EE E6820: Speech & Audio Processing & Recognition

Lecture 9: Speech Recognition

Dan Ellis <dpwe@ee.columbia.edu>
Michael Mandel <mim@ee.columbia.edu>

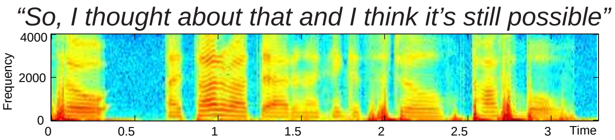
Columbia University Dept. of Electrical Engineering
http://www.ee.columbia.edu/~dpwe/e6820

April 7, 2009

1. Recognizing speech
2. Feature calculation
3. Sequence recognition
4. Large vocabulary, continuous speech recognition (LVCSR)

Outline

Recognizing speech



- What kind of information might we want from the speech signal?
  - words
  - phrasing, 'speech acts' (prosody)
  - mood / emotion
  - speaker identity
- What kind of processing do we need to get at that information?
  - time scale of feature extraction
  - signal aspects to capture in features
  - signal aspects to exclude from features

Speech recognition as Transcription

- Transcription = "speech to text"
  - find a word string to match the utterance
- Gives neat objective measure: word error rate (WER) %
  - can be a sensitive measure of performance

Reference:	THE	CAT	SAT	ON	THE		MAT
Recognized:	–	CAT	SAT	AN	THE	A	MAT

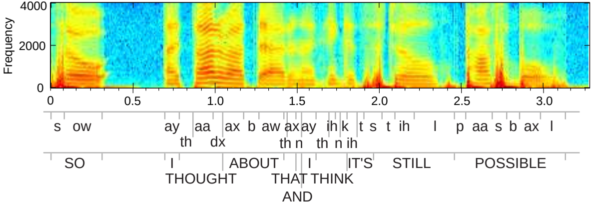
Deletion *Substitution* *Insertion*

- Three kinds of errors:

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

Problems: Within-speaker variability

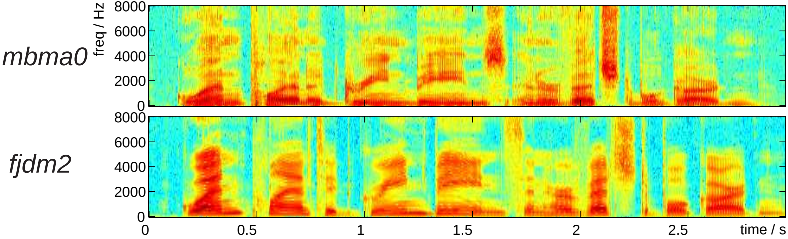
- Timing variation
  - word duration varies enormously



  - 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**?"

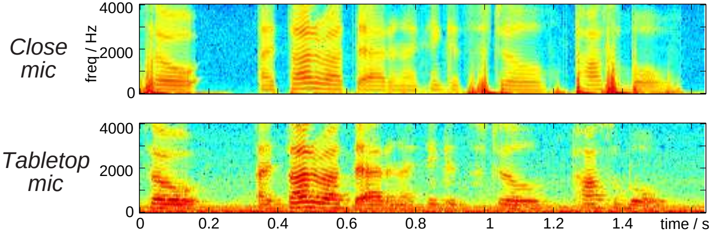
Problems: Between-speaker variability

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



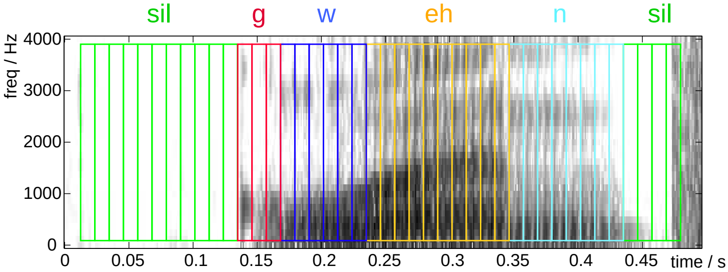
Problems: Environment variability

- 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:



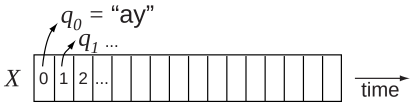
  - other sources of variation...

