CHPATER 1

FUNDAMENTALS OF SPEECH RECOGNITION

1.0 Introduction

- Automatic speech recognition (ASR) can be defined as the independent, computer-driven transcription of spoken language into readable text in real time.
- Automatic Speech Recognition (ASR) is the process of converting a speech signal to a sequence of words, by means of an algorithm implemented as a computer program.
- Having a machine to understand fluently spoken speech has driven speech research for more than 50 years.
- Although ASR technology is not yet at the point where machines understand all speech , in any acoustic environment, or by any person, it is used on a day-to-day basis in a number of applications and services.
- The ultimate goal of ASR research is to allow a computer to recognize in real-time, with 100% accuracy, all words that are intelligibly spoken by any person, independent of vocabulary size, noise, speaker characteristics or accent.
- Today, if the system is trained to learn an individual speaker's voice, then much larger vocabularies are possible and accuracy can be greater than 90%.
- Commercially available ASR systems usually require only a short period of speaker training and may successfully capture continuous speech with a large vocabulary at normal pace with a very high accuracy.
- Recognition software can achieve between 98% to 99% accuracy if operated under optimal conditions. `Optimal conditions' usually assume that users: have speech characteristics which match the training data, can achieve proper speaker adaptation, and work in a clean noise environment (e.g. Quiet space).

IDOL

- The Disciplines that applied to most of the speech recognition problems:

1. **Signal Processing:**

- The process of extracting relevant information from the speech signal in an efficient and robust manner.
- Using this process we can characterize the time-varying properties of the speech signal as well as various types of signal preprocessing and post processing to make the speech signal robust.

2. **Physics(acoustics):**

- The science of understanding the relationship between the physical speech signal and physiological mechanisms or we can say that human vocal tract mechanism that produces speech and with which the speech is perceived.

3. **Pattern recognition:**

- The set of algorithms used to cluster data to create prototypical patterns and to compare a pair of patterns on the basis of feature measurement.

4. **Communication and information theory:**

- The methods for detecting the presence of particular speech pattern, set of coding and decoding algorithm used to search a large but finite grid for best path corresponding to a “best” recognized sequence of words.

5. **Linguistics:**

- The relationship between sounds (phonology), words in a language (syntax), meaning of spoken words (semantics), and sense derived from the meaning (pragmatics).

6. **Physiology:**

- Understanding of the higher-order mechanisms within the human central nervous system that account for speech production and perception in human beings.

7. **Computer science:**

- The study of efficient algorithms for implementing, in software and hardware, the various methods used in a practical speech-recognition system.

8. **Psychology:**

IDOL

- The science of understanding the factors that enable a technology to be used by human beings in practical tasks.

1.1 The Paradigm For Speech Recognition

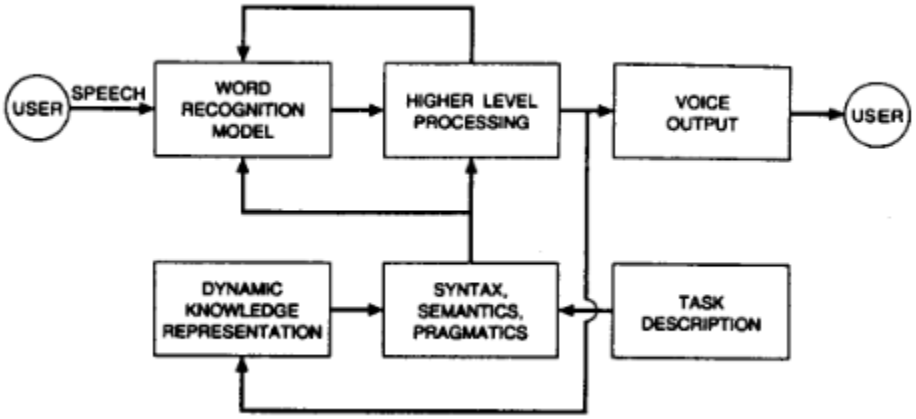


Figure 1.1 General block diagram of a task-oriented speech recognition system [1]

The above diagram it consist of:

- Word recognition model:
  - First the spoken o/p is recognized
  - Then speech signal is decoded into a series of words that are meaningful according to syntax, semantics, and pragmatics.
- Higher-level processor:
  - the meaning of the recognized words is obtained.
  - The processor uses a dynamic knowledge representation to modify the syntax, semantics and the pragmatics according to the context of what it has previously recognized.
- The feedback from Higher-level processor reduces complexity of recognition mode by limiting the search for valid input sentences from the user.
- The system responds to the user in the form of a voice output.

1.2 Outline

2. The speech signal

- In this chapter we will see
  - The speech production and perception processing human beings
  - The different speech sound that has been characterized by spectral and temporal properties that depend upon acoustic phonetic features of sound
  - The three most common approaches to speech recognition
    1. Acoustic-phonetic approach
    2. Pattern recognition approach
    3. Artificial intelligence approach
  - The strength and weakness of this approach

3. Signal processing and analysis method for speech recognition

- In this chapter we will see
  - Two fundamental approaches to spectral analysis and compare and contrast these system in terms of robustness to speech sound and required computation
    1. Filter bank approach
    2. Linear predictive methods
  - The popular source coding technique: Vector Quantization (VQ) in which we created a codebook to represent anticipated range of spectral vectors
  - A spectral analysis model to mimic the processing of human auditory. The model is called as Ear model

4. Pattern comparison technique

- In this chapter we will see
  - Three fundamental aspect of comparing pair of speech patterns
    1. Basic concept of detecting speech i.e. Speech detection

IDOL

2. Computing measure of the local distance or similarity of pair of spectral representation of short time piece of speech signal i.e. Distance of distortion measure
3. Temporally aligning and Globally comparing pair of speech patterns corresponding to different speech utterance i.e. Dynamic time warping algorithm

- And How basic pattern-comparison technique can be combined in a uniform framework for speech recognition

5. Speech recognition system design and implementation issues

- In this chapter we will see
  - How speech recognizer are trained and How we can enhance the basic recognition procedure
    - By adding feature,
    - By exploiting a pre-processor,
    - By use of methods of adaption
    - Or by post processing the recognizer outputs using set of pattern discriminators
  - And discuss the various ways of recognizing speech in adverse environments

6. Theory and implementing of hidden markov models

- In this chapter we will see
  - Aspect and theory of implementation of set of modelling techniques referred to as hidden markov modelling
  - Included within these techniques are
    - The algorithm for scoring a statistical model against speech pattern,

IDOL

- The techniques for aligning the model with the speech pattern so as to recover an estimate of the alignment path between different speech sounds and different model states,
- And the techniques for estimating parameters of the statistical models from a training set of utterances of the sounds being modelled.

- Discuss the practical aspects of building hidden Markov models,
  - Including the issues of scaling of data,
  - Handling of multiple observation sequences,
  - Providing initial estimates of model parameters,
  - And combating the problems of insufficient training data
- Discuss the how a simple, isolated word recognizer would be implemented using hidden Markov models.

7. Speech Recognition Based on Connected Word Models.

- In this chapter we will see
  - The basic set of techniques developed for recognizing an isolated word or phrase can be readily extended to recognizing a sequence of words (e.g., a string of digits of a credit card number) spoken in a fluent or connected manner
  - Describe three "optimal" approaches to solving the recognition part of connected word-recognition problems:
    1. The two-level dynamic programming method,
    2. The level building method,
    3. The time synchronous level building (or the one-pass) method
  - Discuss the properties, and the relative strengths and weaknesses of each method
  - How we can optimally train connected word systems, even if isolated versions of the vocabulary words are not available.

IDOL

8. Large Vocabulary Continuous Speech Recognition.

- In this chapter we will see:
  - The issues in applying speech-recognition technology to the problem of recognizing fluently spoken speech with vocabulary sizes of 1000 or more words
  - It is shown that a number of fundamental problems must be solved to implement such a system including,
    - The choice of a basic sub word speech unit (from which words, phrases. And sentences can be built up),
    - An effective way of modelling the basic speech unit,
    - A way of deriving models of the unit
    - A way of designing and implementing a word lexicon
    - A way implementing task syntax
    - A way of implementing the overall part of the system
    - And way imposing task semantic into system
  - We concentrate on The issues involved in building large vocabulary recognition systems

9. Task oriented Application of Automatic speech recognition

- In this chapter we will see:
  - Overview of how we can apply the ideas discussed so far to building a real speech recognition system.
  - It includes discussions of how we would evaluate recognizer performance and how we might decide whether a proposed task is viable for speech recognition.
  - Also discuss a set of broad classes of applications, which appear to be the most promising ones at this time, along with typical examples of how recognizers have been successfully employed within these broad classes.

IDOL

1.3 Brief History of speech recognition research

1950 **:**

- The earliest attempts to devise systems for automatic speech recognition by machine were made in this years.
- when various researchers tried to exploit the fundamental ideas of acoustic-phonetics.

1952:

- In 1952. at Bell Laboratories, Davis, Biddulph, and Balashek built a system for isolated digit recognition for a single speaker .
- The system relied heavily on measuring spectral resonances during the vowel region of each digit

1956: .

- In an independent effort at RCA Laboratories in 1956, Olson and Belar tried to recognize 10 distinct syllables of a single talker, as embodied in 10 monosyllabic words. The system again relied on spectral measurements primarily during vowel regions

1959:

- at University College in England, Fry and Denes tried to build a phoneme recognizer to recognize four vowels and nine consonants . They used a spectrum analyzer and a pattern matcher to make the recognition decision. A novel aspect of this research was the use of statistical information about allowable sequences of phonemes in English to improve overall phoneme accuracy for words consisting of two or more phonemes

- Another effort of note in this period was the vowel recognizer of Forgie and Forgie, constructed at MIT Lincoln Laboratories in 1959, in which 10 vowels embedded in a /bl-vowel-/t| format were recognized in a speaker-independent

IDOL

- manner. Again a filter bank analyzer was used to provide spectral information, and a time varying estimate of the vocal tract resonances was made to decide which vowel was spoken.

1960

- several fundamental ideas in speech recognition surfaced and were published. However. the decade started with several Japanese laboratories entering the recognition arena and building special-purpose hardware as part of their systems. One early Japanese system described by Suzuki and Nakata of the Radio Research Lab in Tokyo, was a hardware vowel recognizer.
- An elaborate filter bank spectrum analyzer was used along with logic that connected the outputs of each channel of the spectrum analyzer to a vowel-decision circuit, and a majority decision logic scheme was used to choose the spoken vowel.

1962

- Another hardware effort in Japan was the work of Sakai and Doshita of Kyoto University in 1962, who built a hardware phoneme recognizer .
- A hardware speech segmented was used along with a zero crossing analysis of different regions of the spoken input to provide the recognition output

1963

- third Japanese effort was the digit recognizer hardware of Nagata and coworkers at NEC Laboratories in 1963 .
- This effort was perhaps most notable as the initial attempt at speech recognition at NEC and led to a long and highly productive research program

1970

- In the 1970 speech-recognition research achieved a number of significant milestones.

IDOL

- First the area of isolated word or discrete utterance recognition became a viable and usable technology based on fundamental studies by Velichko and Zagoruyko in Russia , Sakoe and Chiba in Japan, and Itakura in the United States.
- The Russian studies helped advance the use of pattern-recognition ideas in speech recognition; the Japanese research showed how dynamic programming methods could be successfully applied; and Itakura`s research showed how the ideas of linear predictive coding (LPC). which had already been successfully used in low-bit-rate speech coding. could be extended to speech- ecognition systems through the use of an appropriate distance measure based on LPC spectral parameters.
- Another milestone of the 1970 was the beginning of a longstanding. highly successful group effort in large vocabulary speech recognition at IBM in which researchers studied three distinct tasks over a period of almost two decades, namely the New Raleigh language for simple database queries. the laser patent text language for transcribing laser patents, and the office correspondence task, called Tensors [161, for dictation of simple memos.

- Finally, at AT&T Bell Labs, researchers began a series of experiments aimed at making speech-recognition systems that were truly speaker independent

- To achieve this goal a wide range of sophisticated clustering algorithms were used to determine the number of distinct patterns required to represent all variations of different words across a wide user population. This research has been refined over a decade so that the techniques for creating speaker-independent patterens are now well understood and widely used.

1980

- the problem of connected word recognition was a focus of research in the 1980. Here the goal was to create a robust system capable of recognizing a

IDOL

fluently spoken string of words (e.g.,digits) based on matching a concatenated pattern of individual words.

- A wide variety of connected word-recognition algorithms were formulated and implemented, including the two-level dynamic programming approach of Sakoe at Nippon Electric Corporation (NEC), the one-pass method of Bridle and Brown at Joint Speech Research Unit (JSRU) in England , the level building approach of Myers and Rabiner at Bell Labs ,and the frame synchronous level building approach of Lee and Rabiner at Bell Labs.
- Each of these "optimal" matching procedures had its own implementations advantages, which were exploited for a wide range of tasks.

- Speech research in 1980s was characterized by a shift in technology from
- template-based approaches to statistical modeling methods-especially the hidden Markov model approach.
- Although the methodology of hidden Markov modeling (HMM) was well known and understood in a few laboratories, it was not until widespread publication of the methods and theory of HMMs, in the mid-1980s, that the technique became widely applied in virtually every speech-recognition research laboratory in the world.

- Finally the 1980s was a decade in which a major impetus was given to large vocabulary , continuous speech recognition system by the defence Advance Research Project Agency (DARPA) community which sponsored a large research program aimed at achieving high word accuracy for a 1000 word database management task

1990

- The DARPA program has continued into the 1990s with emphasis shifting to natural language front ends to the recognizer and the task shifting to retrieval of air travel information

IDOL

- At the same time speech recognition technology has been increasingly used within the telephone networks to automate as well as enhance operator service

References and Further Reading

[1] L.R. Rabiner , B. H. Juang , “Fundamentals of Speech Recognition”, Prentice-Hall, Englewood Cliffs, N.J.

