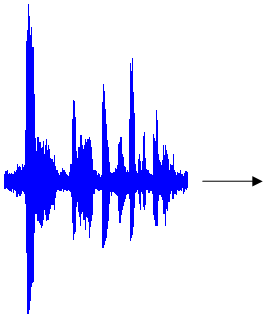
Lecture 14

Speech Recognition

- Aims of Speech Recognition
- Why speech recognition is difficult
- The structure of a speech recogniser
- Statistical Speech Recognition
- Speech Production Model

Aims of Speech Recognition



"This is speech recognition"

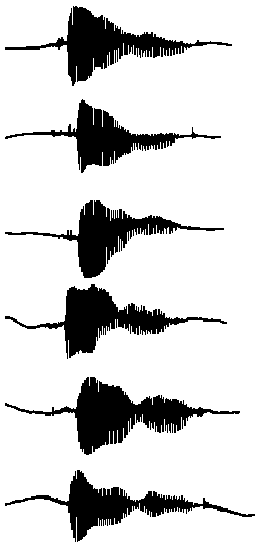
- Convert a microphone waveform to a sequence of words
  - No need to "understand" what the words mean
- Natural Speech: no special behaviour required by speaker
- Large Vocabulary
- Low error rate even with background noise
- Multiple speakers (normally only one at a time)

Difficulties of Speech Recognition

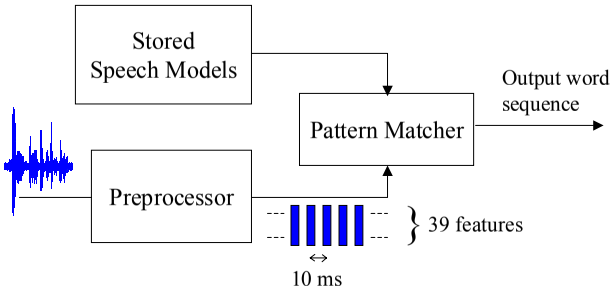
- Speech is highly variable even for the same speaker
  - Variable pitch contour (e.g. surprise, anger)
  - Variable speed
  - Effect of having a blocked nose
- Different speakers pronounce words differently
  - Accents give gross changes: e.g. “bath”
  - Smaller changes within a single accent
- Speech sounds vary according to context
  - “handbag” → “hanbag” → “hambag”
  - “I went by bus” → “I wemp by bus”
  - “Jack and Jill” → “Jack ’n Jill”
- Natural speech includes extraneous sounds
  - Coughs, “umm”, “err”, false starts

Speech Waveforms

- “forward” said by two speakers three times each
- Great variation in *waveforms* despite being the same word
- All recognisers use *spectral information* instead: more consistent.



Recogniser Structure



- Preprocessor converts speech waveform into a sequence of *feature vectors* that describe the speech spectrum:
  - Typical feature vector: 39 different parameters for each 10 ms of speech
    - Different words must give distinctly different feature vectors
    - Different utterances of the *same* word must give similar feature vectors
  - Pattern Matcher uses previously stored speech models to select the sequence of words that is the best match.

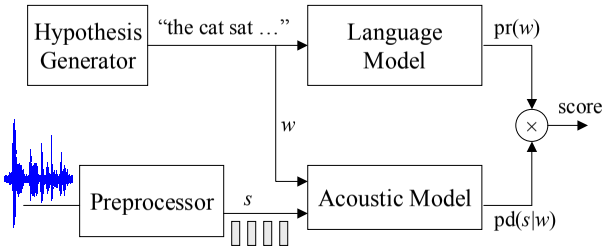
Statistical versus Neural Network

- In a *statistical recogniser*, the pattern matcher
  - uses an *explicit mathematical model* of speech to calculate the probability that a speech signal corresponds to a particular sequence of words. The mathematical model incorporates knowledge about the nature of speech.
  - selects the sequence of words that yields the highest probability.
- In a *neural network* recogniser, the pattern matcher
  - has fewer assumptions about the nature of speech because there is no explicit mathematical model.
  - usually needs *many more parameters* than a statistical recogniser.
- Both types use an iterative algorithm to adjust their parameter values to match *training* examples of speech whose corresponding word sequences are known.
- Statistical recognisers have generally been much more successful.

