Speech Analysis and Interpretation Lab

Overview of Speech Recognition

- ASR:
  - Front End
    - Signal Processing
    - Features & motivation from biology
  - Pattern Matching
    - Basics of Probability Theory
    - Markov Models
    - Hidden Markov Models in speech
  - Creating the database of patterns
  - More probabilities & more Bayes:
    - Language Modeling (the holy grail of DM?)
- The SAIL Lab
  - Examples of where DM is needed!

Speech Recognition Basics

Signal Processing

Signal Acquisition

- Speech is captured by a microphone.
- The analog signal is converted to a digital signal by sampling it 16000 times a second.
- Each value is quantized to 16 bits, a number between -32768 and 32767.

Front End

- Need to find the features of the speech sound
- What are the features:
  - derived from knowledge of the workings of the ear=perception or the production or modeling of resonances etc
  - EAR: Inner Hair Cells tough to model...
Front End

- The ear is divided into three parts: the external ear, the middle ear and the inner ear. The external ear collects sound, the middle ear mechanism transforms the sound and the inner ear receives and transmits the sound.
- Sound vibrations enter the ear canal and cause the eardrum to vibrate. Movements of the eardrum are transmitted across the middle ear to the inner ear fluids by three small ear bones. These middle ear bones (hammer or malleus, anvil or incus and stirrup or stapes) act as a transformer changing sound vibrations in air into fluid waves in the inner ear. The fluid waves stimulate delicate nerve endings in the hearing canals. Electrical impulses are transmitted on the nerve to the brain where they are interpreted as understandable sound.

Spectral Analysis

- Take a **window** of 20-30 ms (320 to 480 samples).
- **FFT** (Fast Fourier Transform) used to compute energy in several frequency bands, typically 20 to 40.
- **Cepstrum** computed, 12 or so coefficients.

Production

- **Tube model:**

  - (1) Nasal cavity,
  - (2) Hard palate,
  - (3) Alveolar ridge,
  - (4) Soft palate (Velum),
  - (5) Tip of the tongue (Apex),
  - (6) Dorsum,
  - (7) Uvula,
  - (8) Radium,
  - (9) Pharynx,
  - (10) Epiglottis,
  - (11) False vocal cords,
  - (12) Vocal cords,
  - (13) Larynx,
  - (14) Esophagus, and
  - (15) Trachea.

Front End

- Remove data keeping information intact
- Periodic excitation (air through the lungs) contains little lexical information

Spectral Analysis

- **A frame**, a snapshot of the input’s spectrum, is represented by a vector of 12 cepstrum coefficients.
- To describe the spectral change with time, one frame is computed every 10 ms or so, *i.e.* 100 frames per second.
- A word that lasts 1 second is described by 1200 numbers.
Discrete Random Variables

- A is a Boolean-valued random variable if A denotes an event, and there is some degree of uncertainty as to whether A occurs.
- Examples
  - A = The US president in 2023 will be male
  - A = You wake up tomorrow with a headache
  - A = You are sick

Probabilities

- Probability of A = P(A) is “the fraction of possible worlds in which A is true”

Axioms visually

- P(A or B) = P(A) + P(B) – P(A and B)

Conditionals

- Example: Rain & Cloudy in Los Angeles
  - There is a small chance of Rain and a slightly larger chance of Cloudy
  - The chances of Cloudy when it is Raining are very large
  - P(C|R)=P(C and R) / P(R)
  - Probability we are in the cyan=cloud given we are in the red=rain is the area of the overlap normalized by the area of the red=rain
Bayes Theorem

- Published after his death

\[
P(C|R) = P(C \text{ and } R) / P(R)
\]

\[
P(R|C) = P(C \text{ and } R) / P(C)
\]

\[
P(C \text{ and } R) = P(C|R) P(R) = P(R|C) P(C)
\]

Weather predictor example of a Markov model

Weather predictor calculation

Given today is sunny (i.e., \( x_1 = 3 \)), what is the probability of “sun-rain-cloud-cloud-sun” with model \( \mathcal{M} \)?

\[
P(X|M) = P(X = (3,3,1,2,2)|M)
\]

\[
= P(x_1 = 3) P(x_2 = 3|x_1 = 3)
\]

\[
P(x_3 = 1|x_2 = 3) P(x_4 = 2|x_3 = 1)
\]

\[
P(x_5 = 2|x_4 = 2) P(x_6 = 3|x_5 = 2)
\]

\[
= \pi_3 a_{33} a_{31} a_{12} a_{22} a_{23}
\]

\[
= 1 \cdot (0.8)(0.1)(0.3)(0.6)(0.2)
\]

\[
= 0.00288
\]

where the initial state probability for state \( i \) is

\[
\pi_i = P(x_1 = i).
\]

Summary of Markov models

Transition probabilities:

\[
A = \{a_{ij}\} = \begin{bmatrix}
0.6 & 0.4 & 0 & 0 \\
0 & 0.9 & 0.1 & 0 \\
0 & 0 & 0.2 & 0.8 \\
0 & 0 & 0 & 0.5
\end{bmatrix}
\]

and

\[
\pi = \{\pi_i\} = \begin{bmatrix} 1 \end{bmatrix}
\]

