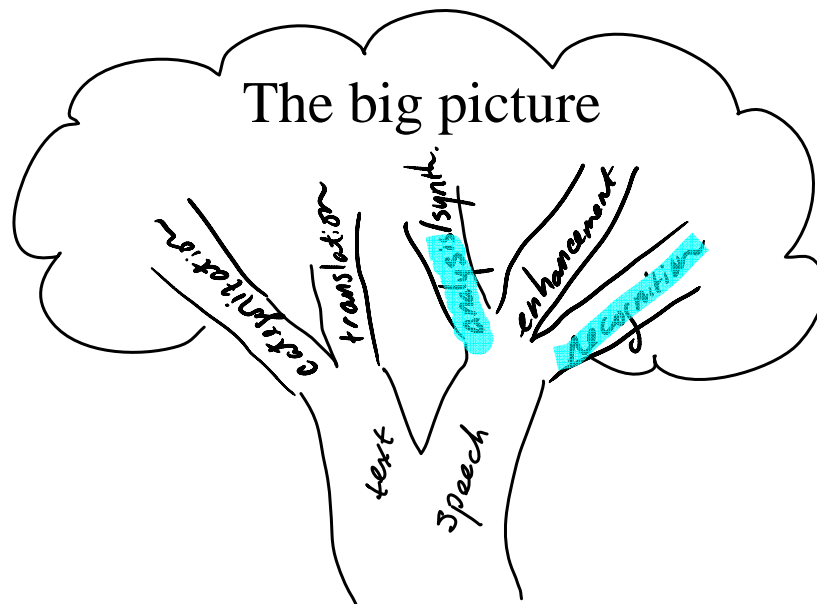


# Speech Processing

Professor Marie Roch  
San Diego State University  
readings HAH: 1.1, 1.2, pp 166-169



# Recognition

- Speech recognition
- Speaker recognition
- Gender recognition
- Language recognition

# Speech Recognition

- Ability to recognize a set of spoken words.
- Speech recognition is not the same as speech understanding.
- Speaker independent versus speaker dependent
- Large versus small vocabulary



## Sample Speech Recognition Applications

- Dialogue systems
- Dictation
- Command and control

## Speaker Recognition

- Determines someone's identity
- Verification versus identification
- Text-dependent versus text-independent
- Closed set versus open set



## Sample Speaker Recognition Applications

- Access control
- Detection
- Verification

## Other Applications

- Bioacoustics
- Auditory scene analysis
- Diarization

# Basic Architecture

- Speech Recognition System

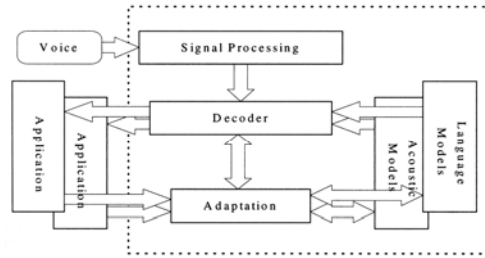
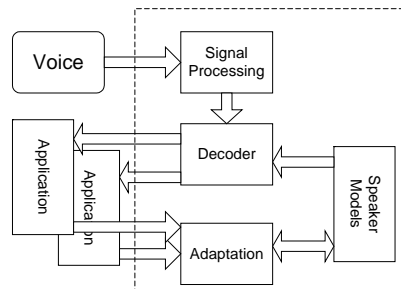


Figure 1.2 Basic system architecture of a speech recognition system [12]. Huang et al. p.5

# Basic Architecture

- Speaker Recognition System



## A simple speaker/speech recognition system

- Training
  - Use a set of training speech to construct models for each speaker or word.
- Developing
  - Use a disjoint set of test speech to determine how well the system works and tune it.
- Evaluating
  - A second test set which is independent of the development set is tested.

