

Hidden Markov Models (HMMs)

From Spelling Recognition to Speech Recognition

Shieu-Hong Lin

Biola University

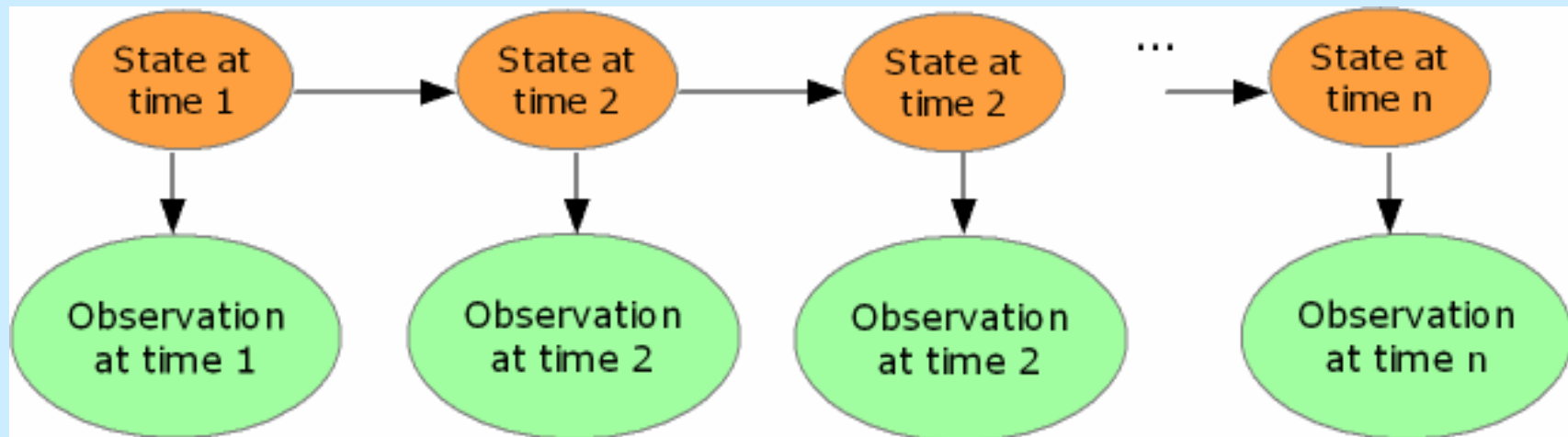
AI: Could the computer “**perceive**”, “act” or “think” like human beings?

- Automatic facial recognition (vision)
- DARPA autonomous vehicles (vision and more)
- Honda Asimo humanoid robot (vision, robotics, and more)
- **Speech recognition, spelling recognition**
- Deep Blue chess program
- ...