IDOL

Chapter 2

The Speech Signal Part - A

OBJECTIVES

In this chapter you will learn

- How speech is produced and perceived in human beings?
- How speech production system works?
- We will see different classes of speech sound and its feature?
- What are the approaches to automatic speech recognition by machine?
- What are the pros and cons of such approaches?

2.1 INTRODUCTION

- Natural form of communication in humans being is speech. For common people speech is just the sound waves coming out of the human mouth and perceived through ears. But there is complex mechanism behind its production. The study of human speech production and perception mechanism is important for the development of devices for hearing aids, speech recognition, speech enhancement, speech simulation, speech modeling etc.

2.2 THE PROCESS OF SPEECH PRODUCTION AND PERCEPTION IN HUMAN BEINGS

- Here ,we will see how speech is produced and perceived in human beings.

IDOL

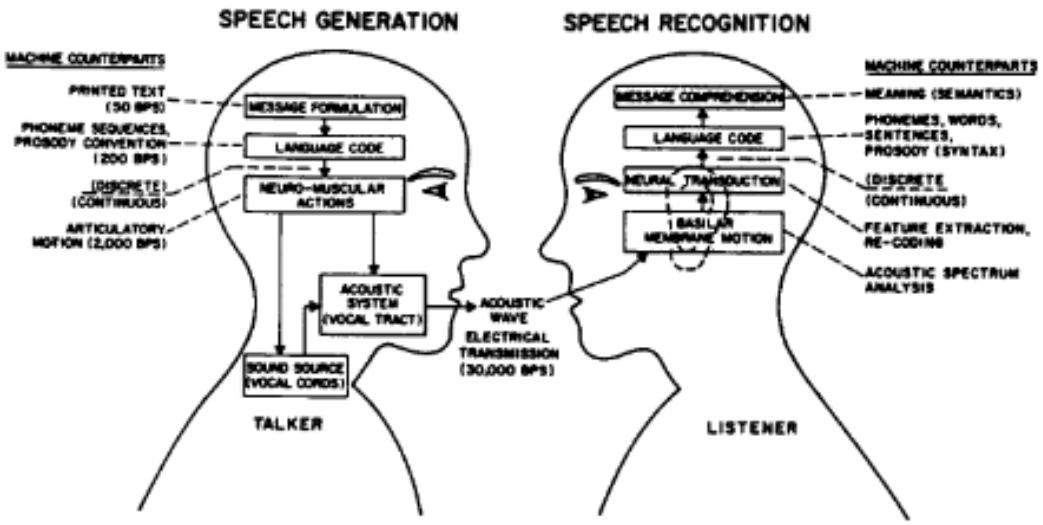


Figure 2.1 Schematic diagram of speech-production/speech-perception process (after Flanagan [unpublished]).

Figure 2.1 [1] : Schematic diagram of speech-production /speech-perception process

2.2.1 SPEECH PRODUCTION

- Speech production starts with the generation of idea or thought in mind about what to speak.This idea is then converted to language code. It means converting the printed text of the message into a set of phoneme sequences.
- Then the talker generates the control signals which vibrate the vocal cords and shape the vocal tract such that the proper speech sounds is produced.
- The control signals control all aspects of articulatory motion including control of lips, jaws, tongue, velum, lungs, vocal cords, glottis etc.
- These organs are flexible in nature and their shape and size alters on the command of control signals received from the brain, as per the type of speech and sound to be produced.

2.2.2 SPEECH PERCEPTION

- After the speech signal is generated through mouth or nasal cavity the speech signals are propagated to the listener, once the signal reaches to human ear the speech-perception process begins.
- Speech is perceived through sensory nerves connecting ear and the human brain.The speech waveform generated from the vocal tract is analyzed by the basilar membrane (Ear), thus spectrum analysis is performed on the continuous signal.
- The features such as duration of vocal cord vibration, intensity of the sound, resonant frequency of the vocal cord are extracted at the neural transduction stage
- The neural activity along the auditory nerve is converted to language code used for communication and finally the interpretation of message is done.

This is how the speech is produced and perceived in human beings

2.3 THE SPEECH PRODUCTION SYSTEM

- Human Vocal mechanism consist of two tract
  - Vocal Tract
  - Nasal Tract

- Vocal Tract
  - from opening of vocal cords to the lips.
  - Total length in average male is about 17cm
  - Cross sectional area of vocal tract is determined by the position of the tongue, lips, jaws and velum, which varies from zero to 20 cm^2.

IDOL

- Nasal Tract
  - from the velum to the nostrils.
  - Velum is the trapdoor like mechanism at the back of the mouth cavity
  - When the velum is lowered the nasal tract is acoustically coupled to the vocal tract to produce the nasal sound of speech.

- Schematic diagram of human vocal mechanism is shown below

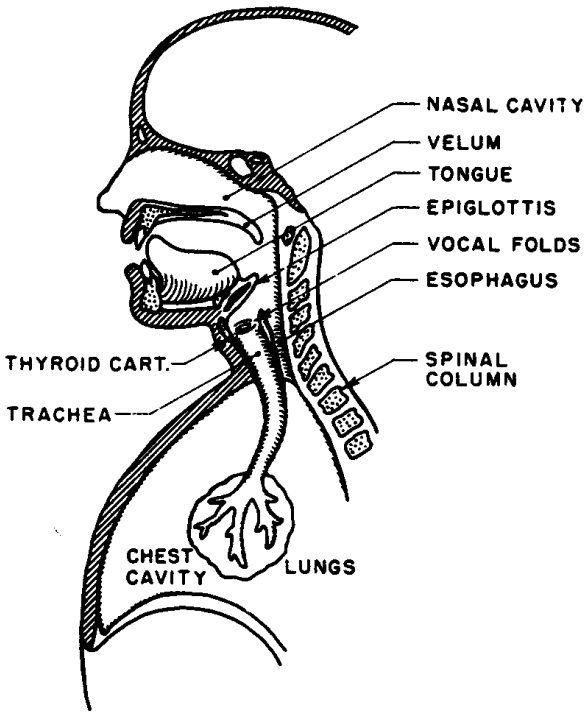


Figure 2.2 schemtic view of the vocal mechanism [1]

- In the normal breathing mechanism when we breathe, the air enters into the lungs.

- When we want to produce the sound, the air is drive out from the lungs through the trachea. Trachea which serves the principle passage for conveying the air to and from the lungs.
- When air is drive out though the trachea the vocal cords within the larynx are caused to vibrate by the air flow.
- Then the air flow is chopped into quasi-periodic pulses which are then modulated in frequency in passing though the pharynx (vocal throat cavity), mouth cavity and possibly nasal cavity.
- Depending on the position of the jaws, tongue, velum, lips, mouth different sound is produced.

- Simplified representation of the complete physiological mechanism of the speech is shown below

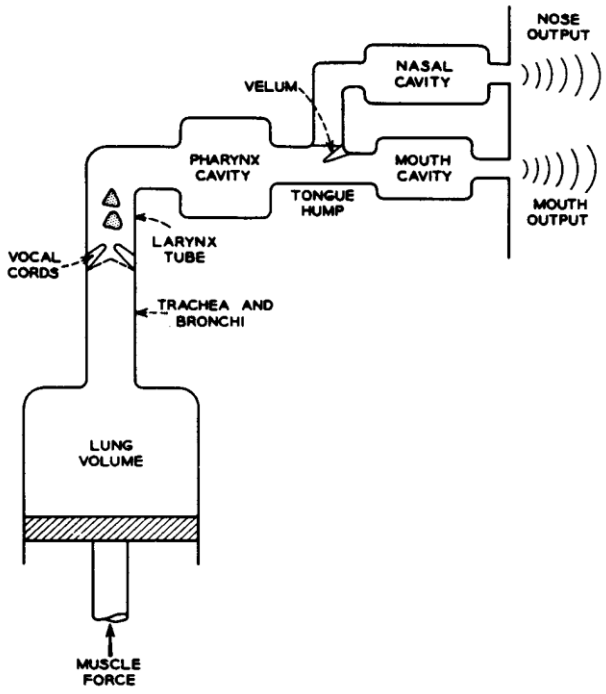


Figure 2.3 schematic representation of the complete physiological mechnism of speech prodution [1]

- The lungs and associated muscle act as source of air for exciting the vocal mechanisms.
- The muscles pushes air out of the lungs though the trachea and bronchi
- When vocal cords are tensed the air flow causes them to vibrate, producing the sound. This is called the voiced speech sound.
- When vocal cords are relaxed, air must pass though the constriction in the vocal tract and thereby becomes the turbulent producing the so called unvoiced sounds.
- Speech is produced as a sequence of sounds. Hence the state of the vocal cords as well as the position, shape and sizes of the various articulators changes over time to reflect the sound being produced.

2.4 REPRESENTING SPEECH IN TIME AND FREQUENCY DOMAIN

- The speech signals are slowly time varying signals,
- when the speech signals are examined over the short period of time (between 5 to 100msec) characteristics of it are fairly stationary, when it examined over a long period of time(more than 100msec) it reflect the characteristic change of different sounds being spoken
- Speech events are classified as follows
  - Silence(s) : when no speech is produced
  - Unvoiced (U) : vocal cords are not vibrating , resulting speech waveform is aperiodic or random in nature
  - Voiced(V) : when air flows from lungs the tensed vocal cords vibrates periodically, resulting speech waveform is quasi-periodic.

Below diagram[1] shows the time waveform corresponding to initial sound in the phrase “IT’s Time...”

IDOL

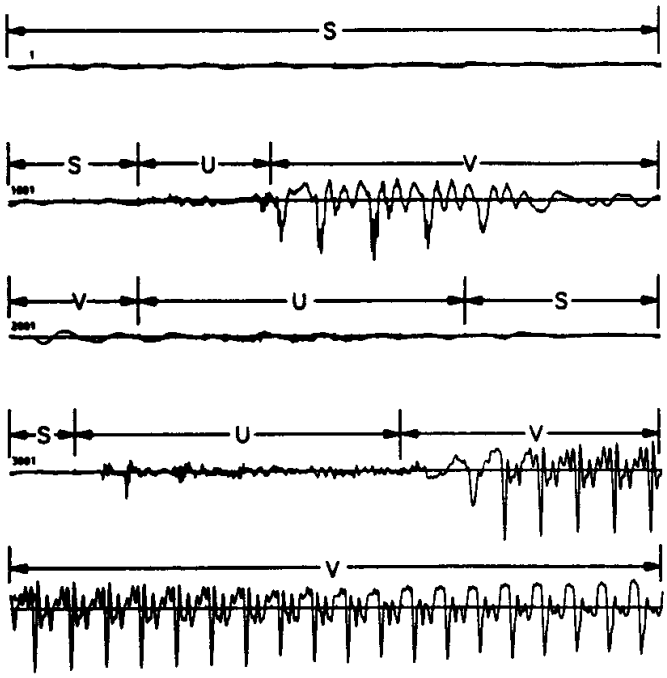


Figure 2.4 waveform of the utterance “speech” [1]

In Above diagram each line of the waveform corresponds to 100milisecond so the entire plot is about 0.5sec.

Before the speech begins the waveform is classified as silence(S). Then the unvoiced(U) is seen before the vowel in word It’s. After that the voiced region (V) then the unvoiced(U) aspiration for devoicing of the vowel, followed by silence region(S) and then long unvoiced(U) region for releasing the /t/ followed by /s/, followed by the /t/ in time and then very long voicing (V) region corresponding to the word time.

- An alternative way of representing the speech signal is via spectral representation

- Most popular representation of this type is the sound spectrogram in which a three-dimensional representation of the speech intensity, in different frequency bands, over time is portrayed.
- Below diagram[1] shows a wideband spectrogram in the first panel, a narrowband spectrogram in the second panel, and a waveform amplitude plot in the third panel, of a spoken version of the utterance "Every salt breeze comes from the sea" by a male speaker

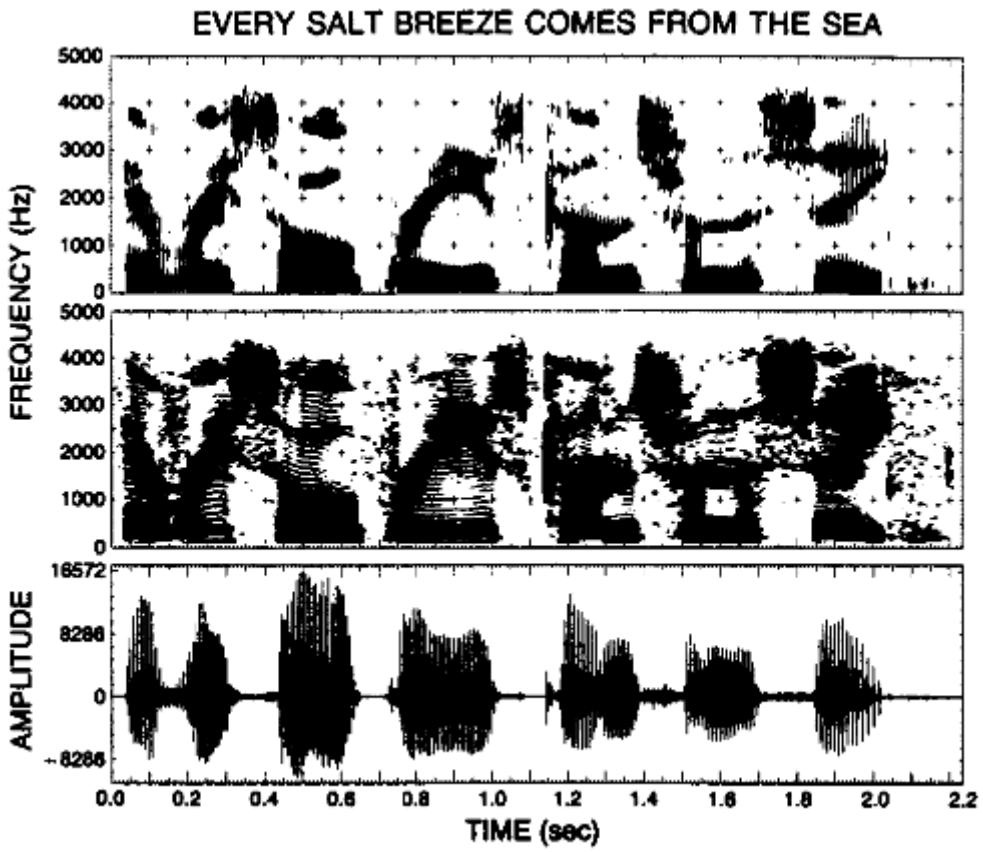


Figure 2.5 wideband and narrowband spectogram and apeech amplitude [1]

- In wideband spectrogram the spectral intensity at each point in time is indicated by the intensity (darkness) of the plot at a particular analysis frequency.
- Because of the relatively broad bandwidth of the analysis filters. the spectral envelope of individual periods of the speech waveform during voiced sections are resolved and are seen as vertical striations in the spectrogram.