## Feature Vectors

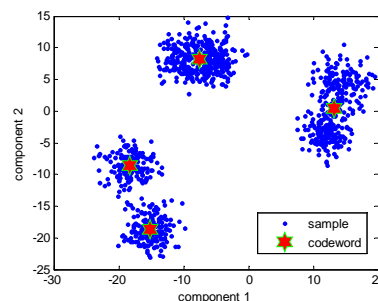
- What is a feature?
- What are feature vectors?
- Assume (for now)
  - Feature vector  $x_i$  extracted every 10 ms. from a speech signal.
  - For simplicity, let  $x_i \in \mathcal{R}^2$

## VQ Speech and Speaker Recognition (we can do much better)

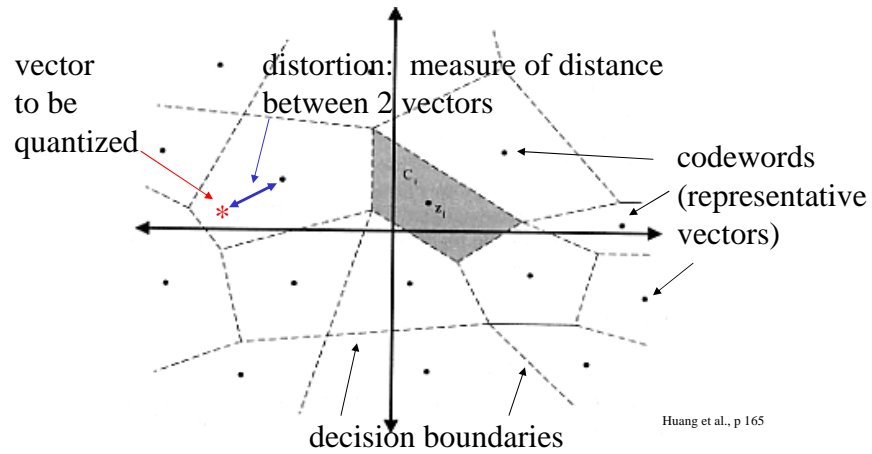
- Popular approach in the 1980s
- Abstraction of nearest neighbor learning
- Instance of the k-means algorithm
- Current usage:
  - Initializing other types of models
  - Variants of VQ (e.g. learning VQ)

## VQ fundamental ideas

- k-means is used to find vectors which are representative of clusters.
- Set of representative vectors is called a codebook.



## Two dimensional VQ partition



## VQ

### Quantization:

Assigns (classifies) a vector to one of a known set of clusters.

Returns codeword (index) which represents the cluster.

### Notes:

We do not know what the cluster represents\*.

How is this helpful for building a recognition system?

# VQ

- Suppose we train one VQ model for each class we wish to recognize.
- How could we use this to our advantage?

# VQ recognition

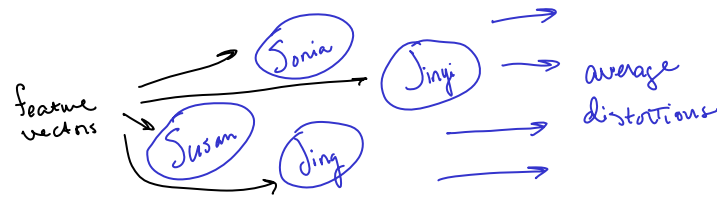
- Training
  - For each speaker or word, determine the k-means from that person's training data.
  - This “codebook” will represent the speaker (or word).



4 speakers →  
4 codebooks,  
each with k-means

## VQ recognition

- Testing
  - Check the average dissimilarity between test data and each code book.



- Label the test as the speaker or word whose code book was most similar (smallest average dissimilarity).

## Quantizing a vector (choosing the representative codeword)

- A vector can be quantized to a codeword as follows:

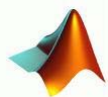
$$q(\vec{x}) = \vec{z}_i \leftrightarrow i = \arg \min_{1 \leq k \leq K} d(\vec{x}, \vec{z}_k)$$

- For classification purposes, we will frequently be interested in the minimum distortion instead of the codeword index.