Probabilistic formulation

- Probability that segment label is correct
  - gives standard form of speech recognizers
- Feature calculation: $s[n] \rightarrow X_m \quad (m = \frac{n}{H})$
  - transforms signal into easily-classified domain
- Acoustic classifier: $p(q^i \mid X)$
  - calculates probabilities of each mutually-exclusive state q^i
- 'Finite state acceptor' (i.e. HMM)

$$Q^* = \underset{\{q_0, q_1, \dots q_L\}}{\operatorname{argmax}} \; p(q_0, q_1, \dots q_L \mid X_0, X_1, \dots X_L)$$

  - MAP match of allowable sequence to probabilities:



Standard speech recognizer structure

- Fundamental equation of speech recognition:

$$Q^* = \operatorname*{argmax}_Q p(Q \mid X, \Theta)$$
$$= \operatorname*{argmax}_Q p(X \mid Q, \Theta) p(Q \mid \Theta)$$

  - X = acoustic features
  - $p(X \mid Q, \Theta)$ = acoustic model
  - $p(Q \mid \Theta)$ = language model
  - argmax_Q = search over sequences
- Questions:
  - what are the best features?
  - how do we do model them?
  - how do we find/match the state sequence?

Outline

Feature Calculation

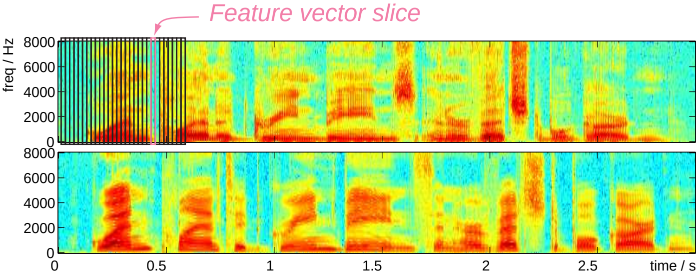
- Goal: Find a representational space most suitable for classification
  - waveform: voluminous, redundant, variable
  - spectrogram: better, still quite variable
  - ...?
- Pattern Recognition: representation is upper bound on performance
  - maybe we *should* use the waveform...
  - or, maybe the representation 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

Features (1): Spectrogram

- Plain STFT as features *e.g.*

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

- Consider examples:



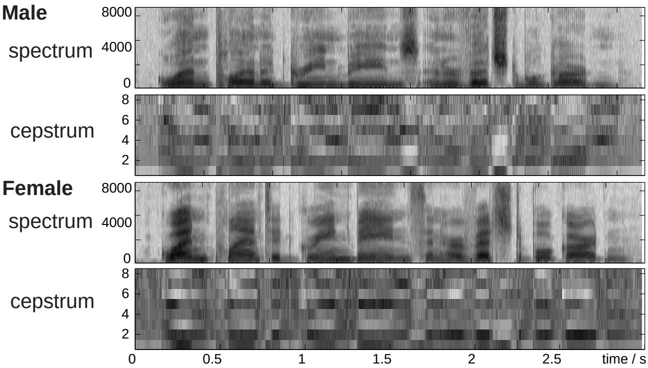
- Similarities between corresponding segments
  - but still large differences

Features (2): Cepstrum

- Idea: Decorrelate, summarize spectral slices:

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

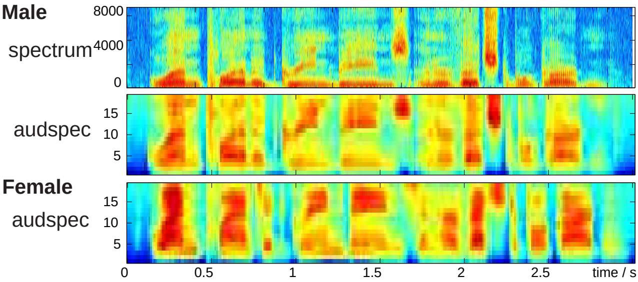
  - good for Gaussian models
  - greatly reduce feature dimension