- In narrowband spectrogram Because of the relatively narrow bandwidth of the analysis &Item, individual spectral harmonics corresponding to the pitch of the speech waveform.during voiced regions, are resolved and are seen as almost-horizontal lines in the spectrogram.
- During periods of unvoiced speech, we see primarily high-frequency energy in the spectrograms ;during periods of silence we essentially see no spectral activity.
- Third way of representing the time varying signal is via parameterization o spectral activity based on model of speech production ,because the Vocal tract is tube of varying cross sectional area, the acoustic theory tells that the transfer function of energy from the excitation source to the output can be described in terms of theory of resonance. Such resonances are called formants for speech.
- The formant frequency representation is a highly efficient representation of time varying characteristic of speech.
- The problem with this is difficulty of reliably estimating the formant frequencies for low level voiced sound and difficulty of defining formants for unvoiced and silence regions.

2.5 SPEECH SOUNDS AND FEATURES

Most of the languages can be described by a set of distinctive phonemes. Phonemes can be represented by phonetic alphabets. For example, the phonemes of the American English can be represented by ARPABET of IPA (international phonetic alphabet)

- There are total 11 vowels, classified as front, mid or back corresponding to the position of the tongue hump in producing the vowels;
- 4 vowel combination or diphthongs;
- 4 semivowels broken down into 2 liquids and 2 glides;
- The nasal consonants;

- The voiced and unvoiced stop consonants
- The voiced and unvoiced fricatives
- Whisper
- Affricatives

2.5.1 VOWELS

- speech-recognition systems rely heavily on vowels recognition but in written text it is having very low importance
- Vowel sound is determined primarily by the position of the tongue
- Generally long in duration(as compared to consonants)
- A convenient and simplified way of classifying vowel articulatory configuration is in terms of the tongue hump position(front,mid,back)and tongue hump height (high, mid, low),where tongue hump is the mass of the tongue at its narrowest constriction within the vocal tract.
- According to this classification the vowels
    - /i/,/I/,/æ/,/ɛ/ and are front vowels
    - /a/,/ʌ/,/ɔ/ are mid vowels
    - /U/,/u/,/o/ are back vowels.

2.5.2 DIPHTHONGS

- Is the sound that is formed by the combination of two vowels,In which sound begins as one vowel and moves towards the another.

IDOL

- There are six diphthongs in American English, namely /ay/(as in buy), /aW/ (as in down), /ey/(as in bait), and /ƆY/ (as in boy), /o/ (as in boat), and /ju/ (as in you).

2.5.3 SEMIVOWELS

- Is the sound that intermediate between a vowel and consonant
- It is quite difficult to characterize.They are having the vowel like nature
- Example : /w/, /l/, /r/ and /y/

2.5.4 NASAL CONSONANT

- A nasal consonant is a type of consonant produced with a lowered velum in the mouth, allowing air to come out through the nose, while the air is not allowed to pass through the mouth
- The air is flows through the nasal tract with sound being radiated at the nostrils
- The three nasal consonant are distinguished by the place along the oral tract at which a total constriction is made.
    - For /m/ the constriction is at the lips
    - For /n/ the constriction is behind the teeth
    - For /ɳ / constriction is just forward the velum itself

2.5.5 UNVOICED FRICATIVES

- Produced by exciting the vocal tract by a steady air flow, which becomes turbulent in the region of a constriction in the vocal tract.
- which fricatives sound is produced is determined by the location of constriction
    - For fricative /f/ the constriction is near the lips,
    - For /θ/ constriction is near the teeth,

IDOL

- for /s/ it is near the middle of the vocal tract,
- for /sh/ it is near the back of the vocal tract

- The system for producing the unvoiced fricatives consists of a source of noise at a constriction, which separates the vocal tract into two cavities.
  - front cavity - Sound Is radiated from the lips.
  - back cavity - Sound Is radiated from nasals.

2.5.6 VOICED FRICATIVES

- The /v/, /z/, /zh/ are the counter parts of the unvoiced fricatives respectively
- The vocal cords are vibrating and thus one excitation sources is at the glottis.
- Since the vocal tract is constricted at some point forward of the glottis, the air flow becomes turbulent in the neighborhood of the constriction.

2.5.6 VOICED AND UNVOICED STOPS

- complete block of air flow followed by sudden release
- The voiced stop consonants /b/,/d/ and /g/ are transient noncontinuant sounds produced by building up pressure behind a total constriction somewhere in the vocal tract and then suddenly releasing the pressure.
  - For /b/ the constriction is the lips
  - For /d/ the constriction is at the back of the teeth
  - For /g/ it is near the velum
- During the period when there is total constriction in the tract, no sound is radiated from the lips

IDOL

- The unvoiced stop consonants /p/,/t/ and /k/ are similar to their voiced with one major exception.
- During the period of total closure of the tract, as the pressure builds up the vocal cords do not vibrate.

2.5.7 AFFRICATES AND WHISPER

- Affricates have a lot of high frequency noise, and either little or no lower frequency energy. They are easiest to identify in the spectrogram
    - /tS/as in chew
    - /J/ as in just
- Whispers are very noisy, ideally flat across all frequencies but influenced by the surrounding letters in reality
    - /h/ as in he

References and further Reading

[1] L.R. Rabiner , B. H. Juang , “Fundamentals of Speech Recognition”, Prentice-Hall, Englewood Cliffs, N.J.

IDOL

Chapter 3

The Speech Signal part – B

OBJECTIVES

In this chapter you will learn

- What are the approaches to automatic speech recognition by machine?
- What are the pros and cons of such approaches?

3.1 APPROACHES TO AUTOMATIC SPEECH RECOGNITION BY MACHINE

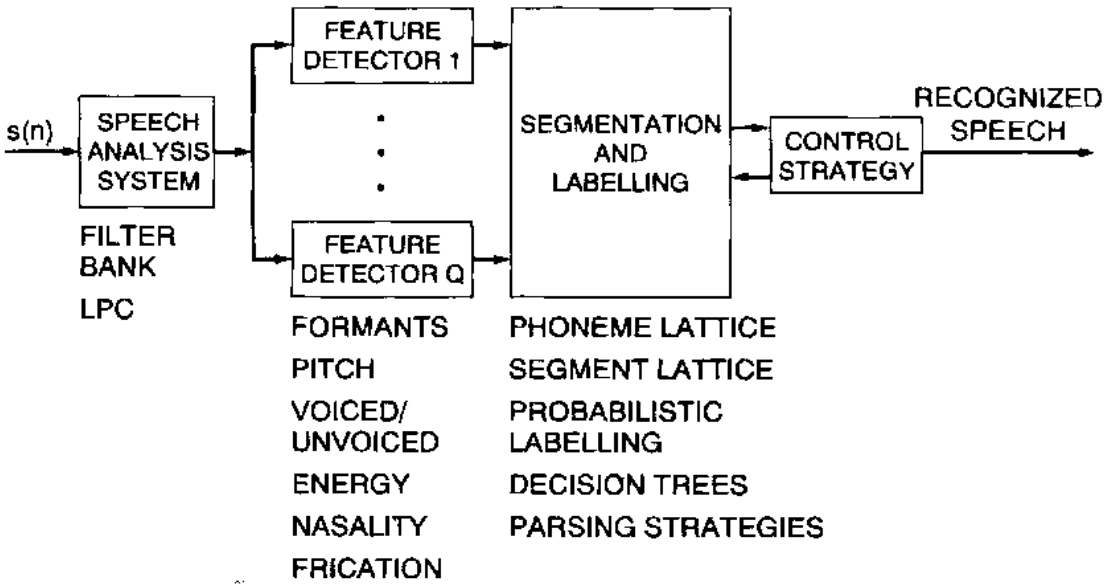
There are several approaches proposed for automatic speech recognition by machine with goal of providing some understanding as to the essentials of each proposed method, and the basic strength and weakness of each approach

Broadly speaking, there are three approaches to speech recognition namely

1. The acoustic phonetic approach
2. The pattern recognition approach
3. The artificial intelligence approach

3.1.1 ACOUSTIC-PHONETIC APPROACH TO SPEECH RECOGNITION

IDOL



acoustic-phonetic speech-recognition system.

Figure 3.1 – acoustic- phonetic speech-recognition system

- There are three steps in acoustic phonetic approach

1. Speech analysis
2. Feature detection
3. Segmentation and labeling

- Speech analysis
  - A step common to all approaches in speech recognition
  - Provides an appropriate(spectral) representation of the characteristics of the time-varying speech signal.
  - Most common technique of spectral analysis are the class of filter bank methods and the class of linear predictive coding(LPC) methods.

- Feature detection

- converts the spectral measurement to a set of features that describes the broad acoustic properties of the different phonetic units
- features proposed for recognition are nasality, friction, formant locations, voiced and unvoiced classification
- it is usually consist of a set of detectors that operate in parallel and use appropriate processing and logic to make the decision as to presence or absence, or value, of a feature.

- Segmentation and labeling

  - Here the speech signal is segmented into stable acoustic regions, followed by attaching one or more phonetic labels to each segmented region, resulting in a phoneme lattice characterization of the speech.
  - from phoneme lattice characterization of the speech, a lexical access procedure determined the best matching word and sequence of words.

- Issues in acoustic-phonetic approach

Following are the problems which account the lack of success in practical speech-recognition system

1. Requires the extensive knowledge of acoustic properties of phonetic units. This knowledge is incomplete and totally unavailable for steady vowels
2. The choice of features is made mostly based on ad hoc consideration
3. The design of the sound classifiers is also not optimal.
4. No well defined automatic procedure exists for tuning the method.
5. There is not even an ideal way of labeling the training speech

3.1.2 PATTERN-RECOGNITION APPROACH TO SPEECH RECOGNITION

IDOL

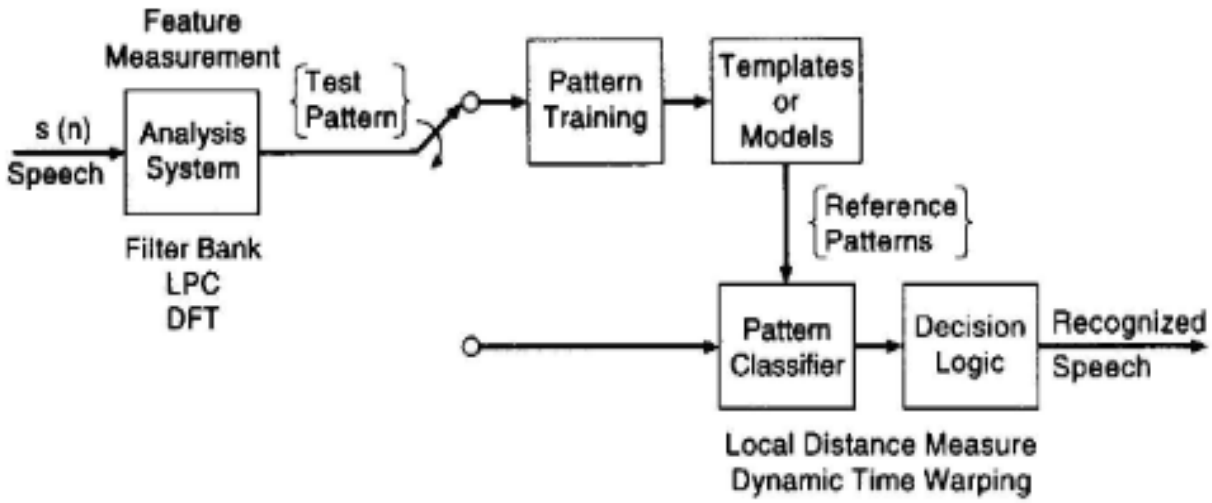


Figure 2.7 block diagram of patter recognition speech recognizer [1]

- The pattern recognition paradigm has four steps,

1. Feature measurement
2. Pattern training
3. Pattern classification
4. Decision logic

1. Feature measurement

- In which a sequence of measurement is made on the input signal to define the “test-pattern”
- For speech signals the feature measurement are usually output of some type of spectral analysis technique such as a filter bank analyzer, a linear predictive coding analysis, or discrete Fourier transform (DFT) analysis.

2. Pattern training

IDOL

- In which one or more test patterns corresponding to speech sound of the same class are used to create a pattern representative of the features of that class
- The resulting pattern generally called a reference pattern, can be exemplar or template, derived from some type of averaging technique.