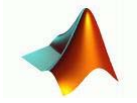
## Matlab – Matrix Laboratory

- Language for numeric computation
- Features
  - dynamic typing
  - vast library of mathematical functions
  - object oriented support
  - rich graphics & GUI
  - interface for C/C++/Fortran/Java



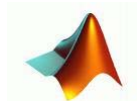
## Matlab – Data types

- Matrices
  - double precision by default
  - other types possible
- Strings (special case matrix)
- Collections



## Matlab data types

- matrix
  - denoted by square brackets
    - comma/space separate columns
    - new line/semi-colon separate row
  - indexing
    - Parenthesized list of indices  
`salinity_profile(longidx, latidx, depthidx)`
    - submatrices possible by specifying ranges  
`confusion(:, end-3:end)`  
`confusion(1:2:end, :)`
  - `size(confusion)` – vector with [#rows, #cols]



## Matlab data types

- strings – single quoted  
`name = 'Ludovic'`
- arrays – collections
  - structures (homogenous)
  - cell (heterogeneous)

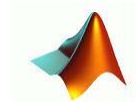
## Structure arrays – homogenous data

- structures – use . followed by name

```
s(idx).name = 'one';  
s(idx).model = ...  
train(feature_vecs);
```

- stand-alone structure

```
results.error_rate = ...  
test(feature_vecs);
```



## Cell arrays – heterogeneous data

- Purpose

– Any time elements are not of the same type or size, e.g.

- Arrays of character matrices

```
speakers = {'Julio'; 'Sue'; 'Ludovic'; 'Joanne'}
```

- Arrays of varying structures

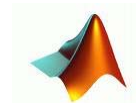
```
models{i}.
```

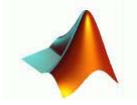
– denoted by { }

– collections of text strings

speakers{1} would display Julio

– other objects which may be different, e.g. matrices of different sizes



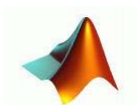


# Operators

- matrix arithmetic:

- addition +
- subtraction –
- multiplication \*
- left / and right \ division, e.g.  
 $A \setminus x = b \rightarrow x = A^{-1}b$
- exponentiation ^

```
for i = 1 to n % This will work...
  for j = 1 to m % faster & easier
    for k = 1 to m
      C(i,j) = C(i,j) + A(i,k) * B(k,j)
    end
  end
end
C = A * B
```



# Operators

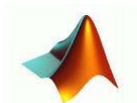
- Element by element
  - .\* ./ .^
- Assignment: =
- Relational operators
  - < <= > >=
  - == ~=
- Logical operators
  - and: &
  - or: |
  - not: ~\
- Matrix concatenation
  - New = [Old1; Old2];





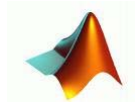
## Syntax and Constants

- commands
  - end with newline
  - line continuation ‘...’
  - semicolon at end suppresses output
- constants:
  - [ ] Inf -Inf i j



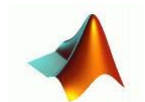
## Functions

- Pass by value semantics
- Multiple outputs are permitted
- Input arguments are positional
- Most Matlab functions operate on column oriented data.



## Matlab functions

- A few useful functions
  - isempty
  - mean
  - sum
  - plot
  - plot3
  - randperm – randomize 1:N
  - find
  - zeros
  - min/max (can return indices as well as values)
  - size/length



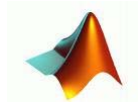
## Defining functions

- Stored in .m files
- Example: foobar.m

```
function [out1, out2] = foobar(in1)
% [out1, out2] = foobar(in1)
% foobar does ...
```

Matlab code to compute outputs, storing them to variables out1 and out2.

