{-(n+k)}) \\
&= \sum_{k=-\infty}^{\infty} x[k] z^{-k} H(z) = X(z) H(z)
\end{aligned}
$$

The autocorrelation of $x[n]$ is defined as

$$R_{xx}[n] = \sum_{m=-\infty}^{\infty} x[m+n] x^*[m] = x[n] * x^*(-n)$$

The Fourier Transform of $R_{xx}[n]$, denoted as $S_{xx}(e^{j\omega})$, is called the *power spectrum* and is just $|X(e^{j\omega})|^2$

Notice also that

$$\sum_{n=-\infty}^{\infty} |x[n]|^2 = 1/(2\pi) \int_{-\pi}^{\pi} |X(e^{j\omega})|^2$$

Lastly, observe that there is a duality between the time and frequency domains; convolution in the time domain is the same as multiplication in the frequency domain, and visa-versa:

$$x[n]y[n] = X(e^{j\omega}) * Y(e^{j\omega})$$

This will become important later when we discuss the effects of windowing on the speech signal.

The DFT — Discrete Fourier Transform

We usually compute the Fourier Transform digitally. We obviously cannot afford to deal with infinite signals, so assuming that $x[n]$ is finite and of length N we can define

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\omega n} = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi kn}{N}}$$

where we have replaced ω with $\frac{2\pi k}{N}$

The inverse of the DFT is

$$\begin{aligned} \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j\frac{2\pi kn}{N}} &= \frac{1}{N}\sum_{k=0}^{N-1}[\sum_{m=0}^{N-1} x[m]e^{-j\frac{2\pi km}{N}}]e^{j\frac{2\pi kn}{N}} \\ &= \frac{1}{N}\sum_{m=0}^{N-1} x[m]\sum_{n=0}^{N-1} e^{j\frac{2\pi k(n-m)}{N}} \end{aligned}$$

Note that the last term on the right is N for $m = n$ and 0 otherwise, so the entire right side is just $x[n]$ Note that the DFT is equivalent to a Fourier series expansion of a periodic version of $x[n]$.

The Fast Fourier Transform

Note that the computation of

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi kn}{N}} = \sum_{n=0}^{N-1} x[n]W_N^{nk}$$

for k=0..N-1, where $W_N^{nk} = e^{-j\frac{2\pi kn}{N}}$ requires $\sim O(N^2)$ operations.

Let $f[n] = x[2n]$ and $g[n] = x[2n+1]$. The above equation becomes

$$\begin{aligned} X[k] &= \sum_{n=0}^{N/2-1} f[n]W_{N/2}^{nk} + W_N^k \sum_{n=0}^{N/2-1} g[n]W_{N/2}^{nk} \\ &= F[k] + W_N^k G[k] \end{aligned}$$

when $F[k]$ and $G[k]$ are the $N/2$ point DFTs of $f[n]$ and $g[n]$. To produce values for $X[k]$ for $N > k \geq N/2$, note that $F[k+N/2] = F[k]$ and $G[k+N/2] = G[k]$.

The above process can be iterated to produce a way of computing the DFT with $O(N \log N)$ operations, a significant savings over $O(N^2)$ operations.

The Discrete Cosine Transform

The Discrete Cosine Transform (DCT) is defined as

$$C[k] = \sum_{n=0}^{N-1} x[n] \cos(\pi k(n + 1/2)/N), 0 \leq k < N$$

If we create a signal

$$\begin{aligned} y[n] &= x[n], 0 \leq n < N \\ y[n] &= x[2N - 1 - n], N \leq n < 2N \end{aligned}$$

then $Y[k]$, the DFT of $y[n]$ is

$$\begin{aligned} Y[k] &= 2e^{j\frac{\pi k}{2N}} C[k], 0 \leq k < N \\ Y[2N - k] &= 2e^{-j\frac{\pi k}{2N}} C[k], 0 \leq k < N \end{aligned}$$

By creating such a signal, the overall energy will be concentrated at lower frequency components (because discontinuities at the boundaries will be minimized). The coefficients are also all real. This allows for easier truncation during approximation and will come in handy later when computing MFCCs.

Windowing

All signals we deal with are finite. We may view this as taking an infinitely long signal and multiplying it with a finite window.