3. Pattern classification

- In which the unknown test pattern is compared with each(sound) class reference pattern and a measure of similarity (distance between test pattern and each reference pattern is computed)
- To compare speech patterns (which consist of sequence of spectral vectors), we require both the spectral "distance" between two well-defined spectral vectors and global time alignment procedure which compensates for different rates of speaking (time scale) of the two patterns.

4. Decision logic

- In which reference pattern similarity scores are used to decide which reference pattern best matches the unknown test patterns

- The general strength and weakness of the pattern recognition model include the following

  Weakness

  1. System performance is sensitive to amount of training data, more the training data higher the performance
  2. Reference patterns are sensitive to speaking environment and transmission medium, because speech spectral characteristics are affected by transmission and background noise.

IDOL

3. No speech-specific knowledge is used to explicitly hence the method is relatively insensitive to choice of vocabulary words, task, syntax and task semantic
4. Computation load is high it is linearly proportional to the number of patterns being trained of recognized hence for large number sound classes it become prohibitive.

Strength

1. Because the system is insensitive to sound class, basic set of techniques developed for one sound class can be generally be directly applied to different sound class without modification to the algorithm
2. It is relatively straight forward to incorporate syntactic(and even semantic) constraints directly into pattern recognition structure, thereby improving accuracy and reducing computation.

3.1.3 ARTIFICIAL INTELLIGENCE APPROACHES TO SPEECH RECOGNITION

The Artificial Intelligence approach is a hybrid of the acoustic phonetic approach and pattern recognition approach. In this, it exploits the ideas and concepts of Acoustic phonetic and pattern recognition methods. Knowledge based approach uses the information regarding linguistic, phonetic and spectrogram. Thus for example, the AI approach to segmentation and labeling would be to augment the generally used acoustic knowledge with phonetic knowledge, lexical knowledge, syntactic knowledge, semantic knowledge, and even pragmatic knowledge.

We first define these different knowledge sources

Acoustic knowledge	-	Knowledge related to sound or sense of hearing
Lexical knowledge	-	Knowledge of the words of the language.
Syntactic knowledge	-	Knowledge according to syntax.
Semantic knowledge	-	Knowledge relating to meaning in language or logic.

IDOL

Pragmatic knowledge - Knowledge based on practical rather than theoretical considerations.

There are several ways to integrate knowledge sources within a speech recognizer

1. Bottom-up
2. Top-down
3. Blackboard

Bottom-up

- in which lowest level processes (e.g., feature detection, phonetic decoding) precede higher level processes (lexical decoding, language model) in a sequential manner

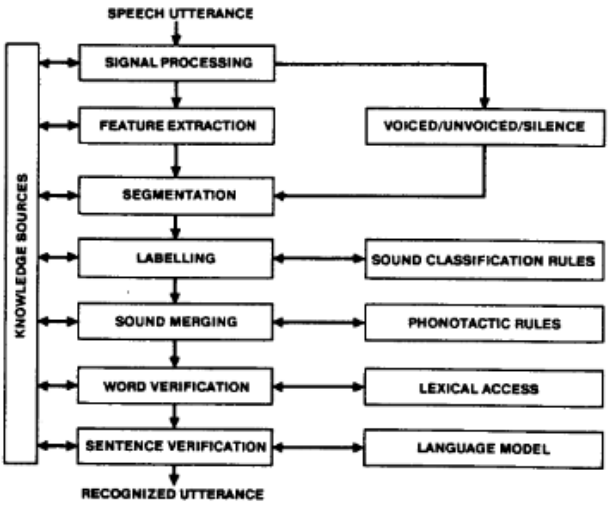


Figure 3.2 A bottom up approach to knowledge integration for speech recognition [1]

Top-down

- In which the language model generates word hypotheses that are matched against the speech signal and syntactically and semantically meaningful sentences are built up on the basis of the word match scores.

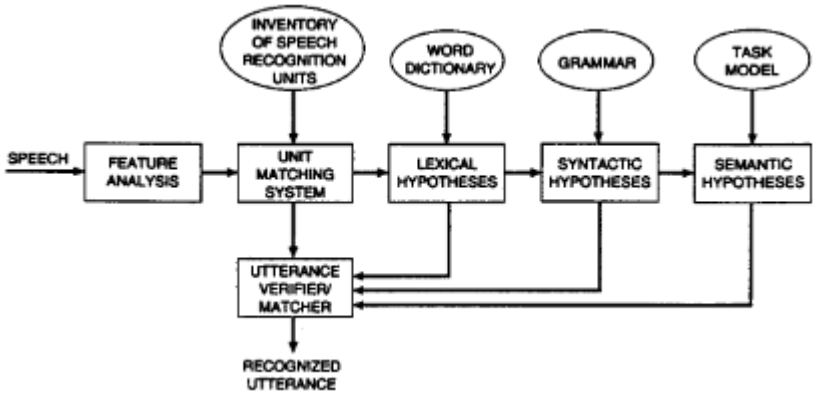


Figure 3.3 A top down approach to knowledge integration for speech recognition [1]

Blackboard

- In this approach all knowledge sources (KS) are considered independent;
- communication among knowledge sources is done via hypothesis-and-test paradigm;
- when the template specified by the Knowledge sources matches with the pattern on the blackboard, the system assigns the cost and utility consideration and overall ratings across all levels

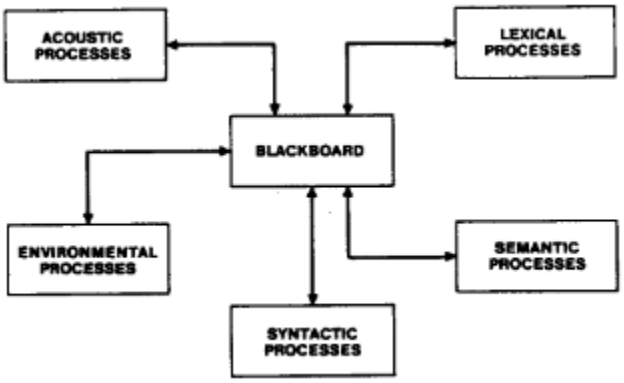


Figure : 3.4 A black board approach to knowledge integration for speech recognition [1]

3.1.4 NEURAL NETWORK AND THEIR APPLICATION TO SPEECH RECOGNITION

- A verity of knowledge sources are required in the AI approach to speech recognition
    - Two key concepts of AI are
    - Automatic knowledge learning
    - adaptation
- These concepts can be implemented by Neural Network approach.

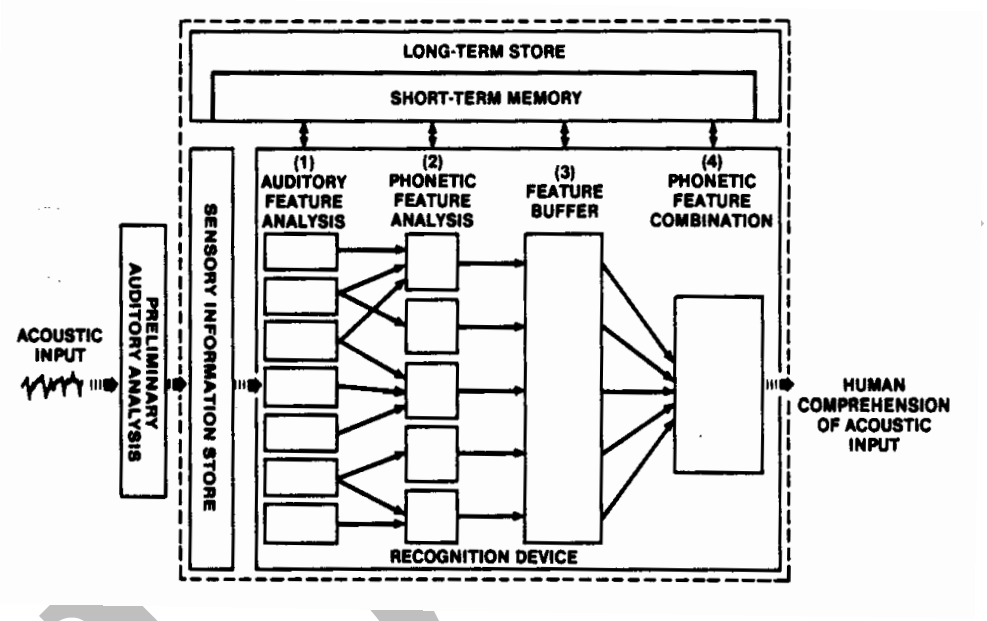


Figure 3.5 conceptual block diagram of a human speech understanding system [1]

- The acoustic input signal is analyzed by “ear model” that provides spectral information about the signal and stores it in a sensory information store.
- Other sensory information(eg from visual and touch) is available in the sensory information store and is used to provide several “feature level” descriptions of the speech.
- Both long-term and short term memory are available to various feature detectors. Finally after several stages of refined feature detection,
- Finally after several stages of feature detection, the final output of the system is an interpretation of the information in the acoustic input.

The system in above diagram shows human speech understanding system. The auditory analysis is based on our understanding of the acoustic processing in ear. The various feature analyses represent processing at various level in neural pathways to brain. The short and long term memory provides external control of neural processes in the ways that are not well understood. The overall form of the mode is that feed forward connectionist network- that is neural net.

3.1.4.1 BASIC OF NEURAL NETWORK

A neural network which is also known as connectionist model , a neural net, or a parallel distributed processing(PDP) model, is basically a dense interconnection of simple, non-linear, computation elements of type shown in figure.

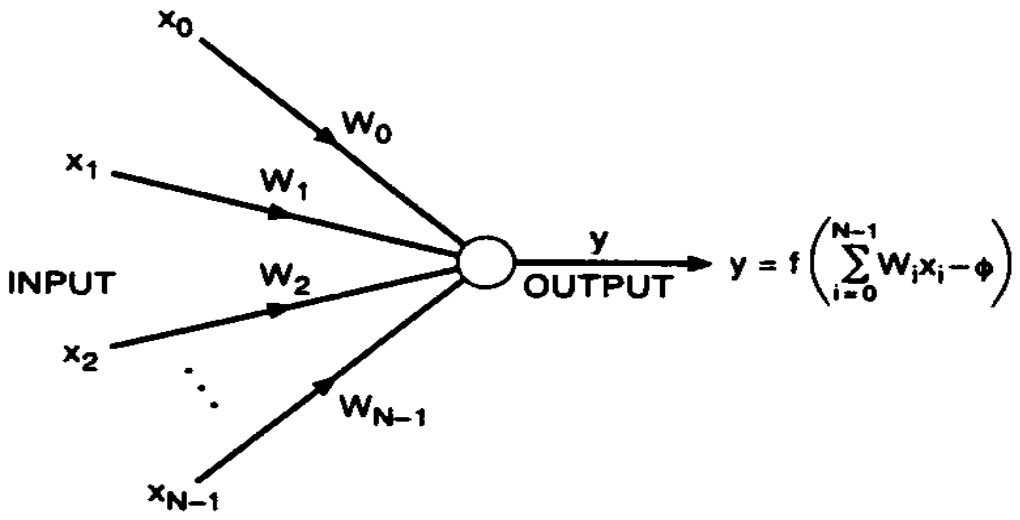


Figure 3.6 simple computation element of a neural network [1]

Where :

- $x_1, x_2, ..., x_n$ are number of inputs
- $w_1, w_2, \ldots w_n$ are summed with weight

- given the output y as

$$y = f\left(\sum_{i=1}^{N} w_i x_i - \phi\right),$$

- Φ is an internal threshold

- f is a non linearity of one of the types given below

1. Hard limiter

$$f(x) = \begin{cases} +1, x \le 0 \\ -1, x < 0 \end{cases}$$

or

2. Signal functions

$$f(x) = \tanh(\beta x), \beta > 0$$

Or

$$f(x) = \frac{1}{1+e^{-\beta x}}, \beta > 0$$

3.1.4.2 NEURAL NETWORK TOPOLOGIES

There are several issues in the design of so-called artificial neural networks(ANNs), Key issue is network topology- that is how the simple computational elements are interconnected.

There are three standard and well known topologies

- Single/multilayer perceptrons
- Hopfield or recurrent networks
- Kohonen or self-organising networks

IDOL

- Single layer/multilayer perceptron

- In this, the output of one or more simple computational elements at one layer from the inputs to a new set of simple computational elements of the next layer.

- Single layer perceptron has N inputs connected to M outputs in the output layer.

- Three layer perceptron has two hidden layers between the input and output layers.

- Nonlinearity at each layer distinguishes multilayer perceptron that enables the mapping between the input and output variables to posses certain particular classification/discrimination properties.

- Hopfield network

- it is recurrent network in which the input to each computational elements includes both the inputs as well as outputs.

- Basic equation for the i_{th} recurrent computational element is

$$y_i(t) = f\left[x_i(t) + \sum_j w_{ij} y_j(t-1) - \phi \right]$$

Where
- $y_i(t)$: output
- $x_i(t)$: input
- w_{ij} : the weight connection the i^{th} node and j^{th} node

Property of Hopfield network is When $w_{ij}=w_{ji}$ then $y_i(t)=y_i(t-1)$ for all i

These fixed points represents stable configurations of the network and can be used in applications that have fixed set of patterns to be matched e.g. printed letters

A simple interpretation of the Hopfield network is shown below.