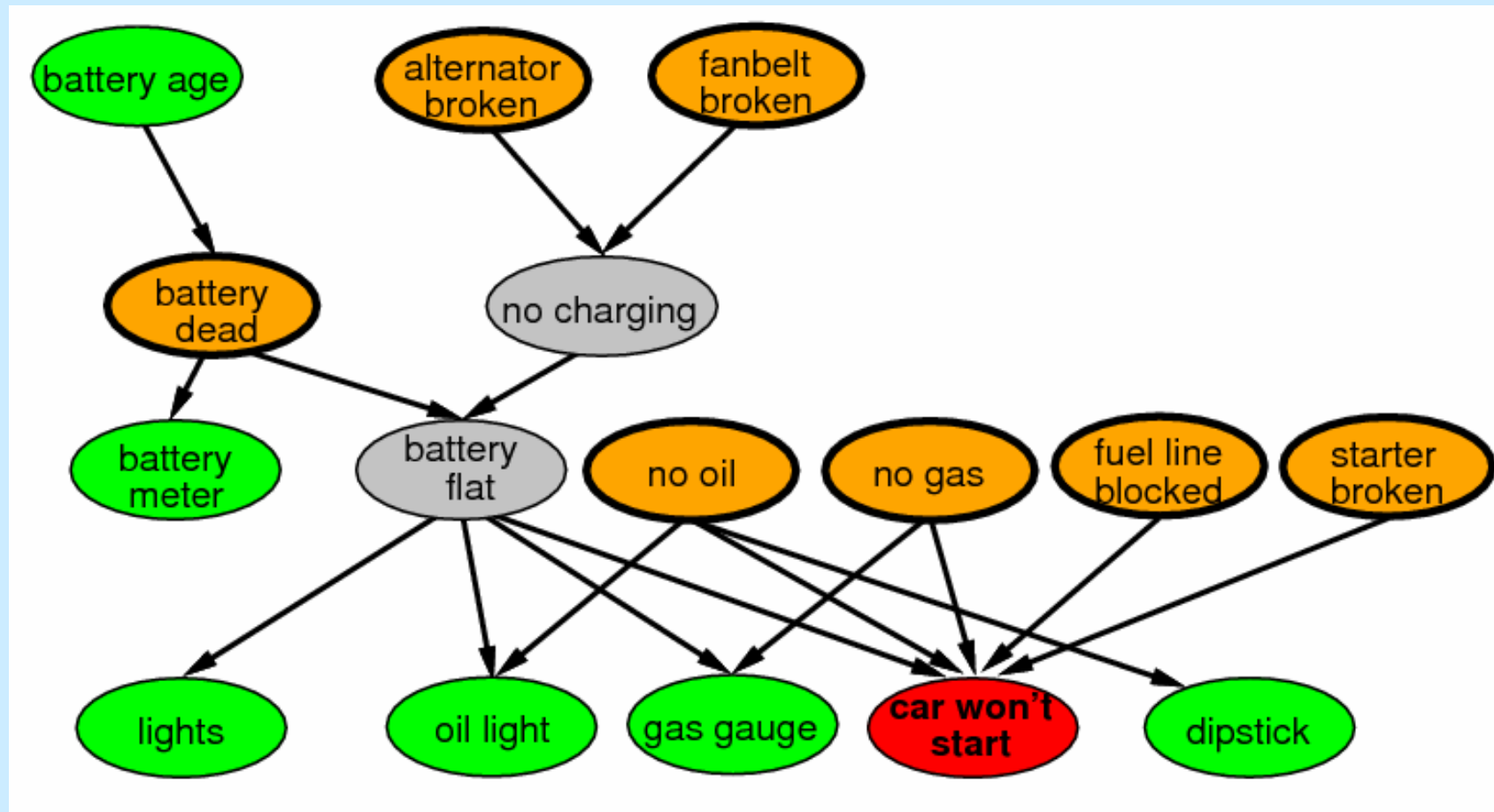
Hidden Markov Models (HMMs)

- A key element in the success of speech recognition, part-of-speech tagging, computer vision, ...
- Infer the hidden things from the observable
- Probabilistic modeling & reasoning

HMMs as Bayesian Networks



General Bayesian Networks



From Spelling Recognition to Speech Recognition

- Spelling recognition
- Modeling & problem solving using HMMs
- Hands-on experiments
- Speech recognition
- Automatic identity recognition
- Conclusions

Spelling Recognition

- A set of words: the vocabulary set Σ
- **Spelling recognition:** Given a character string S , match S with the most likely word W in Σ .
- Example: $\Sigma = \{\text{is, are, am}\}$

iis → is

ae → are

an → am

Spelling Recognition

- Given a word w , what do we mean by *the most likely word* for w in Σ ?
 - ➔ Needs probability models
- How to find the most likely word for w ?
 - ➔ Needs to develop algorithms

Modeling Cognitive Errors

Cognitive errors:

wrong transitions of cognitive states

For the word *his*

Ideally: $h \rightarrow i \rightarrow s$

but things like: $h \rightarrow h \rightarrow i \rightarrow s$

or $h \rightarrow s$

or ...

could happen.

The Spelling Models

Cognitive errors:

How likely, we may

- Repeat a character: his → hh^{is} ,
- Skip characters: hⁱs → hs ,
- ...

?

The Spelling Models

For each word, modeling the probabilistic transitions among cognitive states.