Features (3): Frequency axis warp

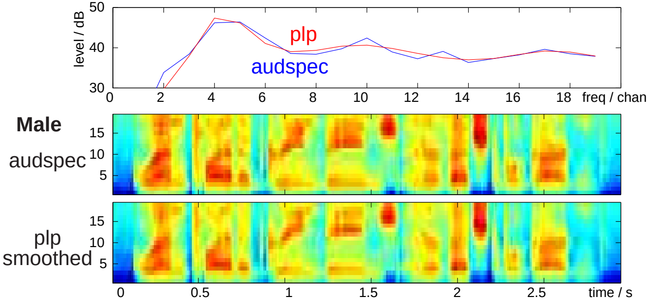
- Linear frequency axis gives equal 'space' to 0-1 kHz and 3-4 kHz
  - but perceptual importance very different
- Warp frequency axis closer to perceptual axis
  - mel, Bark, constant-Q ...

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



Features (4): Spectral smoothing

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



Features (5): Normalization along time

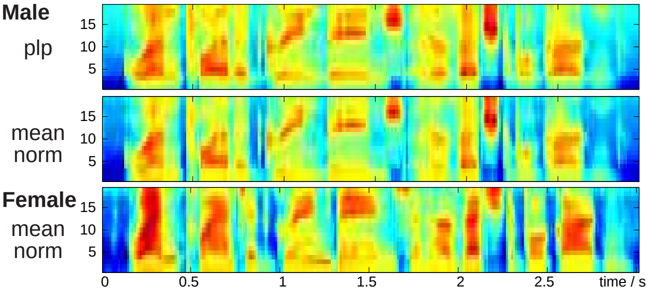
- Idea: feature variations, not absolute level
- Hence: calculate average level and subtract it:

$$\hat{Y}[n,k] = \hat{X}[n,k] - \underset{n}{\text{mean}}\{\hat{X}[n,k]\}$$

- Factors out fixed channel frequency response

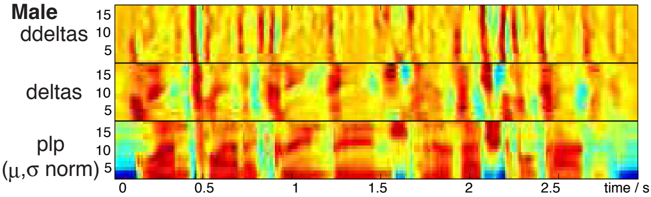
$$x[n] = h_c * s[n]$$

$$\hat{X}[n,k] = \log|X[n,k]| = \log|H_c[k]| + \log|S[n,k]|$$



Delta features

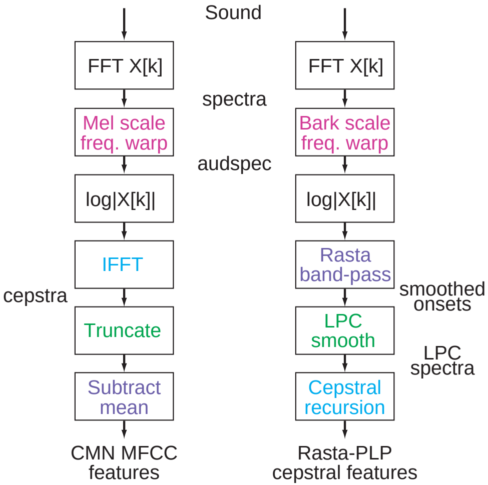
- Want each segment to have 'static' feature vals
  - but some segments intrinsically dynamic!
  - $\rightarrow$ calculate their derivatives—maybe steadier?
- Append dX/dt ($+\ d^2X/dt^2$) to feature vectors



- Relates to onset sensitivity in humans?

