Crash Course in ASR (An attempt)

Automatic Speech Recognition (ASR)

Marco Siniscalchi, Ph.D.
Kore University of Enna
94100 Enna, Italy
Lecture Outline

• The Speech Dialog Circle
  – System components

• Automatic Speech Recognition (ASR)
  – System components and technology dimensions
  – Implementation issues

• Theory of Markov Models
  – Discrete Markov processes
  – Hidden Markov processes

• Solutions to the Three Basic Problems of HMM’s
  – Computation of observation probability
  – Determination of optimal state sequence
  – Optimal training of model
Voice-Enabled System Technology Components

**Components:**
- **TTS (Text-to-Speech Synthesis)**
- **ASR (Automatic Speech Recognition)**
- **SLG (Spoken Language Generation)**
- **SLU (Spoken Language Understanding)**
- **DM (Dialog Management)**

**Processes:**
- **Speech to Speech**
- **Words to Words**
- **Action to Meaning**

**Core Functions:**
- **Data, Rules**
- **Text-to-Speech Synthesis**
- **Automatic Speech Recognition**
- **Spoken Language Generation**
- **Spoken Language Understanding**

**Integration:**
- **Dialog Management**
The Speech Dialog Circle

Voice reply to customer

“Which date do you ...?”

Text-to-Speech Synthesis

Words: What’s next?

“Which date do you want to fly from Catania to Roma?”

Spoken Language Generation

Data, Rules

Action

“Get departure dates”

Dialog Management

“ORIGIN: Catania
DESTINATION: Roma
FLIGHT_TYPE: ROUNDTRIP”

Speech

Automatic Speech Recognition

Words spoken

“I need a flight from Catania to Roma roundtrip”

Spoken Language Understanding

Meaning

“ORIGIN: Catania
DESTINATION: Roma
FLIGHT_TYPE: ROUNDTRIP”
**Automatic Speech Recognition (ASR)**

**Goal:** Accurately and efficiently convert a speech signal into a text message independent of the device, speaker or the environment.

**Applications:** Automation of complex operator-based tasks, e.g., customer care, dictation, form filling applications, provisioning of new services, customer help lines, e-commerce, etc.
A Brief History of Speech Recognition

Cristopher C. discovers a new continent.

Accuracy

Speech Recognition

Time

Mathematical Formalization
Global optimization
Automatic Learning from data

Heuristics
Handcrafted rules
Local optimization

HMMs
Milestones in ASR Technology Research

1962
- Filter-bank analysis;
- Time-normalization;
- Dynamic programming

1967
- Isolated Words;
- Connected Digits;
- Continuous Speech

1972
- Pattern recognition;
- LPC analysis;
- Clustering algorithms;
- Level building

1977
- Hidden Markov models;
- Stochastic Language modeling

1982
- Stochastic language understanding;
- Finite-state machines;
- Statistical learning

1987
- Concatenative synthesis;
- Machine learning;
- Mixed-initiative dialog

1992
- Very Large Vocabulary; Semantics, Multimodal Dialog, TTS

1997
- Spoken dialog; Multiple modalities

2002
- Large Vocabulary; Speech Understanding

Year

Isolated Word Recognition (IWR)

- Computing word score using a sequence of phone scores

Please say the isolated command now.

EDtv

ants → Score = 12.2

Ants

edtv → Score = 32.5

EDtv

payback → Score = 29.4

Payback
Training Speech Recognizer

This is a test.

Thousands of training samples are combined to build 40 sub-word models, one for each phoneme.
Automatic Speech Recognition (ASR)

Concept: a sequence of symbols

Parameterise

Speech Waveform

Speech Vectors

Recognise

S1, S2, S3, etc.
ASR System Components

- Feature Extraction
  - Framing and short-time spectral/cepstral analysis
- Acoustic Modeling of Speech Units
  - Fundamental speech unit selection
  - Statistical pattern matching (HMM unit) modeling
- Lexical Modeling
  - Pronunciation network
- Syntactic and Semantic Modeling
  - Deterministic or stochastic finite state grammar
  - N-gram language model
- Search and Decision Strategies
  - Best-first or depth-first, DP-based search
  - Modular vs. integrated decision strategies
ASR Terminology