IDOL

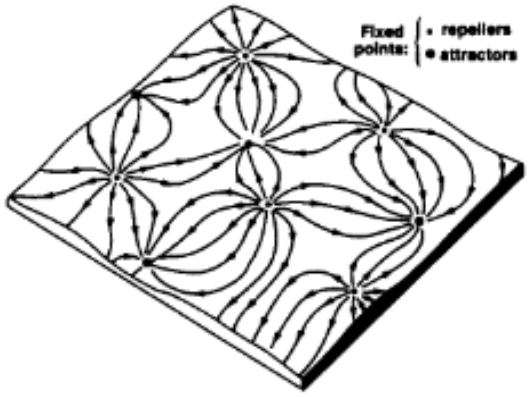


Figure 3.7 fixed point interpretation of the hopefield network [1]

Recurrent network has a stable set of attractors and repellers, each forming a fixed point in the input space.Every i/p vector x is either “attracted” to one of the fixed points or “repelled” from another of the fixed points.

The strength of Hopfield network is its ability to correctly classify “noisy” versions of patterns from stable fixed points.

- Kohonen

-It is self organizing feature map, which is a clustering procedure for providing a codebook of stable patterns in the input space that characterize an arbitrary input vector by a small number of representative clusters.

3.1.4.3 NETWORK CHARACTERISTICS

Four model characteristic must be specified to implement an arbitrary neural network

1. Number and type of inputs – the issue involved in the choice of inputs to neural network are similar to those involved in the choice of features for any pattern-classification system. They must provide the information required to make the decision required of the network

IDOL

2. Connectivity of network – this issue involves the of the network - that is the number of hidden layers and the number of nodes in each layer between input and output. There is no thumb rule as how large such hidden layers must be. If the hidden layer is large then it is difficult to train the network and if it is too small then network may not be able to accurately classify all the desired input patterns

3. Choice of offset – the choice of the threshold Φ, for each computation element must be made as part of the training procedure, which chooses values for the interconnection weights (w_{ij}) and the offset Φ.

4. Choice of nonlinearity – Experience indicates that the exact choice of the nonlinearity *f*, is not every important in terms of the network performance. However *f* must be continuous and differentiable for training algorithm to be applicable.

3.1.4.4 TRAINING OF NEURAL NETWORK PARAMETER

To Completely specify a neural network, values for the weighting coefficient and the offset threshold for each computation element must be determined based on a labeled set of training data.
For multilayer perceptrons a simple iterative, convergent procedure exists for choosing a set of parameters whose value asymptotically approaches a stationary point with a certain optimality property which is known as back propagations learning,

For a simple, single layer network, the training algorithm can be realized via the following convergence steps

Perceptron Convergence Procedure

1. Initialization: At time t = 0, set $w_{ij}(0)$, Φ_j to small random values

IDOL

2. Acquire Input: At time t, obtain a new input x = {x_1 , x_2, . . . , x_N} with the desired output y^x as,

$$y^x = \left\{ y_1^x, y_2^x, \ldots\ldots, y_M^x \right\}$$

3. Calculate output

$$y_j = f\left(\sum_{i=1}^{N} w_{ij}(t) x_i - \phi_j \right)$$

4. Adapt Weights: Update the weights as

$$W_{ij}(t+1) = w_{ij}(t) + T(t)[y_j^x - y_j].x_i$$

where the " step size" T(t) satisfies the constraints:

A.

$$\lim_{T\to\infty} \sum_{t=1}^{T} T(t) = \infty$$

B.

$$\lim_{T\to\infty} \sum_{t=1}^{T} T^2(t) < \infty$$

That is, compute the gradient of the error $\sum_{j=1}^{M} \left(y_j^x - y_j \right)^2$ in the direction of the weight $W_{ij}(t)$. (A conventional choice of T(t) is 1/t.)

5. Iteration: Iterate steps 2- 4 until:

$$W_{ij}(t+1)=w_{ij}(t), \qquad \forall i, t, j$$

The perception convergence procedure is a slow.

IDOL

Advantage of this algorithm is simple and is guaranteed to converge, in probability, under a restricted set of conditions on T(t).
But its speed of convergence in many cases is not sufficiently fast.

3.1.4.5 ADVANTAGES OF NEURAL NETWORKS

Neural networks have been given serious considerable for a wide range of problems (including speech recognition) for several reasons. These include the following:

1. They can readily implement a massive degree of parallel computation.
2. Possess Robustness.
3. Least sensitive of networks to noise or defects within the structure.
4. The connection weights of the network need not be constrained to be fixed they can be adapted in real time to improve performance.
5. Because of the nonlinearity within each computational element, a sufficiently large neural network can approximate (arbitrarily closely) any nonlinearity

References and further Reading

[1] L.R. Rabiner , B. H. Juang , “Fundamentals of Speech Recognition”, Prentice-Hall, Englewood Cliffs, N.J.

IDOL

CHAPTER 4

SIGNAL PROCESSING AND ANALYSIS METHODS FOR SPEECH RECOGNITION PART - A

4.1 Objective

In this chapter you will learn

- What are the two methods of spectral analysis?
- What is the vector quantization?
- What is auditory signal processing model?

4.2 Introduction

- Spectral analysis is defining the speech signal in different parameters these includes the short-term energy, zero crossing rates, level crossing rates and other related parameters
- The most important parametric representation of speech is short time spectral envelope.
- Speech analysis methods are the core of the signal processing front end in speech recognition system

4.2.1 Spectral analysis models

Let's have a look on following two models

1. Acoustic phonetic model
2. Pattern recognition model

In the above two model the pattern measurement step is common

There are three basic steps in Pattern recognition model

1. Parameter measurement
2. Pattern comparison

IDOL

3. Decision making

- parameter measurement

- Is to represent the relevant acoustic events in the speech signal in terms of the compact, efficient set of speech parameter
- The choice of which parameters to use is dictated by computational efficiency, type of Implementation , available memory consideration
- The way in which representation is computed is based on signal processing considerations

In the similar manner in Acoustic Phonetic Model first step i.e. Parameter Measurement is used in similar manner

The two dominant methods of speech analysis

1. Filter bank spectral analysis
2. Linear predictive coding(LPC)

The Structure of bank of filter model [1] are shown below

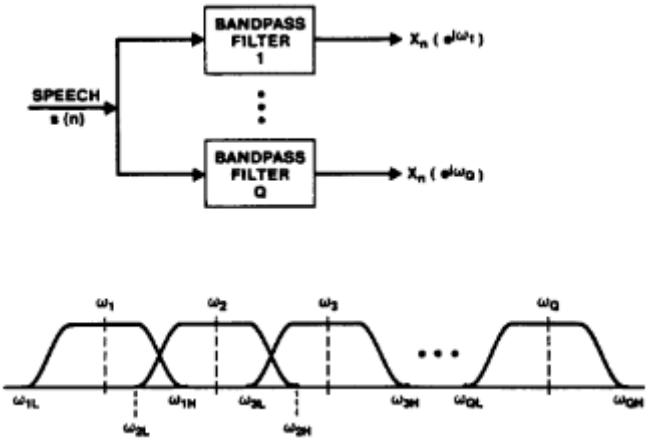


Figure 4.1 Bank-of-filter analysis model [1]

IDOL

4.3 THE BANK OF FILTERS FRONT END PROCESSOR

- One of the most common approaches for processing the speech signal is the bank-of-filters model
- This method takes a speech signal s(n) as input and passes it through a set of bank of Q bandpass filters in order to obtain the spectral representation of each frequency band of interest
- Eg
    - 100-3000 Hz for telephone quality signal
    - 100-8000 Hz for broadband signal
- The individual filters generally overlap in frequency as shown in bottom part of the figure
- The output of the i^{th} bandpass filter $X_n(e^{jw_i})$
- where W_i is the normalized frequency
- Each bandpass filter processes the speech signal independently to produce the spectral representation X_n.

The complete block diagram of filter bank front end analyzer is given below:

IDOL

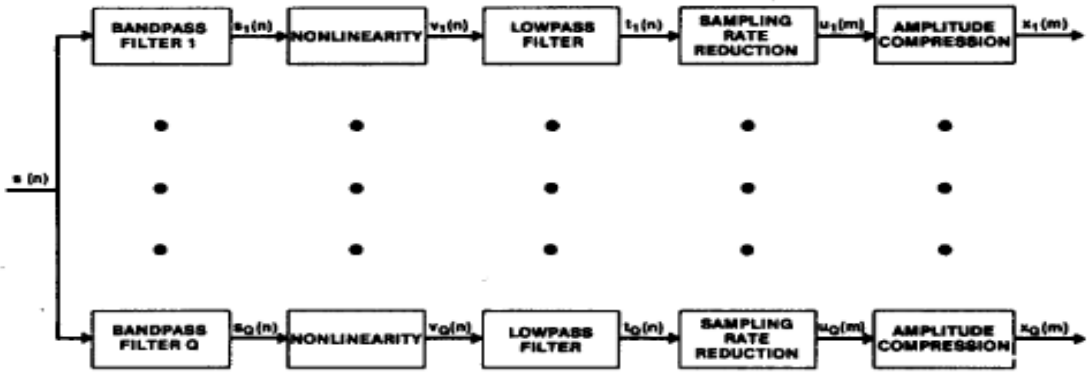


Figure 4.2 complete bank of filter analysis model [1]

- The sampled speech signal, s(n), is passed through a bank of Q Band pass filters, giving the signals

$$s_i(n) = s(n) * h_i(n), \quad 1 \le i \le Q \qquad (4.1)$$
$$= \sum_{m=0}^{M_i - 1} h_i(m) s(n-m)$$

- Here we have assumed that the impulse response of the i^{th} band pass filter is $h_i(m)$ with a duration of M_i samples.

➢ The bank-of-filters approach obtains the energy value of the speech signal considering the following steps:

- Signal enhancement and noise elimination.- Enhance and eliminate the noise from speech signal to make the speech signal more evident to the bank of filters.

IDOL

- Set of bandpass filters - Separate the signal in frequency bands. (uniform/non uniform filters)
- Non linearity - Each bandpass signal is passed through a non linear function such as a wave rectifier, full wave or half wave, for shifting the bandpass spectrum to the low-frequency band.
- Low pass filter - Low pass filter eliminates the high-frequency generated by the non linear function
- Sampling rate reduction and amplitude compression.- These are the final two blocks of model where the resulting signals are represented in a more economic way by re-sampling with a reduced rate and compressing the signal dynamic range using amplitude compression scheme(e.g. Logarithmic encoding, μ law encoding).

- Assume that the output of the i^{th} bandpass filter is a pure sinusoid at frequency ω_I

$$s_i(n) = \alpha_i \sin(\omega_i n)$$

- If full wave rectifier is used as the nonlinearity

$$f(s_i(n)) = s_i(n) \quad \text{for } s_i(n) \geq 0$$
$$= -s_i(n) \text{ for } s_i(n) < 0$$

The nonlinearity output:

$$v_i = f(s_i(n)) = s_i(n).w(n)$$

where

$$w(n) = \begin{Bmatrix} +1 \text{ if } s_i(n) \geq 0 \\ -1 \text{ if } s_i(n) < 0 \end{Bmatrix}$$

Since the nonlinearity output can be viewed as a modulation in time then in frequency domain we get result

$$V_i(e^{jw}) = S_i(e^{jw}) \otimes W(e^{jw})$$

$V_i(e^{jw})$, $S_i(e^{jw})$ and $W(e^{jw})$ are fourier transforms of vi(n),si(n) and w(n) resp and $\circledast$ is circular convolution

IDOL

4.3.1 Types of Filter Bank Used For Speech Recognition

1. Uniform filter bank
2. Non uniform filter bank

4.3.1.1 Uniform filter bank

- The most common filter bank used for speech recognition is the uniform filter bank
- For uniform filters the bandwidth that each individual filters spans is the same. So the name uniform.
- The center frequency, fi, of the ith bandpass filter is defined as

$$f_i = \frac{F_s}{N} i \qquad 1 \le i \le Q$$

Where Fs is the sampling rate of the speech signal
N is the number of uniformly spaced filters required to span the frequency range
Q is the number of filters used

- The actual number of filters used in the filter bank with equality

$$Q \le N - 2$$

bi is the bandwidth of the i^{th} filter

$$bi \ge \frac{Fs}{N}$$

with equality means There should not be any frequency overlap between adjacent filter channels

with inequality means there should be overlap between adjacent filter channels

IDOL

4.3.1.2 Nonuniform filter bank

- Alternative to uniform filter bank is nonuniform filter bank
- The criterion is to space the filters uniformly along a logarithmic frequency scale.
- In nonuniform filter bank bandwidth of the all filters is not same. It keeps on increasing logarithmically.
- For a set of Q bandpass filters with center frequencies fi and bandwidths bi, 1≤i≤Q, we set

$$b_1 = C$$

$$b_i = \alpha b_{i-1,}\ 2 \le i \le Q$$

$$f_i = f_1 + \sum_{j=1}^{i-1} b_j + \frac{(b_i - b_1)}{2}$$

  Where C and *f*i are the arbitrary bandwidth and the center frequency of the first filter and α is the logarithmic growth factor

- The most commonly used values of α=2
- This gives an octave band spacing adjacent filters
- And α=4/3 gives 1/3 octave filter spacing

4.3.2 Implementations of Filter Banks

- Depending on the method of designing the filter bank can be implemented in various ways.
- Design methods for digital filters fall into two classes:
  - Infinite impulse response (IIR) (recursive filters)
  - Finite impulse response

➢ <u>The IIR filter:</u> (Infinite impulse response) or recursive filter

- The current output sample is depends on the current input, past input samples and output samples
- The impulse response extends over an infinite duration
- Advantage:
  - Simple to design

IDOL

- Efficient

- Disadvantage:
  - Phase response is non linear
  - Noise affects more
  - Not stable.

