Topics in AI

Five main areas:
- Problem solving and search
- Logic and knowledge representations
- Planning
- Uncertainty
- Learning

Other topics:
- AI programming languages
- Speech recognition
- Natural language processing
- Image understanding
- Robotics
Speech recognition

- **Objective**: take acoustic signal and convert it to text

Analog acoustic signal:

Sampled, quantized digital signal:

Frames with features:

Frames with vector quantization values:

Discretize features: e.g. to 256 values (8 bits)

---

Speech recognition

- We want to determine the sequence of words that is most probable given the input signal
  \[ P(\text{wordseq} = w \mid \text{signal} = s) \]
- It is easier to define an **acoustic model** that relates:
  \[ P(\text{signal} = s \mid \text{wordseq} = w) \]
- This is like a diagnosis problem, we can use the Bayes rule:
  \[
  P(\text{wordseq} = w \mid \text{signal} = s) = \frac{P(\text{signal} = s \mid \text{wordseq} = w)P(\text{wordseq} = w)}{P(\text{signal} = s)}
  \]
- Assume we have multiple possible word sequences: \( w^1, w^2, \ldots, w^k \)
- The best word sequence:
  \[
  \arg\max_{w'} P(\text{signal} = s \mid \text{wordseq} = w') P(\text{wordseq} = w')
  \]
Speech recognition

- We need to define:
  \[ P(\text{signal}=s \mid \text{wordseq}=w) \] and \[ P(\text{wordseq}=w) \]
  for all possible word and signal sequences

- **Defining the probability:** \[ P(\text{wordseq}=w) \]
  \[ w = w_1 w_2 \cdots w_n \]
  \[ P(\text{wordseq}=w_1 w_2 \cdots w_n) = P(w_1) P(w_2 \mid w_1) \cdots P(w_n \mid w_1 w_2 \cdots w_{n-1}) \]
  – By the chain rule

- **Simplifications:**
  - **Unigram model:** a probability of each word is independent of the previous word
  \[ P(\text{wordseq}=w_1 w_2 \cdots w_n) = P(w_1) P(w_2) P(w_3) \cdots P(w_n) \]
  - **Bigram model:** only the previous word matters
  \[ P(\text{wordseq}=w_1 w_2 \cdots w_n) = P(w_1) P(w_2 \mid w_1) P(w_3 \mid w_2) \cdots P(w_n \mid w_{n-1}) \]

Speech recognition

- **Defining the probability:** \[ P(\text{signal}=s \mid \text{wordseq}=w) \]
  \[ s = s_1 s_2 s_3 \cdots s_m \]
  \[ w = w_1 w_2 \cdots w_n \]

- **Two simplifications:**
  1. Define signal signatures for individual words
     \[ P(s = s_1 s_2 \cdots s_j \mid \text{word}=w_i) \]
  2. Divide the acoustic word models into a sequence of phones and define signal signature models for phones
     \[ P(p = p_1 p_2 \cdots p_n \mid \text{word}=w_i) \]
     \[ P(s = s_1 s_2 \cdots s_e \mid \text{phone}=p_q) \]

  Conditional probabilities of sequences modeled most often as:
  - **Hidden Markov Models (HMMs)**
Speech recognition

HMM models of words \( P(p = p_1 p_2 \ldots p_n \mid \text{word} = w_i) \)

- Example: word: tomato

Word model with dialect variation:

- 2 phones sequences

Word model with coarticulation and dialect variations:

- 4 phones sequences

Speech recognition

HMM model of phones \( P(s = s_1 s_2 \ldots s_r \mid \text{phone} = p_q) \)

Example:

<table>
<thead>
<tr>
<th>Phone HMM for [m]:</th>
<th>Many possible feature sequences:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output probabilities for the phone HMM:</td>
<td>C1 C4 C6</td>
</tr>
<tr>
<td>C1: 0.5</td>
<td>C3: 0.1</td>
</tr>
<tr>
<td>C2: 0.2</td>
<td>C4: 0.7</td>
</tr>
<tr>
<td>C3: 0.3</td>
<td>C5: 0.1</td>
</tr>
<tr>
<td>C4: 0.1</td>
<td>C6: 0.5</td>
</tr>
<tr>
<td>C7: 0.4</td>
<td>C8: 0.4</td>
</tr>
</tbody>
</table>

CS 2710 Foundations of AI
Speech recognition

• **Finding the most probable path** through an HMM for \([m]\)
• **Example:** sequence: C1 C3 C4 C6

Natural language processing

**Goal:** Analyze and interpret the text in the natural language

• **Input:** text sentences.
  – Speech recognition system
  – Optical character recognition (OCR)
  – Documents in the electronic form

• **Output:**
  – Knowledge extracted from the text that supports various inferences

• **Processing (multi-step process):**
  – Syntactic interpretation (parsing)
  – Semantic interpretation
  – Disambiguation & Incorporation
Natural language processing

**Syntactic interpretation (parsing):**
- **Input:** a sentence
- **Output:** a parse tree
- Uses grammar models for parsing the sentence to phrases and terminal symbols
- **Example:** ‘The wumpus is dead’

```
S
   /\  
  NP VP
   /   
Article Noun Verb Adjective

The wumpus is dead
```
- Sometimes we have more than one possible parse. **Stochastic grammars** (quantify the goodness of possible parses)

**Semantic interpretation:**
- **input:** a parse tree
- **output:** a set of meanings, e.g. in First order logic (FOL)
- **Example:** ‘The wumpus is dead’
  - Gives two possible semantic interpretations:
    
    \[ \neg \text{Alive}(Wumpus, \text{Now}) \]
    \[ \text{Tired}(Wumpus, \text{Now}) \]

- **Disambiguation:**
  - chooses the most probable interpretation

- **Incorporation:**
  - The extracted knowledge is checked for consistency against other pieces of knowledge before it is incorporated into the KB
Image processing and vision

- **Classic image processing problem:**
  - Analysis of image and extraction of information from the image
  - Can be used in many applications:
    - Scene analysis
    - Manipulation and navigation tasks
    - Image retrieval
- **Other image processing problems:**
  - **Image enhancement:** degraded image should be improved to restore particular features
  - **Storage and Compression:** Large amounts of data need to be archived or transmitted
  - **Visualization**

Image processing

**Image is defined by**
- a **light intensity function** over the **image plane**
  (Continuous) image is typically **discretized**
- **Image plane is discretized into:**
  - Pixels arranged on the rectangular grid
  - Resolution of the grid determines the spatial quality of the discretization
- **Light intensity values are discretized into:**
  - Integers values in some interval
- **Typical (black and white) image input:**
  - 512x512 pixels
  - Light intensity: 8 bits – 512 types of gray
**Image processing**

Analysis of image and extraction of information from the image

- **Segmentation:**
  - Division of the image to meaningful entities in the scene
  - Relies heavily on edge detection algorithms

---

**Image processing and vision**

Analysis of image and extraction of information from the image

- To recognize (identify) the object from the image we need to compare it with the class pattern
- **Problem:** The position, orientation and the scale of the object in the scene may vary
- **Solution:** Use a set of basic transformations:
  - scaling,
  - translation,
  - rotation of the object
  - Transformations are relatively easy for 2D objects, much harder for 3-D objects
- **Other problems:** light sources and shadows
Image processing and vision

- **More complex task**: analysis of a sequence of related images (videos)
- **Image registration**: the process of measuring visual motion between images.
- **When this is useful**:
  - Video - commercial skip
  - Detection and tracking of objects in the real world

AI programming languages

- **Focus on symbolic processing**

Special AI Languages:
- LISP
- Prolog
- Smalltalk