Rectangular Window

$$h[n] = 1, 0 \le n < N-1, 0\text{otherwise}$$

The FFT can be written in closed form as

$$\frac{\sin \omega N/2}{\sin \omega/2} e^{-j\omega(N-1)/2}$$

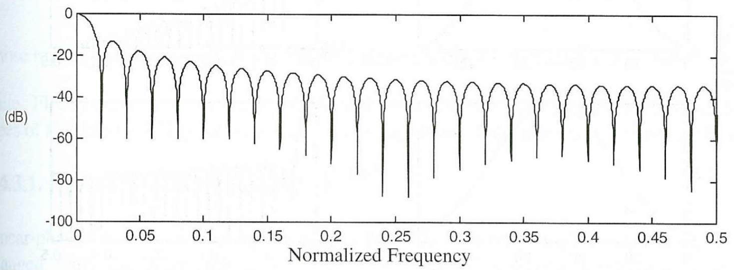


Figure 5.19 Frequency response (magnitude in dB) of the rectangular window with $N = 50$, which is a digital sinc function.

Note the high sidelobes of the window. Since multiplication in the time domain is the same as convolution in the frequency domain, the high sidelobes tend to distort low energy components in the spectrum when there are significant high-energy components also present.

Hamming and Hanning Windows

$$h[n] = .5 - .5\cos 2\pi n/N \text{(Hanning)}$$

$$h[n] = .54 - .46\cos 2\pi n/N \text{(Hamming)}$$

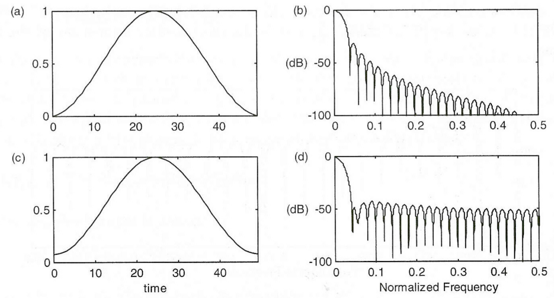


Figure 5.20 (a) Hanning window and (b) the magnitude of its frequency response in dB; (c) Hamming window and (d) the magnitude of its frequency response in dB for $N = 50$.

Observe the different sidelobe behaviors. Both the Hanning and Hamming windows have slightly wider main lobes but much lower sidelobes than the rectangular window. The Hamming window has a lower first sidelobe than a Hanning window, but the sidelobes at higher frequencies do not roll off as much.

Implementation of Filter Banks

A common operation in speech recognition feature extraction is the implementation of filter banks.

The simplest technique is brute force convolution:

$$x_i[n] = x[n] * h_i[n] = \sum_{m=0}^{L_i - 1} h_i[m]x[n-m]$$

The computation is on the order of L_i for each filter for each output point n, which is large.

Say now $h_i[n] = h[n]e^{j\omega_i n}$, a fixed length low pass filter heterodyned up (remember, multiplication in the time domain is the same as convolution in the frequency domain) to be centered at different frequencies. In such a case

$$x_i[n] = \sum h[m]e^{j\omega_i m}x[n-m]$$

$$= \quad e^{j\omega_i n} \sum x[m]h[n-m]e^{-j\omega_i m}$$

The last term on the right is just $X_n(e^{j\omega})$, the Fourier transform of a windowed signal., where now the window is the same as the filter. So we can interpret the FFT as just the instantaneous filter outputs of a uniform filter bank whose bandwidths corresponding to each filter are the same as the main lobe width of the window. Notice that by combining various filter bank channels we can create nom-uniform filterbanks in frequency.

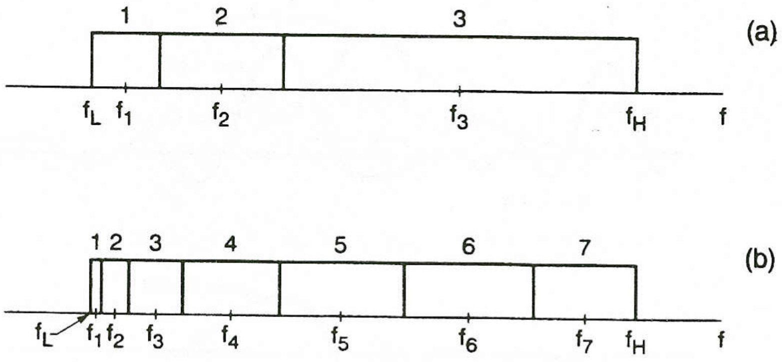


Figure 3.18 Two arbitrary nonuniform filter-bank ideal filter specifications consisting of either 3 bands (part a) or 7 bands (part b).

All this will prove useful in our discussion of mel-scaled filter banks, next week!