➢ <u>The FIR filter:</u> (finite impulse response) or non recursive filter

- The current output is depend on the current input sample and previous input samples
- The impulse response is restricted to finite number of samples

$$x_i(n) = s(n) * h_i(n) \quad 0 \le n \le L-1 \text{ where L are samples}$$
$$= \sum_{m=0}^{L-1} h_i(m)s(n-m) \text{ for i} = 1,2,...Q$$

$h_i(n)$ is the impulse response of the i^{th} channel

$x_i(n)$ is the output of the i^{th} channel

$s(n)$ input signal

- Advantages:
  - Stable
  - Phase response is linear
- Disadvantage:
  - Costly to implement
  - Require high computational facilities

- Less expensive implementation can be derived by representing each bandpass filter by a fixed low pass window ω(n) modulated by the complex exponential

$$h_i(n) = \omega(n)e^{jw_i n}$$
$$x_i(n) = \sum_m \omega(m)e^{jw_i n}s(n-m)$$
$$= \sum_m s(m)\omega(n-m)e^{jw_i(n-m)}$$
$$= e^{jw_i n}\sum s(m)\omega(n-m)e^{-jw_i m}$$
$$= e^{jw_i n}Sn(e^{jw_i})$$

where $S_n(e^{jw_i})$ is the Fourier tr ansform of $s(n)$ at $w_i = 2\Pi fi$

4.3.2.1 Frequency Domain Interpretation For Short Term Fourier Transform

- The short time fourier transform s(m) is defined as

$$S_n(e^{jw_i}) = \sum_m s(m)\omega(n-m)e^{-jw_i m}$$

For fixed $n=n_0$

$$S_{n_0}(e^{jw_i}) = FT[s(m)\omega(n_0-m)]|_{\omega=\omega i}$$

Where FT[.] denotes Fourier Transform

- $S_{n0}(e^{j\omega i})$ is the conventional Fourier transform of the windowed signal, $s(m)w(n_0-m)$, evaluated at the frequency $\omega = \omega_i$

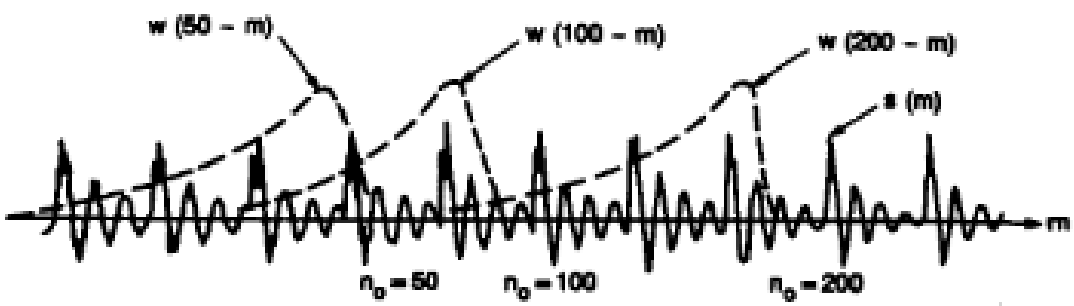


Figure 4.3 The signals s(m) and w(n-m) used in evaluation of the short-time fourier transform [1]

Above figure shows which part of s(m) are used in the computation of the short time Fourier transform

- Since w(m) is an FIR filter with size L then from the definition of $Sn(ej\omega i)$ we can state that
  - If L is large, relative to the signal periodicity then $Sn(ej\omega i)$ gives good frequency resolution
  - If L is small, relative to the signal periodicity then $Sn(ej\omega i)$ gives poor frequency resolution

4.3.2.2 Linear Filtering Interpretation of the short-time Fourier Transform

- The linear filtering interpretation of the short time Fourier Transform is derived by considering equation

$$S_n(e^{jw_i}) = \sum_m s(m)\omega(n-m)e^{-jw_i m}$$

- for fixed values of w_i.

$$S_n(e^{jw_i}) = s(n)e^{-jw_i n}\Theta\omega(n)$$

- i.e $S_n(e^{jwi})$ is a convolution of the low pass window, w(n), with the speech signal, s(n), modulated to the center frequency wi

4.3.2.3 FFT Implementation of Uniform Filter Bank Based on the Short-Time Fourier Transform

- The FFT implementation is more efficient than the direct form structure.
- If we assume that we are interested in a uniform frequency spacing i.e., if

$$fi = i(Fs/N), \qquad i = 0,1,2...., N-1$$

we know that

$$x_i(n) = e^{jw_i n}\sum_m s(m)\omega(n-m)e^{-jw_i m} \qquad \text{where } w_i = 2\pi fi$$

$$= e^{j(\frac{2\pi}{N})in}\sum_m s(m)\omega(n-m)e^{-j(\frac{2\pi}{N})im}$$

Now assume breaking up summation over m into a double summation of r and k

$$m = Nr + k, \qquad 0 \le k \le N-1, \qquad -\infty < r < \infty.$$

Let,

$$s_n(m) = s(m)w(n-m)$$

IDOL

$$x_i(n) = e^{j(\frac{2\pi}{N})in} \sum_r \left[\sum_{k=0}^{N-1} s_n(Nr+k) \right] e^{-j(\frac{2\pi}{N})i(Nr+k)}$$

since $e^{-j2\pi ir} = 1$ for all i, r then

$$x_i(n) = e^{j(\frac{2\pi}{N})in} \sum_{k=0}^{N-1} \left[\sum_r s_n(Nr+k) \right] e^{-j(\frac{2\pi}{N})ik}$$

we define

$$u_n(k) = \sum_r s_n(Nr+k), \quad 0 \le k \le N-1$$

$$x_i(n) = e^{j(\frac{2\pi}{N})in} \left[\sum_{k-0}^{N-1} u_n(k) e^{-j(\frac{2\pi}{N})ik} \right]$$

which is the desired result

4.3.2.4 Nonuniform FIR Filter Bank Implementations

The most general form of a nonuniform FIR filter bank is shown below

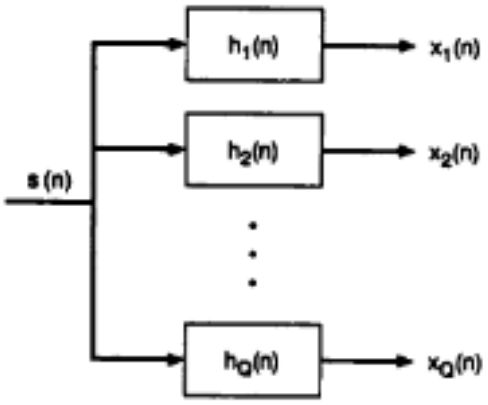


Figure 4.4 Direct from implementation of an arbitrary nonuniform filter bank [1]

- Where The k^{th} bandpass filter impulse response, hk(n), represents a filter with a center frequency ωk, and bandwidth $\Delta\omega k$.
- The set of Q bandpass filters covers the frequency range of interest for the intended speech recognition application

IDOL

- Each band pass filter is implemented via a direct convolution, i.e. No FFT structure can be used
- Each band pass filter is designed via the windowing design method using same lowpass window
- The composite frequency response of the Q-channel filter bank is independent of the number and distribution of the individual filters
- A filter bank with the three filters shown below in (a) has the exact same composite frequency response as the filter bank with the seven filters shown in figure below in (b)

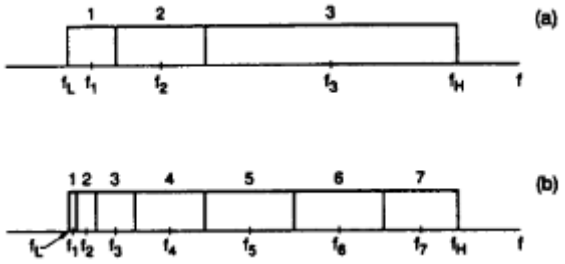


Figure 4.5 two arbitrary uniform filter bank ideal filter specifications consisting of wither 3 band part (a) or 7 bands part (b) [1]

- The impulse response of the k[th] bandpass filter

$$h_k(n) = w(n)\bar{h}_k(n)$$

Where w(n) is the FIR window, $\bar{h}_k(n)$ is the impulse response of ideal bandpass filter

- The frequency response of the k[th] bandpass filter

$$H_k(e^{jw}) = W(e^{jw}) \otimes \tilde{H}_k(e^{jw})$$

- Thus the frequency response of the composite filter bank, H(e^{jw}) can be written as follows,

$$H(e^{jw}) = \sum_{k=1}^{Q} H(e^{jw}) = \sum_{k=1}^{Q} W(e^{jw}) \Theta \tilde{H}_k(e^{jw})$$

By interchanging the summation and the convolution we get,

$$H(e^{jw}) = W(e^{jw}) \Theta \sum_{k=1}^{Q} \tilde{H}_k(e^{jw})$$

By realising the summation of the above equation is the summation of ideal frequency responses, we can write the summation as

$$\hat{H}(e^{jw}) = \sum_{k=1}^{Q} \tilde{H}_k(e^{jw}) = \begin{cases} 1, w_{\min} \le w \le w_{\max} \\ 0, \text{ Otherwise} \end{cases}$$

- Where wmin is the lowest frequency in the filter bank and wmax is the highest frequency
- Equation 1 can be written as

$$H(e^{jw}) = W(e^{jw}) \Theta \hat{H}(e^{jw})$$

- Which is independent of the number of ideal filters, Q, and their distribution in the frequency

4.3.2.5 FFT-Based Nonuniform Filter Banks

- By combining two or more uniform channels the nonuniformity can be created
- Consider taking an N-point DFT of the sequence x(n)

$$X_k = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}nk}, \quad 0 \le k \le N-1$$

Add DFT outputs X_k and X_{k+1}

$$X_k + X_{k+1} = \sum_{n=0}^{N-1} x(n) \left(e^{-j\frac{2\pi}{N}nk} + e^{-j\frac{2\pi}{N}n(k+1)} \right)$$

$$X'_k = X_k + X_{k+1} = \sum_{n=0}^{N-1} \left[x(n) 2 e^{-j\frac{2\pi}{N}} \cos\left(\frac{\pi n}{N}\right) \right] e^{-j\frac{2\pi}{N}nk}$$

- The equivalent kth channel value, X_k' can be obtained by weighing the sequence, x(n) by the complex sequence $2\, e^{-j\pi n/N} \cos(\pi n/N)$.
- If more than two channels are combined, then a different equivalent weighing sequence results.

IDOL

4.3.2.6 Tree Structure Realizations of Nonuniform Filter Banks

- In this method the speech signal is filtered in the stages, and the sampling rate is successively reduced at each stage

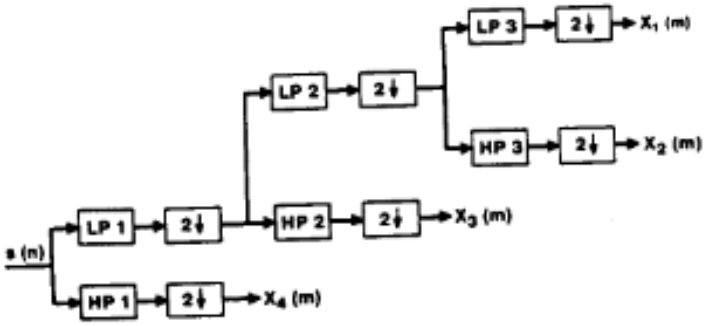


Figure 4.6 tree structure implementation of 4 band [1]

- The original speech signal, s(n), is filtered initially into two bands, a low band and a high band using QMF i.e. quadrature mirror filters, whose frequency responses are complementry
- The high band is reduce by sampling rate of factor of 2 and represents the highest octave band ($\pi/2 \leq \omega \leq \pi$) of the filter bank.
- The low band is similarly reduce by sampling rate of factor of 2 and fed into second filtering stage in which the signal is again split into two equal bands.
- Again the high band of the stage 2 is down sampled by factor 2 and is used as a next highest filter bank output.
- The low band is also down sampled by a factor 2 and fed into a third stage of QMF filters
- These third stage output after down sampling by factor 2, are used as the two lowest filter bands

4.3.3 Generalizations of Filter Bank Analyzer

- The generalized structured of filter bank analyzer is shown below

IDOL

Figure 4.7 generalisation of filter-bank analysis model

- Signal preprocessor: “conditions” the speech signal s(n) to new form which is more suitable for the analysis
- Postprocessor: operate on the ouptut x(m) to give the processed output that are more suitable for recognition

- Preprocessor Operations
  - Signal preemphasis: higher frequencies are increased in amplitude
  - Noise elimination
  - Signal enhancement (to make the formant peaks more prominent)
- The purpose of pre processor is to make the speech signal as clean as possible; hence it eliminates the noise, long spectral trends are removed and signal is spectrally flattened to give the best immunity

- Postprocessor Operations
  - Temporal smoothing of sequential filter bank output vectors.
  - Frequency smoothing of individual filter bank output vectors.
  - Normalization of each filter bank output vector
  - Thresholding and/or quantization of the filter-bank outputs vectors
  - Principal components analysis of the filter bank output vector.
- The purpose of postprocessor is to clean up the output so as to best represent the spectral information in the speech signal.

References and further Reading

[1] L.R. Rabiner , B. H. Juang , “Fundamentals of Speech Recognition”, Prentice-Hall, Englewood Cliffs, N.J.

IDOL

IDOL

CHAPTER 5

SIGNAL PROCESSING AND ANALYSIS METHODS FOR SPEECH RECOGNITION

5.1 LINEAR PREDICTIVE CODING MODEL FOR SPEECH RECOGNITION

- Linear predictive coding(LPC) is defined as a digital method for encoding an analog signal in which a particular value is predicted by a linear function of the past values of the signal
- Human speech is produced in the vocal tract which can be approximated as a variable diameter tube.
- The linear predictive coding (LPC) model is based on a mathematical approximation of the vocal tract represented by this tube of a varying diameter.
- At a particular time, t, the speech sample s(t) is represented as a linear sum of the p previous samples. The most important aspect of LPC is the linear predictive filter which allows the value of the next sample to be determined by a linear combination of previous samples

■ The reasons why LPC has been widely used:

- For the quasi steady state voiced regions of speech LPC provides a good model of the speech recognition
- During unvoiced and transient regions of speech, the LPC model is less effective but it still provides an acceptable useful model for speech recognition purpose
- The method of LPC is mathematically precise and is simple and straightforward to implement in either in software or hardware
- The computation involved in LPC processing is considerably less than that required for implementation of the bank-of-filters model

IDOL

- The performance of speech recognizers, based on LPC front ends is better than that of recognizer based on filter-bank front ends.