• Vocabulary
  – Words that can be recognized in an application
  – More words imply more errors and more computation
• Word spotting
  – Listening for a few specific words within an utterance
• Extraneous speech screening (rejection)
  – Capability to decide whether a candidate key word is a close enough match to be declared a valid key word
• Grammars
  – Syntax (word order) that can be used
  – The way words are put together to form phrases & sentences, some are more likely than others
  – Can be deterministic or stochastic
• Semantics
  – Usually not properly modeled or represented
ASR System and Technology Dimensions

- Isolated word vs. continuous speech recognition
  - Isolated = pauses required between each word
  - Continuous = no pauses required
- Speech unit selection: whole vs. sub-word
  - Whole word = requires data collection of the words to be recognized
  - Sub-word = recognizing basic units of speech (phonemes), so word recognition is easier for new applications
- Read vs. spontaneous (degree of fluency)
- Small vs. medium or large vocabulary
- Multilingual vs. dialect/accent variations
Basic ASR Formulation (Bayes Method)

Speaker's Intention \( w \) → Speech Production Mechanisms \( s(n) \) → Acoustic Processor \( X \) → Linguistic Decoder \( \hat{W} \)

Speaker Model

\[ \hat{W} = \arg \max_w P(W \mid X) \]

\[ = \arg \max_w \frac{P(X \mid W)P(W)}{P(X)} \]

\[ = \arg \max_w P_A(X \mid W)P_L(W) \]

Step 3

Step 1

Step 2
Steps in Speech Recognition

Step 1- acoustic modeling: assign probabilities to acoustic realizations of a sequence of words. Compute $P_A(X|W)$ using hidden Markov models of acoustic signals and words.

Step 2- language modeling: assign probabilities to sequences of words in the language. Train $P_L(W)$ from generic text or from transcriptions of task-specific dialogues.

Step 3- hypothesis search: find the word sequence with the maximum a posteriori probability. Search through all possible word sequences to determine $\text{arg max over } W$. 
Step 1-The Acoustic Model

We build acoustic models by learning statistics of the acoustic features, \( X \), from a training set where we compute the variability of the acoustic features during the production of the sounds represented by the models.

It is impractical to create a separate acoustic model, \( P_A(X | W) \), for every possible word in the language--it requires too much training data for words in every possible context.

Instead we build acoustic-phonetic models for the \(~50\) phonemes in the English language and construct the model for a word by concatenating (stringing together sequentially) the models for the constituent phones in the word.

We build sentences (sequences of words) by concatenating word models.
Step 2-The Language Model

• The language model describes the probability of a sequence of words that form a valid sentence in the language.

• A simple statistical method works well based on a Markovian assumption, namely that the probability of a word in a sentence is conditioned on only the previous N-words, namely an N-gram language model.

\[ P_L(W) = P_L(w_1, w_2, \ldots, w_k) \]

\[ = \prod_{n=1}^{k} P_L(w_n \mid w_{n-1}, w_{n-2}, \ldots, w_{n-N}) \]

where \( P_L(w_n \mid w_{n-1}, w_{n-2}, \ldots, w_{n-N}) \) is estimated by simply counting up the relative frequencies \( f \) of N-tuples in a large corpus of text.
**Step 3-The Search Problem**

- The **search** problem is one of searching the space of all valid sound sequences, conditioned on the word grammar, the language syntax, and the task constraints, to find the word sequence with the maximum likelihood.

- The **size of the search space** can be astronomically large and take inordinate amounts of computing power to solve by heuristic methods.

- The use of methods from the field of **Finite State Automata Theory** provide **Finite State Networks (FSN)** that reduce the computational burden by orders of magnitude, thereby enabling exact solutions in computationally feasible times, for large ASR problems.
Speech Recognition Processes (I)

- **Choose the task** => sounds, word vocabulary, task syntax (grammar), task semantics

- **Example**: isolated digit recognition task
  - Sounds to be recognized—whole words
  - Word vocabulary—zero, oh, one, two, …, nine
  - Task syntax—any digit is allowed
  - Task semantics—sequence of isolated digits must form a valid telephone number
Speech Recognition Processes (II)

• **Train the Models** => create a method for building **acoustic word models** from a speech training data set, a **word lexicon**, a **word grammar** (language model), and a **task grammar** from a text training data set
  - Speech training set— e.g., 100 people each speaking the 11 digits 10 times in isolation
  - Text data training set— e.g., listing of valid telephone numbers (or equivalently algorithm that generates valid telephone numbers)
Speech Recognition Processes (III)