Statistical Recognition

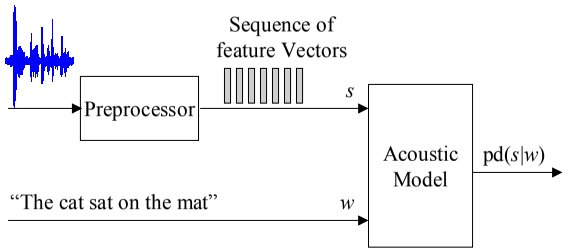
- For a given speech signal, s, we want to find the word sequence, w, that is *most likely* to correspond with it.
  - In mathematical terms we must select w to maximise $\mathrm{pr}(w \mid s)$
    - $\mathrm{pr}(w|s)$ means the probability of w given that s has happened
- We use Bayes' theorem:
  - $\mathrm{pr}(w|s) = \mathrm{pr}(w) \times \mathrm{pr}(s|w) \div \mathrm{pr}(s)$
    - $\mathrm{pr}(w)$ is the *prior probability* of the word sequence w obtained from a *language model* of English (*prior* $\Rightarrow$ before knowing the speech signal s)
    - $\mathrm{pr}(s|w)$ is the *likelihood* of the word sequence w obtained from our stored models of speech. This is a *production probability*: the probability that a speaker will produce the signal s when trying to say the word sequence w.
    - $\mathrm{pr}(s)$ is the *probability* of the signal s.
- $\mathrm{pr}(s)$ is vanishingly small $\Rightarrow$ use probability densities instead:
  - $\mathrm{pr}(w|s) = \mathrm{pr}(w) \times \mathrm{pd}(s|w)ds \div \mathrm{pd}(s)ds$
- Ignore terms that are independent of w. Choose the w that maximises: $\mathrm{pr}(w) \times \mathrm{pd}(s|w)$

Elements of a Recogniser



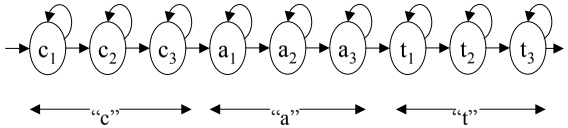
- Preprocessor
  - Converts speech into a sequence of feature vectors at intervals of around 10 ms.
- Language Model
  - Estimate the probability of a word sequence
    - Unlikely word sequences have a low probability
- Acoustic Models
  - Calculate the probability density that an observed sequence of feature vectors corresponds to a particular word sequence
- Hypothesis Generator
  - Try every possible word sequence in turn and choose the one with highest score: a **huge** task.
    - *Essential* to have an efficient algorithm.

Acoustic Modelling



- The acoustic model must give the probability that the observed sequence of feature vectors corresponds to a given word sequence.
  - Convenient to talk about the probability that the model *generates* the observed feature vectors in response to the word sequence.

Speech Production Model



- Each phoneme in a word corresponds to a number of *model states* (typically 3 states per phoneme).
- Each model state represents a distinct sound with its own acoustic spectrum.
  - The feature vectors corresponding to a particular state will be similar but will vary somewhat on different occasions and for different speakers
- When saying a word, the speaker stays in each state for one or more frames and then goes on to the next state.
  - The time in each state will vary according to how fast he/she is speaking.
  - Some speech sounds last longer than others

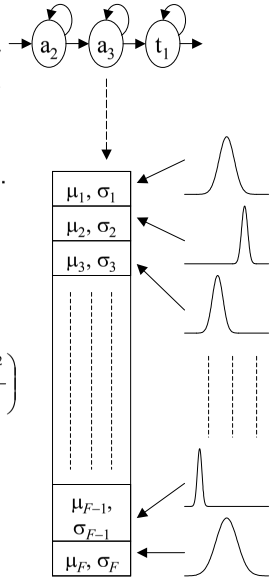
Output Probabilities

- For each state we store the mean, μ, and variance, σ^2, for each of F features.
  - μ_i and σ_i are obtained by averaging the examples of this sound in the training data.
  - A typical value of F is 39.
- Assume the features corresponding to each state have gaussian distributions:

$$\text{pd}(X_i = x) = (2\pi\sigma_i^2)^{-\frac{1}{2}} \exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right)$$

- Assume the features are independent.

$$\text{pd}(\mathbf{X} = \mathbf{x}) = \prod_{i=1}^{F} \text{pd}(X_i = x_i)$$



- For state s, define $d_s(\mathbf{x}) = \text{pd}(\mathbf{X} = \mathbf{x})$

Log Probabilities

- The computation is much easier if we calculate log probabilities instead:

$$d_s(\mathbf{x}) = \prod_{i=1}^{F} \text{pd}(X_i = x_i)$$

$$\log(d_s(\mathbf{x})) = \sum_{i=1}^{F} \log(\text{pd}(X_i = x_i))$$

$$= \sum_{i=1}^{F} -\tfrac{1}{2}\log(2\pi\sigma_i^2) + \sum_{i=1}^{F} -\frac{(x_i-\mu_i)^2}{2\sigma_i^2}$$

- The first term is independent of the observation, $\mathbf{x}$, so can be precalculated for each state.
- Instead of *multiplying* the probability densities for each feature together, we now *add* their logs the total is the log pd that all N features match the observed values.
- Using log probabilities:
  - Avoids calculating exp()
  - Allows a much wider range of values to be stored in the computer