5.1.1 The LPC Model

- The speech sample at time n can be approximated as a linear combination of the past p speech samples,

$$s(n) \approx a_1 s(n-1) + a_2 s(n-2) + ... + a_p s(n-p)$$

- Where the coefficient $a_1, a_2, \ldots, a_p$ are assumed constant over the speech analysis frame we convert above equation to an equality by including an excitation term G u(n)

$$s(n) = \sum_{i=1}^{P} a_i s(n-i) + Gu(n)$$

- Where u(n) is excitation and G is the gain of excitation, in Z-domain we get the relation

$$S(z) = \sum_{i=1}^{P} a_i z^{-i} S(z) + GU(z)$$

Leading to transfer function

$$H(z) = \frac{S(z)}{GU(z)} = \frac{1}{1 - \sum_{i=1}^{P} a_i z^{-i}} = \frac{1}{A(z)}$$

- The interpretation of above equation is given in following figure

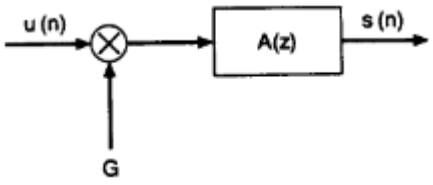


Figure 5.1 linear predication model of speech [1]

IDOL

- In above diagram the excitation source u(n) being scaled by the gain G and acting as input to all-pole system H(z) = 1/A(z) to produce speech signal s(n).
- Excitation source can be either a quasi periodic train pulses (voiced) or a random noise source (unvoiced sounds).

➢ The appropriate model for speech corresponding to LPC analysis is shown below.

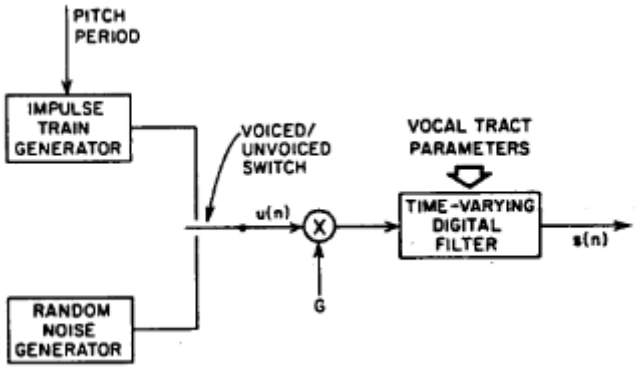


Figure 5.2 speech synthesis model based on LPC model

- Here in above diagram the voiced/unvoiced switch Chooses either a quasi periodic train of pulses as the excitation for voiced sounds or a random noise sequence for unvoiced sounds
- The appropriate Gain G of the source is estimated from speech signal and scaled source is used as input to a digital filter which is controlled by vocal tract parameter
- Thus The parameters of this model
  - Voiced/unvoiced classification
  - Pitch period for voiced sounds
  - The gain parameter
  - The coefficients of the digital filter {ak},
- These parameters all very slowly with time

5.1.2 LPC Analysis Equations

- Based on the diagram figure 3.8 the exact relation between s(n) and u(n) is

$$s(n) = \sum_{k=1}^{P} a_k s(n-k) + Gu(n)$$

- Consider linear combination of past speech samples, defined as

$$\tilde{s}(n) = \sum_{k=1}^{P} a_k s(n-k)$$

- The predication error e(n), defined as

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^{P} a_k s(n-k)$$

With error transfer function

$$A(z) = \frac{E(z)}{S(z)} = 1 - \sum_{k=1}^{p} a_k z^{-k}$$

- The basic problem is to determine the set of predictor coefficients, $\{a_k\}$, directly from the speech signal so that the spectral properties of the digital filter match those of the speech waveform within the analysis window.
- Since the spectral characteristics of speech vary over time, the predictor coefficients at a given time, n, must be estimated from a short segment of the speech signal occurring around time n
- The need is to find a set of predictor coefficients that minimize the mean squared prediction error over a short segment of speech waveform.
- To solved the predictor coefficients, we define short term speech and error segments at time n

$$s_n(m) = s(n+m)$$
$$e_n(m) = e(n+m)$$

and we seek to minimize the mean squared error signal at time n

$$E_n = \sum_m e_n^2(m)$$

IDOL

By the definition of $e_n(m)$ in terms of $s_n(m)$

$$E_n = \sum_m \left[s_n(m) - \sum_{k=1}^{P} a_k s_n(m-k) \right]^2$$

To calculate prediction coefficients differentiate E_n with respect to each a_k and set the result to zero

$$\frac{\partial E_n}{\partial a_k} = 0, \quad \mathrm{k} = 1,2,,.....\mathrm{p}$$

giving

$$\sum_m s_n(m-i)s_n(m) = \sum_{k=1}^{p} \hat{a}_k \sum_m s_n(m-i)s_n(m-k)$$

by determining the above terms are the terms of short-term covariance of $s_n(m)$ i.e.,

$$\phi_n(i,k) = \sum_m s_n(m-i)s_n(m-k)$$

So the compact notation of this is

$$\phi_n(i,0) = \sum_{k=1}^{p} \hat{a}_k \phi_n(i,k)$$

Which describes a set of p equation in p unknowns, The minimum mean-squared error, $\hat{\mathrm{E}}_n$ can be expressed as

$$\hat{E}_n = \sum_m s^2{}_n(m) - \sum_{k=1}^{p} \hat{a}_k \sum_m s_n(m-i)s_n(m-k)$$

$$= \Phi_{\mathrm{n}}(0,0) - \sum_{k=1}^{p} \hat{a}_k \Phi_{\mathrm{n}}(0,k)$$

The mean square error consist of $\Phi_{\mathrm{n}}(0,0)$ is a fixed term and $\sum_{k=1}^{p} \hat{a}_k \Phi_{\mathrm{n}}(0,k)$ term depend on predictor coefficients

IDOL

- Two methods of defining the range of speech (m)
  - The autocorrelation method
  - The covariance method

5.1.3 The autocorrelation method

$$s_n(m) = \begin{cases} s(n+m).w(m), & 0 \le m \le \mathrm{N}-1 \\ 0 & \text{otherwise} \end{cases} \qquad \text{--- eq. (1)}$$

- The speech signal s(m+n) is multiplied by a finite window, w(m) Which is zero outside the range 0≤m ≤N-1
- The purpose of window in the above equation is to taper the signal near m=0 and near m=N-1 so as to minimize the errors at section boundaries
- Based on eq. (1) equation .
  - For m<0 , the prediction error i.e $e_n(m)=0$ since $s_n(m)=0$ for all m<0 and therefore there is no prediction error
  - For m>N-1+p there is no prediction error because $s_n(m)=0$ for all m>N-1

Based on the eq. (1) equation the mean squared error becomes

$$\mathrm{E_n} = \sum_{m=0}^{N-1+p} e_n^2(m)$$

And $\phi_n(i,k)$ can be expressed as

$$\phi_n(i,k) = \sum_{m=0}^{N-1+p} s_n(m-i)s_n(m-k), 1 \le i \le \mathrm{p},\ 0 \le \mathrm{k} \le \mathrm{p}$$

Which is equivalent to the form

$$\phi_n(i,k) = \sum_{m=0}^{N-1-(i-k)} s_n(m-i)s_n(m+i-k), 1 \le i \le \mathrm{p},\ 0 \le \mathrm{k} \le \mathrm{p}$$

The covariance function, $\phi_n(i,k)$ reduced to the simple autocorrelation function

$$\Phi_n(i,k) = r_n(i-k) = \sum_{m=0}^{N-1-(i-k)} s_n(m)s_n(m+i-k)$$

Since the auto correlation function is symmetric i.e. $r_n(-k) = r_n(k)$, the LPC equations can be expressed as

$$\boxed{\sum_{k=1}^{p} r_n(|i-k|)\hat{a}_k = r_n(i), \quad 1 \le i \le p}$$

And can be expressed in matrix form as

$$\begin{bmatrix} r_n(0) & r_n(1) & r_n(2) & \cdots & r_n(p-1) \\ r_n(1) & r_n(0) & r_n(1) & \cdots & r_n(p-2) \\ r_n(2) & r_n(1) & r_n(0) & \cdots & r_n(p-3) \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ r_n(p-1) & r_n(p-2) & r_n(p-3) & \cdots & r_n(0) \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \vdots \\ \hat{a}_p \end{bmatrix} = \begin{bmatrix} r_n(1) \\ r_n(2) \\ r_n(3) \\ \vdots \\ r_n(p) \end{bmatrix}$$

- The p×p matrix of autocorrelation values is a Toeplitz matrix (symmetric with all diagonal elements equal)
- Can be solved by several procedures like Durbin algorithm

IDOL

5.1.4 Covariance Method

- Instead of using weighting function or window for defining $s_n(m)$ we can fix the interval over which the mean - squared error is computed to the range $0 \le m \le N-1$

$$E_n = \sum_{m=0}^{N-1} e_n^2(m)$$

$$\Phi_n(i,k) = \sum_{m=0}^{N-1} s_n(m-i)s_n(m-k), \quad \begin{matrix} 1 \le i \le p \\ 0 \le k \le p \end{matrix}$$

- By extending the speech interval to define the covariance values , the matrix form of the LPC analysis equations becomes:

$$\begin{bmatrix} \Phi_n(1,1) & \Phi_n(1,2) & \Phi_n(1,3) & \cdots & \Phi_n(1,p) \\ \Phi_n(2,1) & \Phi_n(2,2) & \Phi_n(2,3) & \cdots & \Phi_n(2,p) \\ \Phi_n(3,1) & \Phi_n(3,2) & \Phi_n(3,3) & \cdots & \Phi_n(3,p) \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \Phi_n(p,1) & \Phi_n(p,1) & \Phi_n(p,1) & \cdots & \Phi_n(p,p) \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a} \\ \vdots \\ \hat{a}_1 \end{bmatrix} = \begin{bmatrix} \Phi_n(1,0) \\ \Phi_n(2,0) \\ \Phi_n(3,0) \\ \vdots \\ \Phi_n(p,0) \end{bmatrix}$$

- The resulting covariance matrix is symmetric (since $\phi_n(i,k) = \phi_n(k,i)$) but not Toeplitz, can be solved by Cholesky decomposition method.

IDOL

5.1.5 LPC Processor for Speech Recognition

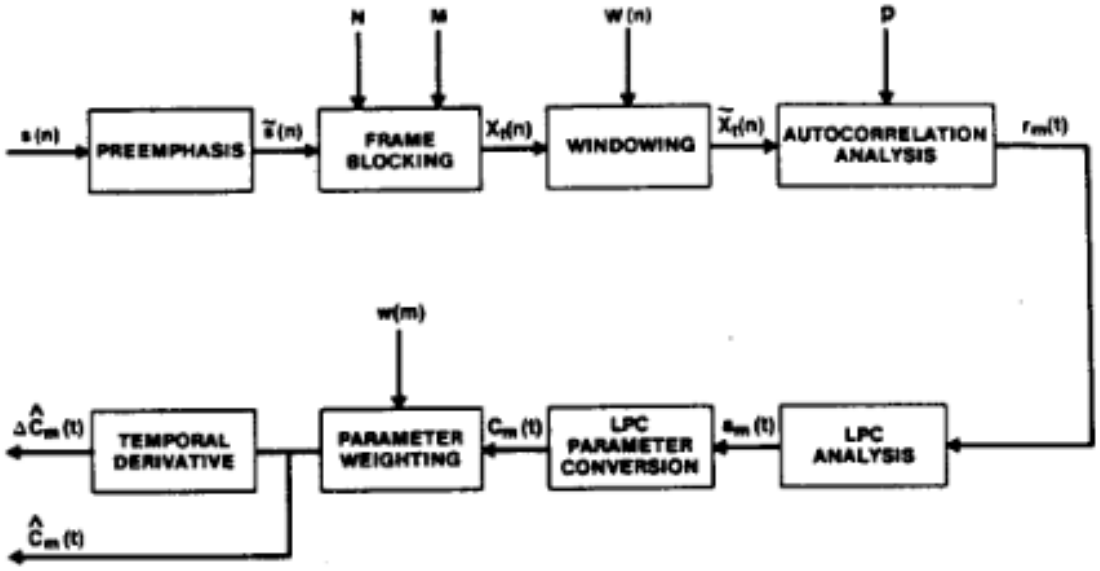


Figure 5.3 block diagram of LPC processor for speech recognition

1. Preemphasis:
   - The digitized speech signal, s(n), is put through a low order digital system, to spectrally flatten the signal and to make it less susceptible to finite precision effects later in the signal processing. The output of the preemphasizer network, , is related to the input to the network, s(n), by difference equation:

$$\tilde{s}(n) = s(n) - \tilde{a}s(n-1)$$

2. Frame Blocking:
   - The output of preemphasis step $\tilde{s}(n)$, is blocked into frames of N samples, with adjacent frames being separated by M samples. If $x_l(n)$ is the l^{th} frame of speech, and there are L frames within entire speech signal, then

$$X_l(n) = \tilde{s}(ml+n)$$

   where n = 0,1,…,N – 1 and l = 0,1,…,L – 1

3. Windowing:
    - After frame blocking, the next step is to window each individual frame so as to minimize the signal discontinuities at the beginning and end of each frame. If we define the window as w(n), 0 ≤ n ≤ N – 1, then the result of windowing is the signal:

$$\tilde{\chi}_l(n) = \chi_l(n)w(n),$$

where 0 ≤ n ≤ N – 1

Typical window is the Hamming window, which has the form

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \le n \le N-1$$

4. Autocorrelation Analysis:
    - The next step is to auto correlate each frame of windowed signal in order to

$$r_l(m) = \sum_{n=0}^{N-1-m} \tilde{x}_l(n)\tilde{x}_l(n+m), \quad \mathrm{m} = 0,1,2,....,\mathrm{p}$$

where the highest autocorrelation value, p, is the order of the LPC analysis.