- **Evaluate performance** => determine word error rate, task error rate for recognizer
  - Speech testing data set—25 new people each speaking 10 telephone numbers as sequences of isolated digits
  - Evaluate digit error rate, phone number error rate
- **Testing algorithm** => method for evaluating recognizer performance from the testing set of speech utterances
Speech Recognition Process

Input Speech

Feature Analysis (Spectral Analysis)

\( s(n), W \)

Language Model (N-gram)

\( X_n \)

Pattern Classification (Decoding, Search)

Acoustic Model (HMM)

\( W \)

Confidence Scoring (Utterance Verification)

"Hello World" (0.9) (0.8)
Goal: extract a representation for speech sounds. Ideally, speech features are discriminative, robust, and parsimonious.

Method: spectral analysis through either a bank-of-filters or through LPC followed by non-linearity and normalization (cepstrum).

Result: signal compression where for each window of speech samples where 30 or so cepstral features are extracted (64,000 b/s -> 5,200 b/s).

Challenges: robustness to environment (office, airport, car), devices (speakerphones, cellphones), speakers (acents, dialect, style, speaking defects), noise and echo. Feature set for recognition—cepstral features or those from a high dimensionality space.
What Features to Use?

• **Short-time Spectral Analysis:**
  
  **Acoustic features:**
  - Mel cepstrum (LPC, filterbank, wavelets), energy
  - Formant frequencies, pitch, prosody

  **Acoustic-Phonetic features:**
  - Manner of articulation (e.g., stop, nasal, voiced)
  - Place of articulation (e.g., labial, dental, velar)

  **Articulatory features:**
  - Tongue position, jaw, lips, velum

  **Auditory features:**
  - Ensemble interval histogram (EIH), synchrony

• **Temporal Analysis:** approximation of the velocity and acceleration typically through first and second order central differences.
Feature Extraction Process

Sampling and Quantization

\( s(t) \) \( \rightarrow \) \( s(n) \)
\( \alpha \) \( \rightarrow \) \( s'(n) \)

Preemphasis

Segmentation (blocking)

\( M, N \) \( \rightarrow \) \( s''(n) \)

Windowing

\( W(n) \) \( \rightarrow \) \( s'_m(n) \)

Energy Zero-Crossing

\( s'_m(n) \) \( \rightarrow \) \( s''_m(n) \)

Spectral Analysis

Noise Removal, Normalization

\( a'_m(l) \) \( \rightarrow \) \( a''_m(l) \)

Filtering

Cepstral Analysis

Pitch Formants

\( c'_m(l) \) \( \rightarrow \) \( c''_m(l) \)

Equalization

Bias removal or normalization

\( \Delta c'_m(l) \)

Temporal Derivative

Delta cepstrum Delta^2 cepstrum

Energy Zero-Crossing
ASR: Statistical Methodology

• We assume that the speech features are generated according to the probability models of the underlying linguistic units.

• During the training phase, the model parameters are learned from labeled data (features).

• During the testing phase, the learned models are used to find the hypothesis with the maximum a posteriori (MAP) probability given speech features.
ASR Solution

\[ \hat{W} = \arg \max_{W \in \Gamma} \bar{P}_\Omega(W) \cdot \bar{p}_\Lambda(X | W) \]

\( \bar{P}_\Lambda(X|W) \) — Acoustic Model (AM): gives the probability of generating feature \( X \) when \( W \) is uttered
- Need a model for every \( W \) to model all speech features from \( W \) à HMM is an ideal model for speech units
  - Sub-word unit is more flexible (better)

\( \bar{P}_\Omega(W) \) — Language Model (LM): gives the probability of \( W \) (word, phrase, sentence) is chosen to say.
- Need a flexible model to calculate the probability for all kinds of \( W \) à Markov chain model (\( n \)-gram)
Acoustic Model

- **Goal:** map acoustic features into distinct phonetic labels (e.g., /s/, /aa/) or word labels.

- **Hidden Markov Model (HMM):** statistical method for characterizing the spectral properties of speech by a parametric random process. A collection of HMMs is associated with a phone or a word. HMMs are also assigned for modeling extraneous events.

- **Advantages:** powerful statistical method for dealing with a wide range of data and reliably recognizing speech.