Overall feature calculation

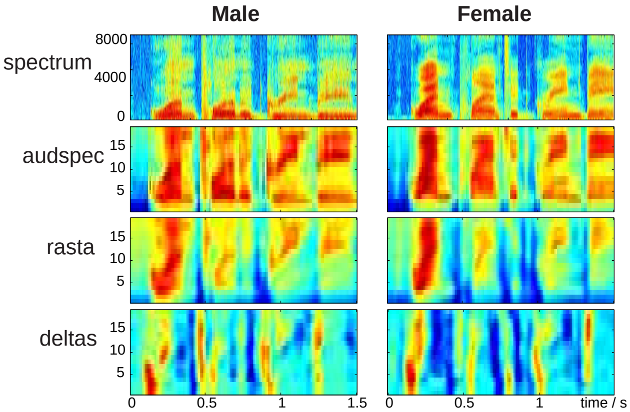
MFCCs and/or RASTA-PLP



Key attributes:

- spectral, auditory scale
- decorrelation
- smoothed (spectral) detail
- normalization of levels

Features summary

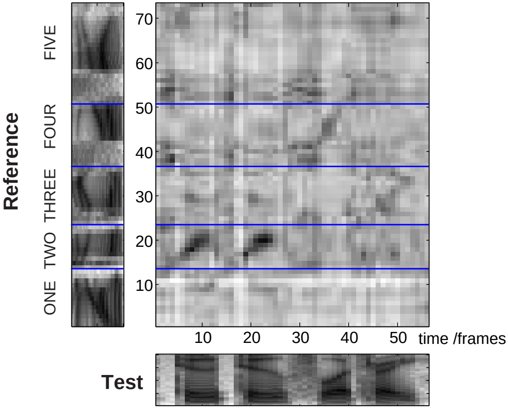


- Normalize same phones
- Contrast different phones

Outline

Sequence recognition: Dynamic Time Warp (DTW)

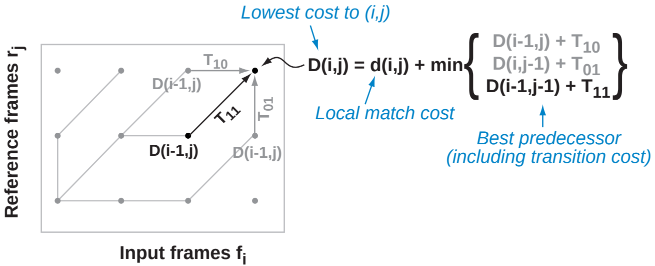
- Framewise comparison with stored templates:



  - distance metric?
  - comparison across templates?

Dynamic Time Warp (2)

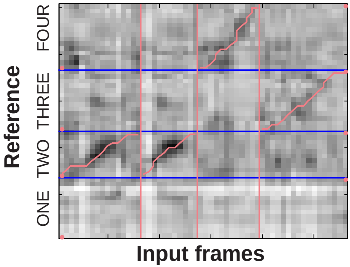
- Find lowest-cost constrained path:
  - matrix $d(i, j)$ of distances
    between input frame f_i and reference frame r_j
  - allowable predecessors and transition costs T_{xy}



- Best path via traceback from final state
  - store predecessors for each (i, j)

DTW-based recognition

- Reference templates for each possible word
- For isolated words:
  - mark endpoints of input word
  - calculate scores through each template (+prune)



  - continuous speech: link together word ends
- Successfully handles timing variation
  - recognize speech at reasonable cost

Statistical sequence recognition

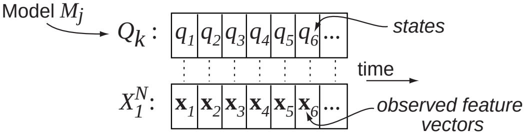
- DTW limited because it's hard to optimize
  - learning from multiple observations
  - interpretation of distance, transition costs?
- Need a theoretical foundation: Probability
- Formulate recognition as MAP choice among word sequences:

$$Q^* = \operatorname*{argmax}_Q p(Q \mid X, \Theta)$$

  - X = observed features
  - Q = word-sequences
  - Θ = all current parameters

State-based modeling

- Assume discrete-state model for the speech:
  - observations are divided up into time frames
  - model $\rightarrow$ states $\rightarrow$ observations:



- Probability of observations given model is:

$$p(X \mid \Theta) = \sum_{\text{all } Q} p(X_1^N \mid Q, \Theta)\, p(Q \mid \Theta)$$

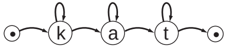
  - sum over all possible state sequences Q
- How do observations X_1^N depend on states Q?
- How do state sequences Q depend on model Θ?