What are

$$\Pr(\text{next} = \text{'h'} \mid \text{current} = \text{'h'}) = ?$$

$$\Pr(\text{next} = \text{'i'} \mid \text{current} = \text{'h'}) = ?$$

$$\Pr(\text{next} = \text{'s'} \mid \text{current} = \text{'h'}) = ?$$

...

The Spelling Models

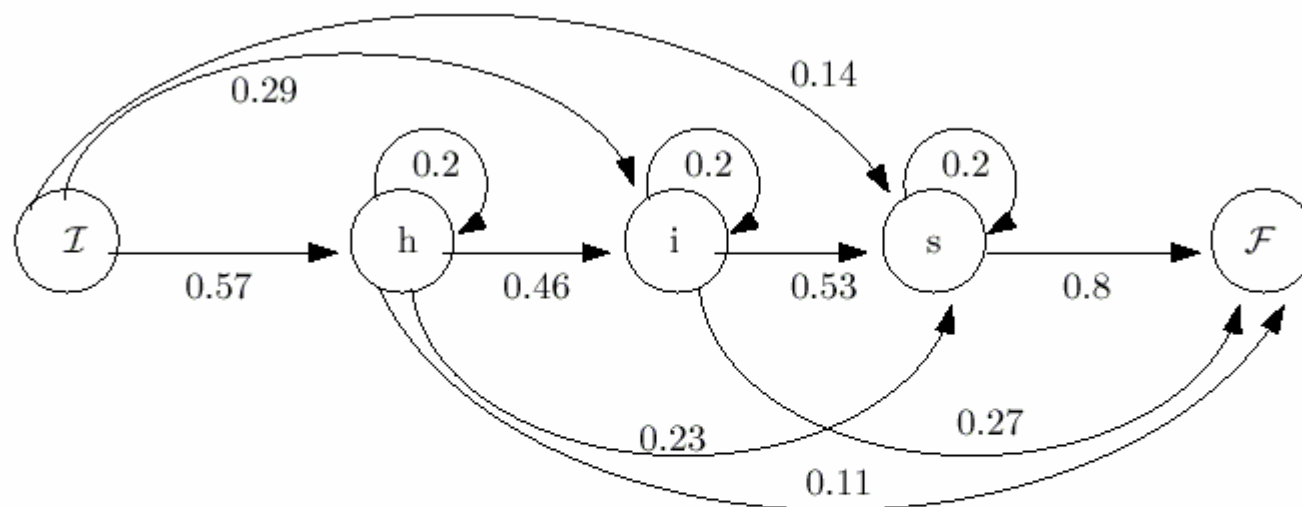


Figure 1: The spelling model regarding the word “his” with the parameters $deg_{sp} = 2$, $p_{repeat} = 0.2$, and $p_{moveOn} = 0.8$

Modeling Typographic Errors

Typographic errors:

cognitively right, but press a wrong key

– his → hia

You want to type s

Ideally: you touch s

but you might touch a, b , c ,..., x , y , z

The keyboard models

For each character, model the probability distribution over the keys possibly pressed when typing. What are

$$\Pr(\text{touch 'a'} \mid \text{intend to type 's'}) = ?$$

$$\Pr(\text{touch 'b'} \mid \text{intend to type 's'}) = ?$$

...