- **Challenges:** understanding the role of classification models (ML Training) versus discriminative models (MMI training). What comes after the HMM—are there data driven models that work better for some or all vocabularies.
Markov Models

• A Markov model consists of a set of states, an initial distribution, and a transition probability matrix

• Two model assumptions
  • The probability of state $S_t$ given $S_{t-1}$ is independent of any state prior to $(t-1)$
    \[ p(S_t | S_{t-1}, S_{t-2}, \ldots, S_1) = p(S_t | S_{t-1}) \]
  • The transition probability does not vary with $t$
    \[ p(S_t = j | S_{t-1} = i) = a_{ij} \text{ for all } t \]
Hidden Markov Models (HMMs)

- In an HMM, the state identities are hidden and the observed sequences depend probabilistically on the state sequence.
- In addition to the components required in a Markov model, in HMM there are the observation likelihoods, denoted by $b_i(X_t)$, representing the probability of observing $X_t$ when the state $S_t = i$. 

Coin-toss Models

- Suppose there are a number of coins, each with its own bias.
- One of the coins, coin $S_t$, is randomly selected. The selection probability is dependent on the identity of the previous coin, $S_{t-1}$.
- Coin $S_t$ is tossed and the outcome (head or tail) $X_t$ is recorded, but not the coin.
HMM: An Ideal Speech Model

- Variations in speech signals: temporal & spectral
- Each state represents a process of measurable observations
- Inter-process transition is governed by a finite state Markov chain
- Processes are stochastic and individual observations do not immediately identify the hidden state.

HMM models spectral and temporal variations simultaneously
In a typical system, each phoneme in the language is modeled by a 3-state left-to-right continuous density HMM (CDHMM), and background noise is modeled by a 1-state CDHMM.

Up to thousand of hours of speech data have been used to train HMM’s.
HMM for Speech

- Phone model: ‘s’

- Word model: ‘is’ (ih-z)

\[ \begin{array}{c}
\text{i1} \\
\text{s1} \\
\text{s2} \\
\text{s3} \\
\end{array} \]

\[ \begin{array}{c}
\text{i1} \\
\text{i1} \\
\text{i2} \\
\text{i3} \\
\text{s1} \\
\text{s2} \\
\text{s3} \\
\end{array} \]

‘ih’  ‘z’
Isolated Word HMM

Left-right HMM – highly constrained state sequences
Basic Problems in HMMs

• Given acoustic observation \( X \) and model \( \Phi \):

**Evaluation:** Given the observation \( X \) and the model parameters \( \Phi \), \( P(X | \Phi) \)

**Decoding:** choose optimal state sequence, that is, given the observation \( X \) and \( \Phi \), determine the optimal state sequence \( S^* \)

\[
S^* = \arg \max_S p(X, S | \Phi)
\]

**Re-estimation:** adjust \( \Phi \) to maximize \( P(X | \Phi) \)
Evaluation: Forward-Backward Algorithm

- Denote the parameters in HMM by
  - The initial probability \( \pi_i = a_{1i} \)
  - The transition probability \( a_{ij} \)
  - The observation probability \( b_i(X_t) \)
Definition of the forward probability $\alpha$ as

$$\alpha_i(t) = p(X_1, ..., X_t, S_t = i)$$

Then

$$\alpha_j(1) = a_{1j}b_j(X_1)$$

$$\alpha_j(t) = \sum_{i=2}^{N-1} \alpha_i(t-1)a_{ij}b_j(X_t),$$

$$\alpha_N(T) = \sum_{i=2}^{N-1} \alpha_i(T)a_{iN}.$$
**Evaluation: Forward-Backward Algorithm**

- Define the backward probability $\beta$ as
  \[ \beta_i(t) = p(X_{t+1}, \ldots, X_T \mid S_t = i) \]
  
  Then
  \[ \beta_i(T) = a_{1N} \]
  
  \[ \beta_i(t) = \sum_{j=2}^{N-1} a_{ij} b_j(X_{t+1}) \beta_j(t + 1), \]
  
  \[ \beta_1(1) = \sum_{j=2}^{N-1} a_{1j} b_j(X_1) \beta_j(1) \]
Evaluation: Data Likelihood