HMM review

HMM is specified by parameters Θ:

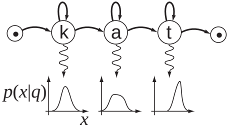
- states q^i

- transition probabilities a_{ij}



	k	a	t	•
•	1.0	0.0	0.0	0.0
k	0.9	0.1	0.0	0.0
a	0.0	0.9	0.1	0.0
t	0.0	0.0	0.9	0.1

- emission distributions $b_i(x)$



(+ initial state probabilities π_i)

$$a_{ij} \equiv p(q_n^j \mid q_{n-1}^i) \qquad b_i(x) \equiv p(x \mid q_i) \qquad \pi_i \equiv p(q_1^i)$$

HMM summary (1)

- HMMs are a generative model: recognition is inference of $p(Q \mid X)$
- During generation, behavior of model depends only on current state q_n:
  - transition probabilities $p(q_{n+1} \mid q_n) = a_{ij}$
  - observation distributions $p(x_n \mid q_n) = b_i(x)$
- Given states $Q = \{q_1, q_2, \ldots, q_N\}$
  and observations $X = X_1^N = \{x_1, x_2, \ldots, x_N\}$
- Markov assumption makes

$$p(X, Q \mid \Theta) = \prod_n p(x_n \mid q_n) p(q_n \mid q_{n-1})$$

HMM summary (2)

- Calculate $p(X \mid \Theta)$ via forward recursion:

$$p(X_1^n, q_n^j) = \alpha_n(j) = \left[\sum_{i=1}^{S} \alpha_{n-1}(i) a_{ij} \right] b_j(x_n)$$

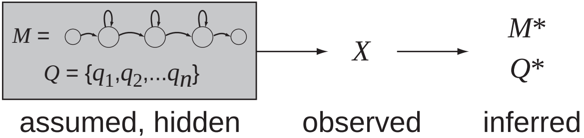
- Viterbi (best path) approximation

$$\alpha_n^*(j) = \left[\max_i \left\{ \alpha_{n-1}^*(i) a_{ij} \right\} \right] b_j(x_n)$$

  - then backtrace...

$$Q^* = \underset{Q}{\operatorname{argmax}}(X, Q \mid \Theta)$$

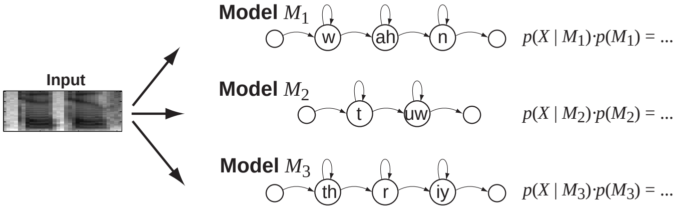
- Pictorially:



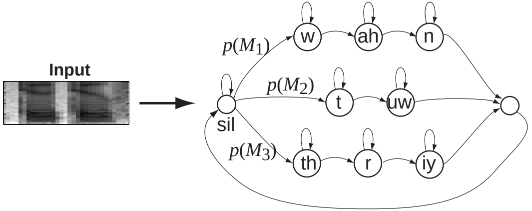
Outline

Recognition with HMMs

- Isolated word
  - choose best $p(M \mid X) \propto p(X \mid M)p(M)$



- Continuous speech
  - Viterbi decoding of one large HMM gives words

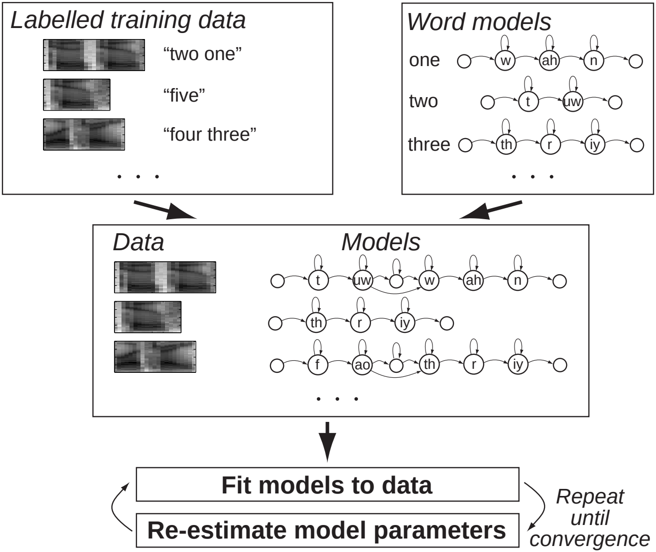


Training HMMs

- Probabilistic foundation allows us to train HMMs to 'fit' training data

  i.e. estimate a_{ij}, $b_i(x)$ given data
  - better than DTW...
- Algorithms to improve $p(\Theta \mid X)$ are key to success of HMMs
  - maximum-likelihood of models...
- State alignments Q for training examples are generally unknown
  - ... else estimating parameters would be easy
- Viterbi training
  - 'Forced alignment'
  - choose 'best' labels (heuristic)
- EM training
  - 'fuzzy labels' (guaranteed local convergence)

Overall training procedure



Language models

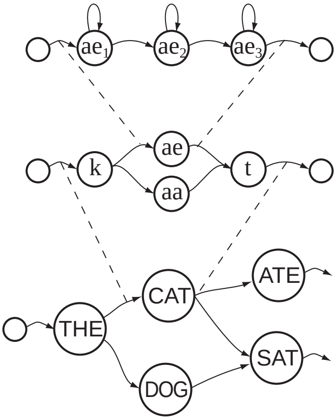
- Recall, fundamental equation of speech recognition

$$Q^* = \operatorname*{argmax}_Q p(Q \mid X, \Theta)$$
$$= \operatorname*{argmax}_Q p(X \mid Q, \Theta_A) p(Q \mid \Theta_L)$$

- So far, looked at $p(X \mid Q, \Theta_A)$
- What about $p(Q \mid \Theta_L)$?
  - Q is a particular word sequence
  - Θ_L are parameters related to the language
- Two components:
  - link state sequences to words $p(Q \mid w_i)$
  - priors on word sequences $p(w_i \mid M_j)$

HMM Hierarchy

- HMMs support composition
  - can handle time dilation, pronunciation, grammar all within the same framework



$$
\begin{aligned}
p(q \mid M) &= p(q, \phi, w \mid M) \\
&= p(q \mid \phi) \\
&\quad \cdot p(\phi \mid w) \\
&\quad \cdot p(w_n \mid w_1^{n-1}, M)
\end{aligned}
$$

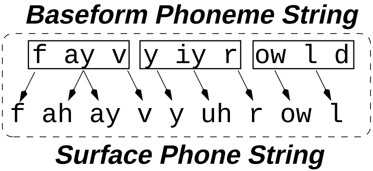
Pronunciation models

- Define states within each word $p(Q \mid w_i)$
- Can have unique states for each word ('whole-word' modeling), or . . .
- Sharing (tying) subword units between words to reflect underlying phonology
  - more training examples for each unit
  - generalizes to unseen words
  - (or can do it automatically. . .)
- Start *e.g.* from pronunciation dictionary:

```
ZERO(0.5)     z iy r ow
ZERO(0.5)     z ih r ow
ONE(1.0)      w ah n
TWO(1.0)      tcl t uw
```

Learning pronunciations

- ‘Phone recognizer’ transcribes training data as phones
  - align to ‘canonical’ pronunciations



  - infer modification rules
  - predict other pronunciation variants
- *e.g.* ‘d deletion’:

$$d \rightarrow \emptyset | \ell_{\text{stop}} \quad p = 0.9$$

- Generate pronunciation variants; use forced alignment to find weights

Grammar

- Account for different likelihoods of different words and word sequences $p(w_i \mid M_j)$
- 'True' probabilities are very complex for LVCSR
  - need parses, but speech often agrammatic

$\longrightarrow$ Use n-grams:

$$p(w_n \mid w_1^L) = p(w_n \mid w_{n-K}, \ldots, w_{n-1})$$

e.g. n-gram models of Shakespeare:

n=1	To him swallowed confess hear both. Which. Of save on ...
n=2	What means, sir. I confess she? then all sorts, he is trim, ...
n=3	Sweet prince, Falstaff shall die. Harry of Monmouth's grave...
n=4	King Henry. What! I will go seek the traitor Gloucester. ...