$$\Pr(\text{touch 'z'} \mid \text{intend to type 's'}) = ?$$

Spelling Recognition

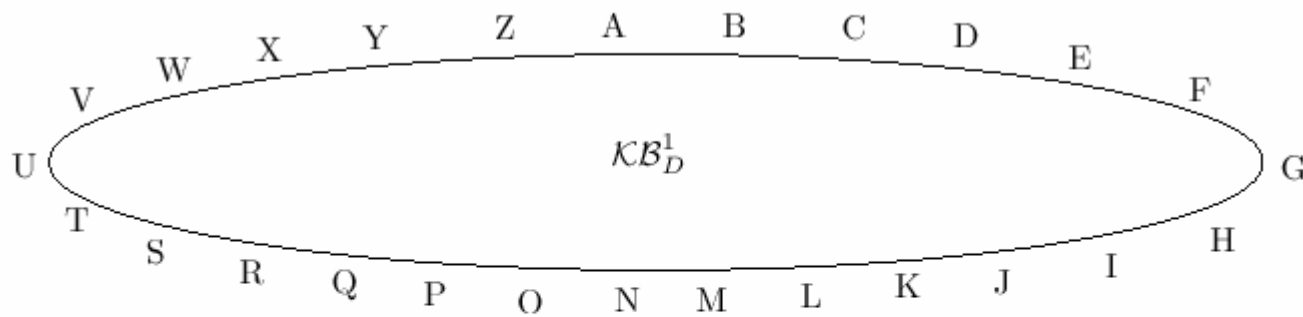


Figure 2: The one-dimensional keyboard model $\mathcal{KB}_D^1$

Hands-on Experiments: Setting

Keyboard-typing characteristics of

Mr. L in a chat-room:

- Type with a one-dimensional keyboard
- For any 2 keys x , y ($x \neq y$),
 $\Pr(\text{touch } x \mid \text{intend to type } x) = 0.8$
 $\Pr(\text{touch } y \mid \text{intend to type } x)$ proportional
to the **inverse** of distance between x , y
on the keyboard

Hands-on Experiments: Setting

Characteristics of Mr. L's cognitive transitions:

For any 2 characters x , y in a word W

(y after x in W),

- $\Pr(\text{next} = x \mid \text{current} = x) = 0.1$
- $\Pr(\text{next} = y \mid \text{current} = x)$ proportional to the **inverse** of the 5th power of distance between x , y in the word

Hands-on Experiments

Spelling recognition with Mr. L in a chat-room:

<http://csci.biola.edu/HMM/DemoExecutable.zip>

1. Vocabulary: **testVocabulary.txt**
2. Results of random words chosen and typed by Mr. L: **corruptedMessage.txt**
3. Results of spelling recognition:
recoveredTestMessage.txt
4. Executable: **demo.exe**

Task of Spelling Recognition

Input:

a string of observed character $o_{t1} o_{t2} \dots o_{tm}$

Output:

the most likely k matching words in the vocabulary

Hidden Markov Models (HMMs)

A HMM: a four tuple $\langle \mathbf{S}, \mathbf{O}, \mathbf{P}_T, \mathbf{P}_o \rangle$

- $\mathbf{S}$: states, not directly observable
- $\mathbf{O}$: directly observable evidences of states
- $\mathbf{P}_T$: state-transition probabilities
- $\mathbf{P}_o$: state-observation probabilities

HMMs for Spelling Recognition

For each word W , the spelling model and the keyboard model combined to form a HMM

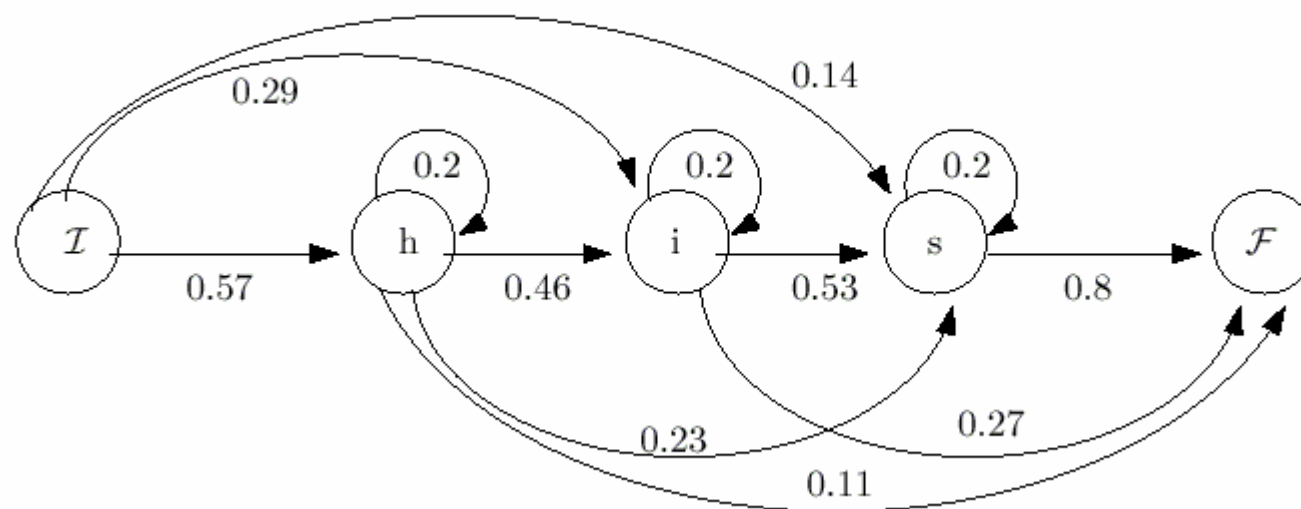
- **S** : the cognitive states regarding the individual characters in W
- **O**: Observations of the 26 letters