• The joint probability of $S_t=j$ and $X$ is

$$p(X^T, S_t = j) = \alpha_j(t)\beta_j(t)$$

The data likelihood is

$$p(X^T) = \sum_j p(X^T, S_t = j) = \sum_j \alpha_j(t)\beta_j(t)$$
Evaluation: The Baum-Welch Algorithm

\[ \alpha_t(i) = P(X_t^t, s_t = i \mid \Phi) = \left[ \sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji} \right] b_i(X_t) \quad 2 \leq t \leq T; \quad 1 \leq i \leq N \]

\[ \beta_t(j) = P(X_{t+1}^T \mid s_t = j, \Phi) = \left[ \sum_{i=1}^{N} a_{ji} b_i(X_{t+1}) \beta_{t+1}(i) \right] \quad t = T - 1 \ldots 1; \quad 1 \leq j \leq N \]
Decoding: Viterbi Algorithm

Optimal alignment between X and S
Reestimation: Training Algorithm

• Find model parameters $\Phi=(A, B, \pi)$ that maximize the probability $p(X|\Phi)$ of observing some data $X$:

$$P(X|\Phi) = \sum_s P(X,S|\Phi) = \sum_s \prod_{t=1}^T a_{s_{t-1}s_t} b_{s_t}(X_t)$$

• There is no closed-form solution and thus we use the **EM algorithm**: given $\Phi$, we compute model parameters $\hat{\Phi}$ that maximize the following auxiliary function $Q$

$$Q(\Phi,\hat{\Phi}) = \sum_s \frac{P(X,S|\hat{\Phi})}{P(X|\Phi)} \log P(X,S|\hat{\Phi}) = Q_{a_j}(\Phi,\hat{a}_j) + Q_{b_j}(\Phi,\hat{b}_j)$$

it is guaranteed to have increased (or the same) likelihood
Iterative Training Procedure

• Several iterations of the process:
Continuous Observation Density HMM’s

Most general form of pdf with a valid re-estimation procedure is:

\[
b_j(x) = \sum_{m=1}^{M} c_{jm} \cdot [x, \mu_{jm}, U_{jm}], \quad 1 \leq j \leq N
\]

- \( x \) = observation vector = \( \{x_1, x_2, \ldots, x_D\} \)
- \( M \) = number of mixture densities
- \( c_{jm} \) = gain of \( m \)-th mixture in state \( j \)
- \( \mu_{jm} \) = mean vector for mixture \( m \), state \( j \)
- \( U_{jm} \) = covariance matrix for mixture \( m \), state \( j \)

\( c_{jm} \geq 0, \quad 1 \leq j \leq N, \quad 1 \leq m \leq M \)

\[
\sum_{m=1}^{M} c_{jm} = 1, \quad 1 \leq j \leq N
\]

\[
\int_{-\infty}^{\infty} b_j(x) dx = 1, \quad 1 \leq j \leq N
\]
Word Lexicon

Goal:
Map legal phone sequences into words according to phonotactic rules. For example,

David /d/ /ey/ /v/ /ih/ /d/

Multiple Pronunciation:
Several words may have multiple pronunciations. For example

Data /d/ /ae/ /t/ /ax/
Data /d/ /ey/ /t/ /ax/

Challenges:
How do you generate a word lexicon automatically; how do you add new variant dialects and word pronunciations.
Lexical Modeling

• Assume each HMM → a monophone model
  American English: 42 monophone → 42 HMMs
  – Concatenation of phone models (phone HMM’s)
  – Lexicon: /science/ = /s/+/ai/+/e/+/n/+/s/ or /s/+/ai/+/n/+/s/
  – Multiple pronunciations and pronunciation network
Word-Juncture Modeling

- Co-articulation Effect
  - Simple concatenation of word models (word HMM’s)
  - Hard change: “did you” = /d/+/i/+/dzj/+/u/
  - Soft change: possible pronunciation variations
  - Source of major errors in many ASR systems
  - Easier to handle in syllabic languages with open syllables (vowel or nasal endings, e.g. Japanese, Mandarin)
Modeling Triphone

- Monophone modeling is too simple to model coarticulation phenomenon ubiquitous in speech
- Modeling context-dependent phonemes: triphone
  - American English: 42X42X42 triphones $\rightarrow$ 74,088 HMMs
Grammar Network Expansion: Monophones
Grammar Network Expansion: Triphones
Grammar Network Expansion: Cross-Word
Goal: Model “acceptable” spoken phrases, constrained by task syntax.