- Big win in recognizer WER
  - raw recognition results often highly ambiguous
  - grammar guides to 'reasonable' solutions

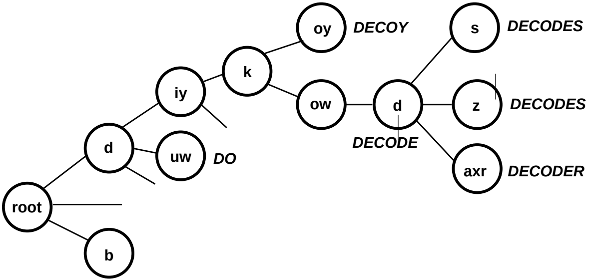
Smoothing LVCSR grammars

- n-grams ($n = 3$ or 4) are estimated from large text corpora
  - 100M+ words
  - but: not like spoken language
- 100,000 word vocabulary $\rightarrow$ 10^{15} trigrams!
  - never see enough examples
  - unobserved trigrams should NOT have $Pr = 0$!
- Backoff to bigrams, unigrams
  - $p(w_n)$ as an approx to $p(w_n \mid w_{n-1})$ etc.
  - interpolate 1-gram, 2-gram, 3-gram with learned weights?
- Lots of ideas *e.g.* category grammars
  - $p(\text{PLACE} \mid \text{“went”}, \text{“to”})p(w_n \mid \text{PLACE})$
  - how to define categories?
  - how to tag words in training corpus?

Decoding

- How to find the MAP word sequence?
- States, pronunciations, words define one big HMM
  - with 100,000+ individual states for LVCSR!

$\rightarrow$ Exploit hierarchic structure
  - phone states independent of word
  - next word (semi) independent of word history



Decoder pruning

- Searching 'all possible word sequences'?
  - need to restrict search to most promising ones: beam search
  - sort by estimates of total probability
    $= Pr(\text{so far}) +$ lower bound estimate of remains
  - trade search errors for speed
- Start-synchronous algorithm:
  - extract top hypothesis from queue:
    $$[Pn, \quad \{w_1, \ldots, w_k\}, \quad n]$$
    pr. so far — words — next time frame
  - find plausible words $\{w_i\}$ starting at time $n \rightarrow$ new hypotheses:
    $$[P_n p(X_n^{n+N-1} \mid w^i) p(w^i \mid w_k \ldots), \quad \{w_1, \ldots, w_k, w^i\}, \quad n+N]$$
  - discard if too unlikely, or queue is too long
  - else re-insert into queue and repeat

Summary

- Speech signal is highly variable
  - need models that absorb variability
  - hide what we can with robust features
- Speech is modeled as a sequence of features
  - need temporal aspect to recognition
  - best time-alignment of templates = DTW
- Hidden Markov models are rigorous solution
  - self-loops allow temporal dilation
  - exact, efficient likelihood calculations
- Language modeling captures larger structure
  - pronunciation, word sequences
  - fits directly into HMM state structure
  - need to 'prune' search space in decoding

Parting thought

Forward-backward trains to generate, can we train to discriminate?

References

Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

Mehryar Mohri, Fernando Pereira, and Michael Riley. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88, 2002.

Wendy Holmes. *Speech Synthesis and Recognition*. CRC, December 2001. ISBN 0748408576.

Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of Speech Recognition*. Prentice Hall PTR, April 1993. ISBN 0130151572.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall, January 2000. ISBN 0130950696.

Frederick Jelinek. *Statistical Methods for Speech Recognition (Language, Speech, and Communication)*. The MIT Press, January 1998. ISBN 0262100665.

Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Prentice Hall PTR, April 2001. ISBN 0130226165.