HMMs for Spelling Recognition

For each word W , the spelling model and the keyboard model combined to form a HMM

- P_T , Transition probability :
from the spelling model of W
- P_o , Observation probability :
from the keyboard model

HMM for Spelling Recognition



The Viterbi algorithm

Given a HMM $H = \langle S, O, P_T, P_o \rangle$, and
a series of m observations $o_{t1} o_{t2} \dots o_{tm}$
over time, $\rightarrow$
efficiently determines the probability of
 H generating $o_{t1} o_{t2} \dots o_{tm}$
in $O(mn^2)$ time where n is size of S
instead of $O(n^m)$ time using a naïve
approach

HMM for Spelling Recognition

Input: a character string $o_{t1} o_{t2} \dots o_{tm}$

Output: the most likely k matching words

1. For each word **W** in the vocabulary, use the Viterbi algorithm to calculate the probability of seeing **S** produced when trying to type the word **W**
2. Pick the k words that yield top k probabilities in step 1 above

Speech Recognition

- Recognize words from observable acoustic signals
- Similar HMMs built and used for speech recognition
- Construct HMMs from word models and phone models
- Use the Viterbi algorithm to find most likely words from observed acoustic signals

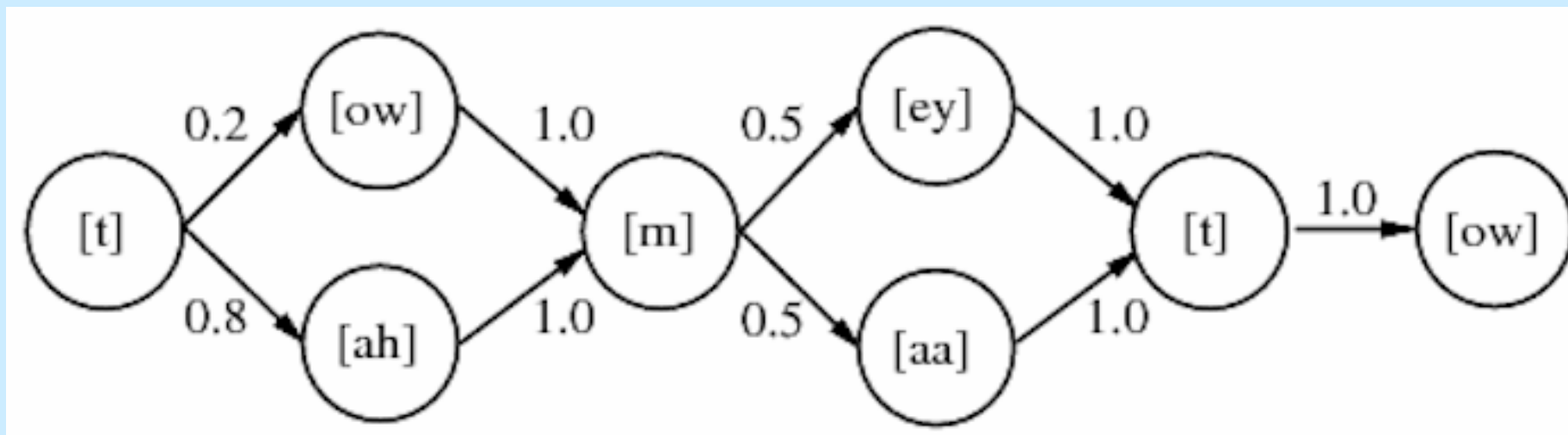
Speech Recognition & HMMs

- Word models in speech recognition
→ like the spelling models
in spelling recognition
- Phone models in speech recognition
→ like the keyboard models
in spelling recognition