Probability of a given state sequence \( X \):

\[
P(X|M) = \pi_{x_1} a_{x_1 x_2} a_{x_2 x_3} a_{x_3 x_4} \cdots
\]

\[
= \pi_{x_1} \prod_{i=2}^{n} a_{x_{i-1} x_i}.
\]
Hidden Markov Models

**Elements of a discrete HMM, $\lambda$**

1. Number of states $N$, $x \in \{1, \ldots, N\}$;
2. Number of events $K$, $k \in \{1, \ldots, K\}$;
3. Initial-state probabilities, $\pi = \{\pi_i\} = \{P(x_1 = i)\}$ for $1 \leq i \leq N$;
4. State-transition probabilities, $A = \{a_{ij}\} = \{P(x_t = j|x_{t-1} = i)\}$ for $1 \leq i, j \leq N$;
5. Discrete output probabilities, $B = \{b_i(k)\} = \{P(o_t = k|x_t = i)\}$ for $1 \leq i \leq N$ and $1 \leq k \leq K$.

**Hidden Markov model example**

- $P(O|X, \lambda) = b_1(o_1) b_2(o_2) b_3(o_3) b_4(o_4) b_5(o_5)$
- $P(X|\lambda) = \pi_1 a_{11} a_{12} a_{23} a_{34} a_{45}$
- $P(O, X|\lambda) = \pi_1 b_1(o_1) a_{11} b_2(o_2) a_{12} b_3(o_3) \ldots$ (17)
- $P(O, X|\lambda) = \pi_1 b_1(o_1) a_{11} b_1(o_2) a_{12} b_2(o_3) \ldots$ (18)

Hidden Markov Models (HMM)

**Isolated digit recognition**

- 10 templates: one template $M_i$ per digit.
- Compare input $x$ with all templates.
- Select the most similar template $j$:
  
  $j = \min \{ f(x, M_j) \}$
  
  where $f$ is the comparison function.
Probability of a path

\[ P = \prod_{i=1}^{n} \left( a_{x_i} b_{x_i} a_{i2} b_{i2} a_{i3} b_{i3} a_{i4} b_{i4} a_{i5} b_{i5} \right) \]

Is the probability of the best path.

Problem: We need to evaluate all possible paths, and there grow exponential with number of states and input frames.

Solution: The Viterbi algorithm has complexity that grows linearly with number of states and input frames.

Large vocabulary

- Problem:
  - Huge number of models: 60,000 models
  - Models are hard to “train”
  - Cannot easily add new words.
- Solution: Use phoneme models

Context Dependent Models

- To improve accuracy, phone models are context-dependent.
- Example: THIS = DH IH S

Continuous speech

- Example: THIS IS A TEST

\[ \alpha_t(i) = P(x_1, x_2 \ldots x_t, s_t = i \mid \lambda) \]

\[ \alpha_{t+1}(j) = \sum_{i=1}^{N} \alpha_t(i) a_{ij} b_j(x_{t+1}) \]

Baum-Welch Algorithm

\[ P(X \mid \lambda) = \sum_{i=1}^{N} \alpha_t(i) \]

Exact solution.
Speeding Search: Pruning

- Viterbi algorithm: replace sum with max.
- Speed-Accuracy Tradeoff: Computations can be reduced by eliminating paths that are “not promising”, at the expense of having a chance of eliminating the “best” path.
- Typically, computations can be reduced by more than a factor of 10, without affecting the error rate significantly.

HMM Training

- A lot of speech is needed to train the models. Done with an iterative algorithm:
  1. Take initial model (i.e. uniform probabilities).
  2. Segment Database (Run Viterbi algorithm).
  3. Update models.
     \[ \hat{\mu}_j = \frac{\sum_{i=1}^{N} x_{ij}}{N} \quad \hat{\sigma}^2_j = \frac{\sum_{i=1}^{N} (x_{ij} - \hat{\mu}_j)^2}{N} \]
  4. If Converged stop, otherwise go to 2.

Bayes Rule in ASR

\[ P(W \mid A) = \frac{P(A \mid W)P(W)}{P(A)} \]
\[ \hat{W} = \max P(W \mid A) = \max P(A \mid W)P(W) \]
where \( A \) is the Acoustics and \( W \) the sequence of words. \( P(W) \) is the language model.
**Statistical Language Model**

\[
P(W) = P(W_1W_2\cdots W_n) = P(W_1)P(W_2|W_1) \\
P(W_2|W_1)P(W_3|W_2\cdots)P(W_n|W_{n-1}\cdots) = P(W_i)P(W_{i+1}|W_i) \cdots P(W_n|W_{n-1})
\]

- **Unigram**
- **Bigram**
- **Trigram**

**Trigrams**

\[
P(\text{THIS IS A TEST}) = P(\text{THIS}) P(\text{IS|THIS}) P(\text{A|THIS IS}) P(\text{TEST|IS A})
\]

- These statistical language models predict the probability of the current word given the past history.