Rule-based: Deterministic grammars that are knowledge driven. For example,
fly from \$city\ to \$city\ on \$date\n
Statistical: Compute estimate of word probabilities (N-gram, class-based, CFG). For example
fly from Newark to Boston tomorrow

Challenges: How do you build a language model rapidly for a new task?
Formal Grammars

Rewrite Rules:

1. \( S \rightarrow NP \ VP \)
2. \( VP \rightarrow V \ NP \)
3. \( VP \rightarrow AUX \ VP \)
4. \( NP \rightarrow ART \ NP1 \)
5. \( NP \rightarrow ADJ \ NP1 \)
6. \( NP1 \rightarrow ADJ \ NP1 \)
7. \( NP1 \rightarrow N \)
8. \( NP \rightarrow NAME \)
9. \( NP \rightarrow PRON \)
10. \( NAME \rightarrow Mary \)
11. \( V \rightarrow loves \)
12. \( ADJ \rightarrow that \)
13. \( N \rightarrow person \)
## Chomsky Grammar Hierarchy

<table>
<thead>
<tr>
<th>Types</th>
<th>Constraints</th>
<th>Automata</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase structure grammar</td>
<td>$\alpha \rightarrow \beta$. This is the most general grammar.</td>
<td>Turing machine</td>
</tr>
<tr>
<td>Context-sensitive grammar</td>
<td>Subset of $\alpha \rightarrow \beta \mid</td>
<td>\alpha</td>
</tr>
<tr>
<td>Context-free grammar (CFG)</td>
<td>$A \rightarrow w$ and $A \rightarrow BC$, where $w$ is a terminal and $B$, $C$ are non-terminals.</td>
<td>Push down automata</td>
</tr>
<tr>
<td>Regular grammar</td>
<td>$A \rightarrow w$ and $A \rightarrow wB$</td>
<td>Finite-state automata</td>
</tr>
</tbody>
</table>
Other Examples of Grammar Network

Word-loop grammar:

- For all possible sentences.
- Each branch represents a word in vocabulary
- May add transition probabilities from language models

Grammar for Voice Dialing
\[ P(\mathbf{W}) = P(w_1, w_2, \ldots, w_N) \]
\[ = P(w_1) P(w_2 | w_1) P(w_3 | w_1, w_2) \cdots P(w_N | w_1, w_2, \ldots, w_{N-1}) \]
\[ = \prod_{i=1}^{N} P(w_i | w_1, w_2, \ldots, w_{i-1}) \]

Trigram Estimation

\[ P(w_i | w_{i-1}, w_{i-2}) = \frac{C(w_{i-2}, w_{i-1}, w_i)}{C(w_{i-2}, w_{i-1})} \]
Example: bi-grams
(Probability of a word given the previous word)

I want to go from Boston to Denver tomorrow
I want to go to Denver tomorrow from Boston
Tomorrow I want to go from Boston to Denver
with United
A flight on United going from Boston to
Denver tomorrow
A flight on United
Boston to Denver
Going to Denver
United from Boston
Boston with United tomorrow
Tomorrow a flight to Denver
Going to Denver tomorrow with United
Boston Denver with United
A flight with United tomorrow

I want to go from Boston to Denver tomorrow
I want to go to Denver tomorrow from Boston
Tomorrow I want to go from Boston to Denver
with United
A flight on United going from Boston to
Denver tomorrow
A flight on United
Boston to Denver
Going to Denver
United from Boston
Boston with United tomorrow
Tomorrow a flight to Denver
Going to Denver tomorrow with United
Boston Denver with United
A flight with United tomorrow
Generalization for $N$-grams (back-off)

If the bi-gram $ab$ was never observed, we can estimate its probability:

Probability of word $b$ following word $a$ as a fraction of probability of word $b$

$$
\begin{array}{c}
0.0 & 0.0 & 0.02 \\
0.02 & 0.0 & 0.01 & 0.02 \\
\end{array}
$$

I want a flight from Boston to Denver $\rightarrow$ 0.0

BAD!!!

$$
\begin{array}{c}
0.001 & 0.002 & 0.02 \\
0.02 & 0.003 & 0.01 & 0.02 \\
\end{array}
$$

I want a flight from Boston to Denver $\rightarrow$ 4.8e-15

GOOD
**Goal:** Combine information (probabilities) from the acoustic model, language model and word lexicon to generate an “optimal” word sequence (highest probability).