Speech Recognition & HMMs

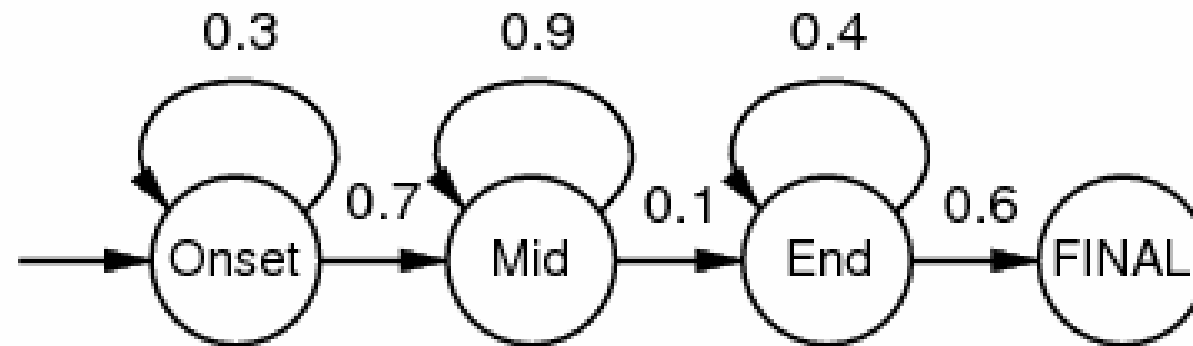
- Word models: Cognitive states about phones in words.

Example, the word model for “Tomato”



Speech Recognition & HMMs

- Phone models: e.g, the phone model for the phone [m]