- While simplistic, they contain a lot of information

\[
P(\text{IS|THIS}) >> P(\text{IS|HAVE})
\]

**N-Gram Performance**

<table>
<thead>
<tr>
<th>Error Rate</th>
<th>Multiplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>x5.0</td>
<td></td>
</tr>
<tr>
<td>x3.0</td>
<td></td>
</tr>
<tr>
<td>x1.5</td>
<td></td>
</tr>
<tr>
<td>x1.0</td>
<td></td>
</tr>
</tbody>
</table>

**Perplexity**

- Perplexity measures the average branching of a text when presented to a language model. \( PP = 2^{\frac{1}{P}} = \hat{P}(W_1, W_2, \cdots W_n)^{-\frac{1}{n}} \)

- An empirical observation \( E = k \sqrt{P} \) where \( E \) is the error rate and \( P \) the perplexity.

- A language model with low perplexity is good for ASR.

- Tradeoff between perplexity and coverage.

**Context Free Grammar**

- **Advantages:**
  - Low error rate (because of low perplexity).
  - Compact.
  - Easy for application developers.

- **Disadvantages:**
  - Poor coverage (out of language sentences).

**Applications**
**ASR problem Space**

- Error Rate Multiplier
- Unconstrained Task
- Noisy Environment
- Large Vocabulary
- Speaker Variance
- More Constraint
- Less Constraint

**ASR Accuracy in the Lab**

- **Command & Control (300 words + CFG)**
  - 1% word error rate (speaker-independent)
- **Discrete Dictation (60K words)**
  - 3% word error rate (speaker-dependent)
- **Continuous Dictation (60K words)**
  - 7% word error rate (speaker-dependent)
- **Telephone Digits**
  - 0.3% word error rate (speaker-independent)

**C & C Types of Errors**

- Pronunciation errors make some words *unrecognizable*.
- User says something outside the vocabulary/grammar.
- User speaks while machine is talking.
- Background noise fires recognizer (false alarm). Rejection needs improvement.
- Spontaneous speech (uhms, ahms)

**ASR in Real Applications**

- New word addition
- Rejection
- Barge-in.
- Robustness to noise.
- ASR is just a part of an application, User Interface is critical.

**Interface NLP/ASR**

- Loose coupling: NLP selects correct input from a list of top N candidate sentences:
  - Easy to implement, not optimal
- Tight coupling. NLP provides the language probabilities for the search:
  - Optimal but hard to implement.
  - Can a NLP system reduce perplexity?
**Speech Analysis and Interpretation Lab**

- Human communication involves a complex orchestration of cognitive, physiological, physical, social processes
  - Sophisticated human production-perception chain: can learn and adapt
  - Inherently multimodal: natural sensory communication involves speech, gestures, touch: Spoken language plays a key role
- Speech signal carries crucial information: intent, desires, emotions
  - Nonlinear progression and unfolding of events that can be marked with linguistic and extra-linguistic tags

**Labs Mission**

- Implementing the multi-level mapping: deciphering what, who and how
  - “what”: speech content, the primary object of speech recognition
    - Robustly map speech signal to words
  - “who”: speech source, the primary object of speaker recognition biometrics
    - Map speech and other cues to verify or determine speaker identity
  - “how”: speech style and emotions, object of expressive speech processing (analysis, synthesis and recognition)
    - How – literary assessment
    - Emphasis, contrast of new, important information (linguistic)
      - Boundary information: utterance, phrase, word,
      - Speech rate information
      - Content word-function word distinction
      - Prominence detection
  - User state such as frustration, attention and emotions such as anger and sadness

**Conversational Systems**

- Information resides at multiple time scales, through multiple cues
  - Time-frequency modulation of speech signal not completely localized: interactions across multiple events, synchronous and asynchronous
  - Information contained at various levels of linguistic abstraction: neuro cognitive, articulatory, acoustic lexical, syntactic, semantic, discourse

**Conversational Systems – Translation**

- Potential broad impact in
  - Hospitals: Especially Spanish for the Latino population in US
  - Refugee processing

**Multimodal Sensing**

- User Sensing
- Explicit User Input
- Implicit Cues
  - User Interface
  - ASR
  - Predictive Module
  - Dialog Management
  - Synthesis
  - Meta Features
**Total Communication**
- Domain specific Spoken language translation
- Cross-cultural communication
- Rapid deployment for:
  - new language pairs
  - new domains
- Co-evolution of technology after deployment (inc. field adaptation & extension of functionality, and flow-back of data from field to development centers)
- Exploit Domain Knowledge – Agent - for improved performance and aiding user
- Employ Translation for Cultural & language training

**Scalable Immersive Environments (IMSC & SAIL)**
- Sense the user:
  - Identity of speaker
  - Location of speaker
  - Recognition of speech
  - Emotion recognition
  - Participant locations
- and working on:
  - Gaze recognition
  - Engagement ....
- Low latency, HD streaming
- Summarization
- ....

Note: Lectures put together from own material as well as material from tutorials of Alex Acero & Philips Jackson