**Method:** Decoder searches through all possible recognition choices using a Viterbi decoding algorithm.

**Challenges:** How do we build efficient structures (FSMs) for decoding and searching large vocabulary, complex language models tasks;
- features x HMM units x phones x words x sentences can lead to search networks with $10^{22}$ states
- FSM methods can compile the network to $10^8$ states—14 orders of magnitude more efficient
  
  what is the theoretical limit of efficiency that can be achieved?
ASR: Viterbi Search

• Assume we build the grammar network for the task, with all trained HMMs attached in the network
• Unknown utterance → a sequence of feature vectors $Y$
• Speech recognition is nothing more than a viterbi search:
  – The whole network viewed as a composite HMM $\Lambda$
  – $Y$ viewed as input data, find the optimal path (viterbi path) $S^*$ traversing the whole network (from START to END)

$$S^* = \arg \max_{S \in \Theta} \Pr(S) \cdot p(Y, S | \Lambda) = \arg \max_{S \in \Theta} \Pr(W_S) \cdot p(Y, S | \Lambda)$$

  – Once $S^*$ is found, the recognition results (word sequence) can be derived by backtracking the Viterbi path
ASR Issues

• Training stage:
  – *Acoustic modeling*: how to select speech unit and estimate HMMs reliably and efficiently from available training data.
  – *Language modeling*: how to estimate $n$-gram model from text training data; handle data sparseness problem.

• Test stage:
  – *Search*: given HMM’s and $n$-gram model, how to efficiently search the optimal path from the huge grammar network.
    • Search space is extremely large
    • Call for an efficient pruning strategy
Acoustic Modeling

• Selection of Speech Units: modeled by a HMMDigit string recognition: a digit by a HMM \( \rightarrow 10-12 \) HMMs
  – Large vocabulary: monophone \( \rightarrow \) biphone \( \rightarrow \) triphone

• HMM topology selection
  – Phoneme: 3-state left-right without skipping state
  – Digit/word: 6-12 states left-right no state skipping

• HMM type selection
  – Top choice: Gaussian mixture CDHMM
  – Number of Gaussian mixtures in each state (e.g., 16)

• HMM parameters estimation:
  – ML (Baum-Welch algorithm) and others
Selecting Speech Segments for each HMM

Monophone HMMs

Reference Segmentation

Triphone HMMs

Reference Segmentation
Reference Segmentation

- Where the segmentation information comes from?
  - Human labeling: tedious, time-consuming;
    - Only a small amount is affordable; used for bootstrap.
  - Automatic segmentation if a simple HMM set is available.
    - Forced-alignment: Viterbi algorithm; Need transcription only
    - HMMs + transcription $\Rightarrow$ segmentation information

Transcription: This is a test.

Run the Viterbi algorithm to backtrack segmentation information
Embedded Training

- Only need transcription for each utterance; no segmentation is needed; automatically tune to optimal segmentation during training

Transcription: This is a test.

Word network
Phoneme network
Composite HMM

Add optional 1-state silence models between words
Isolated Word HMM Recognizer

SPEECH SIGNAL S → LPC FEATURE ANALYSIS, (VECTOR QUANTIZATION) → OBSERVATION SEQUENCE O → PROBABILITY COMPUTATION P(0|λ^1) → HMM FOR WORD 1

... → PROBABILITY COMPUTATION P(0|λ^2) → HMM FOR WORD 2

... → PROBABILITY COMPUTATION P(0|λ^V) → HMM FOR WORD V

INDEX OF RECOGNIZED WORD

ν* = \text{ARGMAX} \{P(0|λ^ν)\} \quad 1 ≤ ν ≤ V

SELECT MAXIMUM
Choice of Model Parameters

• Left-to-right model preferable to ergodic model (speech is a left-right process)

• Number of states in range 2-40 (from sounds to frames)
  – Order of number of distinct sounds in the word
  – Order of average number of observations in word

• Observation vectors
  – Cepstral coefficients (and their time derivatives) derived from LPC (1-9 mixtures), diagonal covariance matrices
  – Vector quantized discrete symbols (16-256 codebook sizes)

• Constraints on $b_j(O)$ densities
  – $b_j(k) > \varepsilon$ for discrete densities
  – $C_{jm} > \delta$, $U_{jm}(r,r) > \delta$ for continuous densities
Performance vs. Number of States in Model

ERROR RATE IN PERCENT

$N$, NUMBER OF STATES IN HMM
Goal: Identify possible recognition errors and out-of-vocabulary events. Potentially improves the performance of ASR, SLU and DM.