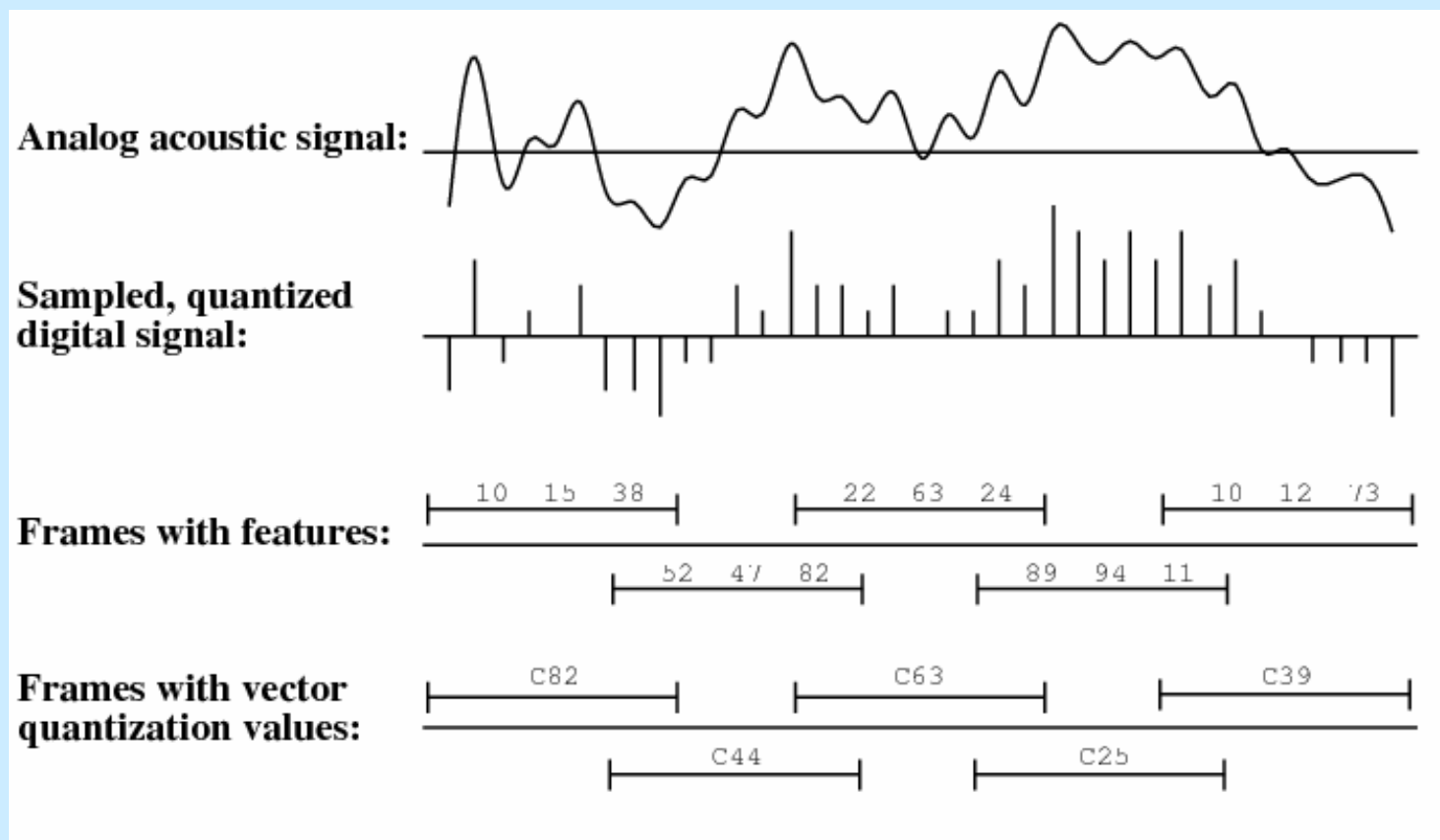


Observation probabilities

Onset:	Mid:	End:
C1: 0.5	C3: 0.2	C4: 0.1
C2: 0.2	C4: 0.7	C6: 0.5
C3: 0.3	C5: 0.1	C7: 0.4

Speech Recognition & HMMs

- From acoustic signals to frames in the phone models



Recognition of Sentences

- **Recognition of sentences, not just words individually:**

Given a series of frames of acoustic signals (or character strings for spelling recognition)

$S_1 S_2 \dots S_m$, find the most likely word series $W_1 W_2 \dots W_m$ formed by words from Σ .

- Example: $\Sigma = \{I, is, are, am\}$

J an → I am

Extension of Probability Models

The Language Model:

e.g. you see: J ar happy

much more likely from

I am happy

than

I an happy

The Language Model:

- Model the probability of sentences as sequence of words?
- A good language model (such as the bi-gram model) can further improve the accuracy by considering the context.

Automatic Identity Recognition

Example:

- Given eight text files A, B, C, D, E, F, G, and H resulting from four persons Johnny, Winnie, Manny, and Cathy,
- determine for each text who types the text.

Automatic Identity Recognition

Approach:

- For each document $\mathbf{d}$ in $\{A, B, C, D, E, F, G\}$ and each person $\mathbf{p}$ in $\{\text{Johnny, Winnie, Manny, Cathy}\}$,
- calculate the probability $\text{Pr}(\mathbf{d}/\mathbf{p})$ that document $\mathbf{d}$ is the resulting text when person $\mathbf{p}$ types the Biola vision text.
- For each document $\mathbf{d}$, pick the person $\mathbf{p}$ with the highest probability $\text{Pr}(\mathbf{d}/\mathbf{p})$.

Conclusions

- Mathematical structures and algorithms: two key components of modern AI research.
- HMMs are very useful for modeling and probabilistic reasoning
- Design and implementation of efficient algorithms is critical in problem solving
- Bayesian networks more expressive than HMMs are widely used in expert systems