*Comments immediately following the command will be displayed when you type help foobar.*



## Control structures

if/else/elseif:

```
if ~ isempty(x)
    result = sum(x);
else
    error('Nothing in x')
end
```

while:

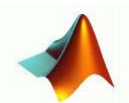
```
while n < 50
    n = n+1;
end
```

for:

```
for x = 1:2:N
    fprintf('x = %d\n', x)
end
```

switch:

```
switch lower(method)
case {'edge', 'edge2'}
    disp('Method is edge detection')
case 'bimodal'
    disp('Method is bimodal')
otherwise
    error('Unknown method %s. ', ...
        method)
end
```



## Simple data structures

```
Speakers{1}.feature_src = {'a.mfcc', 'b.mfcc',  
    'c.mfcc', 'd.mfcc', 'e.mfcc'};
```

```
Speakers{1}.id = 'lm'; % speaker name
```

```
Speakers{1}.train = [1 2 3];
```

```
Speakers{1}.test = [4 5];
```

Easy to use, e.g.:

```
for s = 1:length(Speakers)
    model{s} = train(Speakers{s});
end
```

## A simple speaker recognition system in Matlab

- We can read feature data produced by the hidden Markov model toolkit (HTK) with the custom function

```
[data, info] = spReadFeatureDataHTK('filename')
```

- data will be an  $D \times N$  matrix where
  - $D$  dimension of feature vectors
  - $N$  number of feature vectors
- info contains meta information about the feature set

## VQ Creating code books

Select an initial codebook of  $N$  codewords

done = false;

old\_distortion = compute average distortion for all training vectors

while not done

    For each training vector  $t_i$

        Compute the distortion between  $t_i$  and each code word  $c_j$

    Partition training vectors by minimum distortion codeword

    Compute new code words by taking centroid of each partition

    distortion = compute average distortion for all training vectors

    done = distortion / old\_distortion > threshold

    old\_distortion = distortion

## Distortion - Euclidean

- Euclidean distortion
  - Square of “map distance” between two points

$$d_{Euclidean}(\bar{x}, \bar{z}) = (\bar{x} - \bar{z})^t (\bar{x} - \bar{z}) = \sum_{i=1}^d (x_i - z_i)^2$$

- Assumes that all dimensions are equally important/scaled.
- How would we write this in Matlab?

## VQ Recognizer

For  $m_i$  in the set of code books

$$\text{distortions}(m_i) = 0$$

For each  $t_j$  in the set of test vectors

$$\text{distortions}(m_i) = \text{distortions}(m_i) + \text{distortion of smallest distorting codeword in } m_i.$$

Compute average distortions

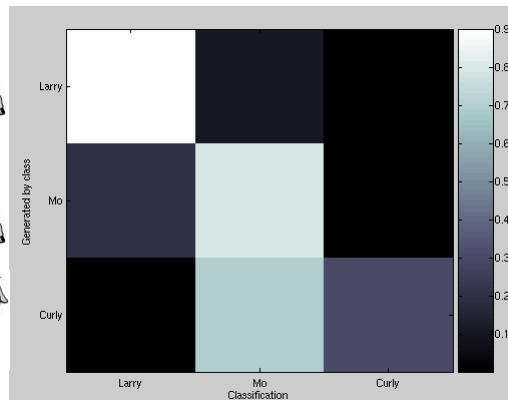
Select smallest average distortion codebooks as speaker label.

## Reporting the results

- What percentage of tests were incorrectly identified?
- Were there any patterns to the errors?
- What type of confidence do we have in the results?

## Speaker Recognition Of goats, sheep, and wolves...

- Confusion matrices



## Producing a confusion matrix in Matlab

```
Confusion = [.9 .1 0; .2 .8 0; 0 .7 .3]
colormap(bone); % specify suitable colors
imagesc(Confusion); % display as bitmap
[Rows, Cols] = size(Confusion);

set(gca, 'xtick', 1:Rows)
set(gca, 'ytick', 1:Cols);
colorbar % display scale

xlabel('Classification') % Label axes
ylabel('Generated by class')
```

## Analyzing the results

- Why did the errors occur?
  - Acoustic similarities/differences?
    - speaker
    - environment
    - collection
  - Problems with the representation?
    - feature set
    - model
  - Adequate training data?