Method: A confidence score based on a hypothesis test is associated with each recognized word. For example:

<table>
<thead>
<tr>
<th>Label:</th>
<th>credit please</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recognized:</td>
<td>credit fees</td>
</tr>
<tr>
<td>Confidence:</td>
<td>(0.9) (0.3)</td>
</tr>
</tbody>
</table>

\[ P(W \mid X) = \frac{P(W)P(X \mid W)}{\sum_W P(W)P(X \mid W)} \]

Challenges: Rejection of extraneous acoustic events (noise, background speech, door slams) without rejection of valid user input speech.
Robustness

Problem:
a mismatch in the speech signal between the training phase and testing phase can result in performance degradation.

Methods:
traditional techniques for improving system robustness are based on signal enhancement, feature normalization or/and model adaptation.

Perception Approach:
extract fundamental acoustic information in narrow bands of speech. Robust integration of features across time and frequency.
Robust Speech Recognition

- A mismatch in the speech signal between the training phase and testing phase results in performance degradation
Rejection

**Problem:**
Extraneous acoustic events, noise, background speech and out-of-domain speech deteriorate system performance.

**Measure of Confidence:**
Associating word strings with a verification cost that provide an effective measure of confidence (Utterance Verification).

**Effect:**
Improvement in the performance of the recognizer, understanding system and dialogue manager.
State-of-the-Art Performance

TTS ➔ ASR

SLG ➔ DM

SLU

Input Speech ➔ Feature Extraction ➔ Pattern Classification (Decoding, Search) ➔ Confidence Scoring ➔ Recognized Sentence

Acoustic Model

Language Model

Word Lexicon
How to Evaluate Performance?

• Dictation applications: Insertions, substitutions and deletions

Word Error Rate = $100\% \times \frac{\#\text{Subs} + \#\text{Dels} + \#\text{Ins}}{\text{No. of words in the correct sentence}}$

• Command-and-control: false rejection and false acceptance => ROC curves
## Word Error Rates

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Type</th>
<th>Vocabulary Size</th>
<th>Word Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connected Digit Strings--TI Database</td>
<td>Spontaneous</td>
<td>11 (zero-nine, oh)</td>
<td>0.3%</td>
</tr>
<tr>
<td>Connected Digit Strings--Mall Recordings</td>
<td>Spontaneous</td>
<td>11 (zero-nine, oh)</td>
<td>2.0%</td>
</tr>
<tr>
<td>Connected Digits Strings--HMIHY</td>
<td>Conversational</td>
<td>11 (zero-nine, oh)</td>
<td>5.0%</td>
</tr>
<tr>
<td>RM (Resource Management)</td>
<td>Read Speech</td>
<td>1000</td>
<td>2.0%</td>
</tr>
<tr>
<td>ATIS (Airline Travel Information System)</td>
<td>Spontaneous</td>
<td>2500</td>
<td>2.5%</td>
</tr>
<tr>
<td>NAB (North American Business)</td>
<td>Read Text</td>
<td>64,000</td>
<td>6.6%</td>
</tr>
<tr>
<td>Broadcast News</td>
<td>News Show</td>
<td>210,000</td>
<td>13-17%</td>
</tr>
<tr>
<td>Switchboard</td>
<td>Conversational Telephone</td>
<td>45,000</td>
<td>25-29%</td>
</tr>
<tr>
<td>Call Home</td>
<td>Conversational Telephone</td>
<td>28,000</td>
<td>40%</td>
</tr>
</tbody>
</table>

factor of 17 increase in digit error rate
NIST Benchmark Performance

Word Error Rate

Year

ATIS

NAB

Resource Management

Spontaneous Speech

Broadcast Speech

Read Speech

Conversational Speech

Varied Microphones

10k

5k

1k

Noisy

1%

10%

100%

Accuracy for Speech Recognition

Switchboard/Call Home Vocabulary: 40,000 words  Perplexity: 85
Challenges in ASR

System Performance
- Accuracy
- Efficiency (speed, memory)
- Robustness

Operational Performance
- End-point detection
- User barge-in
- Utterance rejection
- Confidence scoring

Machines are 10-100 times less accurate than humans
The End

Thank you!
Speech Recognition Capabilities

- Spontaneous Speech
- Fluent Speech
- Read Speech
- Connected Speech
- Isolated Words

- Speaking Style
  - Vocabulary Size

- 1997
- 2001

- Name Dialing
- City Dialing
- Word spotting
- System driven dialogue
- Speaker verification
- User driven dialogue
- Personal assistant
- Office dictation
- Natural conversation

- By B. S. Atal