5. LPC Analysis:
    - The next processing step is the LPC analysis, which converts each frame of p + 1 autocorrelations into LPC parameter set by using Durbin's method. This can formally be given as the following algorithm:

$$E^{(0)} = r_l(0) \qquad \text{-- (1)}$$

$$k_i = \left\{ r_l(i) - \sum_{j=1}^{L-1} \alpha_j^{(i-1)} r_l(|i-j|) \right\} \Big/ E^{(i-1)}, \quad 1 \le i \le \mathrm{p} \qquad \text{-- (2)}$$

$$\alpha_{\mathrm{i}}^{(\mathrm{i})} = k_i \qquad \text{-- (3)}$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)} \qquad \text{-- (4)}$$

IDOL

$$E^{(i)} = (1-k_i^2)E^{(i-1)} \qquad \text{-- (5)}$$

After solving the above equation (1) and() for i=1,2,…,p the final LPC coefficient a_m is given as

$$a_m = \alpha_m^{(p)}$$

6. LPC Parameter Conversion to Cepstral Coefficients:
   - A very important parameter set which can be derived directly from the LPC coefficient set, is the LPC cepstral coefficients, c(m)

     The recursion used is

$$c_0 = \ln \sigma^2 \text{.....1}$$

$$c_m = a_m + \sum_{k=1}^{m-1}\left(\frac{k}{m}\right)c_k a_{m-k}, \quad 1 \le m \le p \text{.....2}$$

$$c_m = \sum_{k=1}^{m-1}\left(\frac{k}{m}\right)c_k a_{m-k}, \quad m > p \text{.....3}$$

     where σ^2 is the gain term in the LPC model.

   - The cepstral coefficients are the coefficients of the FT representation of the log magnitude spectrum.
   - The cepstral coefficients are more robust reliable feature set for speech recognition than the LPC coefficients, the PARCOR coefficients, or the log area ratio coefficients.

7. Parameter Weighting**:**
   - because of the sensitivity of the low-order cepstral coefficients to overall spectral slope and sensitivity of the high-order cepstral coefficients to noise, it is necessary to weight the cepstral coefficients by a tapered window to minimize these sensitivities.
   - By differentiating the Fourier representation of log magnitude spectrum

$$\log\left|S(e^{j\omega})\right| = \sum_{m=-\infty}^{\infty} c_m e^{-j\omega m}$$

IDOL

8. Temporal Cepstral Derivative:
    - the Cepstral representation of the speech spectrum provides a good representation of the local spectral properties of the signal for the given frame
    - Better representation can be obtained by including the information about temporal Cepstral derivative.

$$\frac{\partial}{\partial t}\left[\log\left|S(e^{j\omega},t)\right|\right]=\sum_{m=-\infty}^{\infty}\frac{\partial c_m(t)}{\partial t}e^{-j\omega m}$$

Where $c_m(t)$ is the m^{th} cepstral coefficient at time t

To approximate $\frac{\partial c_m(t)}{\partial t}$ over a finite length window that is,

$$\frac{\partial c_m(t)}{\partial t}=\Delta c_m(t)\approx\mu\sum_{k=-K}^{K}kc_m(t+k)$$

Where μ is an appropriate normalization constant and (2K+1) is the number if frames over which the computation is performed. Typically K = 3 for each frame t, the result of LPC analysis is a vector of Q weighted cepstral coefficient and vector of Q cepstral derivatives

For each frame t, the result of the LPC analysis vector Q weighted cepstral coefficient and an appended vector of Q cepstral time derivative that is

$$o'_t=(\hat{c}_1(t),\hat{c}_2(t),.....,\hat{c}_Q(t),\Delta c_1(t),\Delta c_2(t),....,\Delta c_Q(t))$$

O't is a vector with 2Q component and denotes matrix transpose

5.1.6 Typical LPC Analysis Parameters

- The computation of the LPC analysis of figure 3.10 is specified number of paramters
    - N: number of samples in analysis frame.
    - M: number of samples shift between analysis frames.

IDOL

- p: LPC analysis order.
- Q: dimension of LPC derived cepstral vector.
- K: number of frames over which cepstral time derivatives are computed.

5.2 Vector Quantization

- The vector quantization is the technique of compressing the data.
- The main objective of data compression is to maintain the necessary fidelity of the data while reducing the bit rate for transmission or data storage
- The Vector Quantization is the extension of scalar quantization
- The output of both filter bank and LPC analysis is in the form of vectors
- The VQ encoder encodes a given set of k-dimensional data vectors with a much smaller subset. The subset C is called a codebook and its elements C_i are called codewords, codevectors, reproducing vectors, prototypes or design samples.
- Only the index i is transmitted to the decoder. The decoder has the same codebook as the encoder, and decoding is operated by table look-up procedure.
- Figure 3.11 shows the diagram of encoder and decoder
- The commonly used vector quantizers are based on nearest neighbour called Voronoi or nearest neighbour vector quantizer. Both the classical K-means algorithm and the LBG algorithm belong to the class of nearest neighbour quantizers.

IDOL

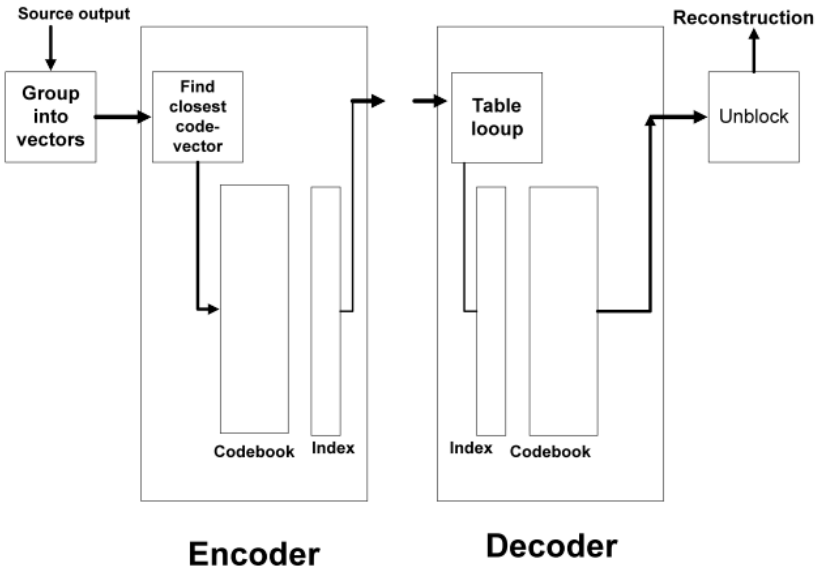


Figure 5.4 Encoder and decoder in vector quantizer

- Advantages of vector quantization
  - Reduced storage for spectral analysis information
  - Reduced computation for determining similarity of spectral analysis vectors
  - Major of the computation required in speech recognition is for determining the spectral similarity between a pair of vectors. Using the vector quantization this computation is reduced
  - Discrete representation of speech sounds

- Disadvantage
  - An inherent spectral distortion in representing the actual analysis vector: There is only a finite number of codebook vectors, the process of choosing the most similar representation of spectral vector inherently equivalent to quatizing the vector leads to certain level of quantization erroras the size of the codebook increases, the size of the quantization error decreases and vice versa
  - The storage required for the codebook is important: larger the codebook lesser the quantization error but more the storage required for the codebook

IDOL

To build a code book of vector quantization we need

- To form a training set, we need a large set of spectral analysis vector v1,v2….vn. Training set is used to create set of codebook vectors for representing the spectral variability observed in training set
- a measure of similarity or distance between a pair of spectral analysis vectors so as to be able to cluster the framing set vectors into unique codebook entries.

  We denote spectral distance between two vetors as

$$d(v_i, v_j) = d_{ij} \begin{cases} = 0 & if\ v_i = v_j \\ > 0 & otherwise \end{cases}$$

- a centroid computation procedure, from the L training set vectors into M clusters, we choose the M codebook vectors as the centroid of each of the M clusters.
- a classification procedure , The classification procedure is essentially a quantizer that accepts, as input, a speech spectral vector and provides, as output, the codebook index of the codebook vector that best matches the input
- Following diagram shows the basic VQ training and classification structure

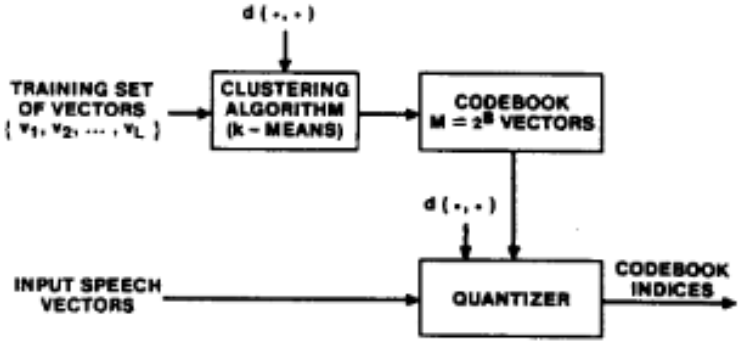


Figure 5.5 Block diagram of basic VQ training and classification structure

Clustering Algorithms:

- Assume that we have a set of L training vectors and we need a codebook of size M. A procedure that does the clustering is the K-Means clustering algorithm also known as Lloyd algorithm

  - Initialization: Arbitrarily choose M vectors as the initial set of code words in the codebook.
  - Nearest-Neighbor Search: For each training vector, find the codeword in the current codebook that is closest in terms of spectral distance and assign that vector to the corresponding cell.
  - Centorid Update: Update the code word in each cell using the centroid of the training vectors assigned to that cell.
  - Iteration: Repeat steps 2 and 3 until the average distance(Distortion) falls below a preset threshold

Vector classification procedure

- It is basically a full search through the codebook to find the "best" match
- Best match: The quantization error is minimum
- if we denote codebook of M vectors as y_m and we denote spectral vector to be classified as v then index m* of the best code book entry is

$$m^* = \arg \min_{1 \le m \le M} d(\mathbf{v}, \mathbf{y}_m).$$

- for codebook with large values of M the computation of above equation could be excessive

5.3 Auditory based spectral analysis model

- Here we are investing the physiologically based spectral analysis methods to understand how humans auditory system process the speech so as to be able to design the methods of analyzing and representing the speech.
- Below diagram is the physiological model of the human ear

IDOL

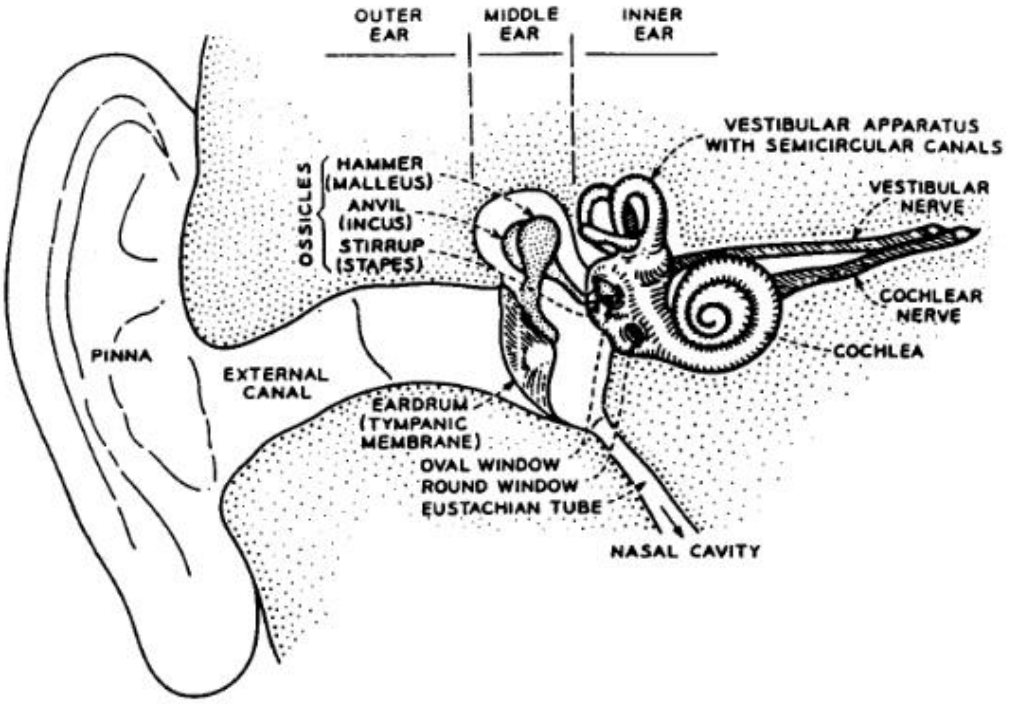


Figure 5.6 Figure physiological model of the human ear [1]

- There is three distinct region
  - Outer ear, Middle ear and Inner ear

- Outer ear :
  - Consist of
    - pinna i.e. the ear surface surroundings of canal
    - External canal
  - Sound waves travels from outer ear to the middle ear

- Middle ear:
  - Consist of
    - Eardrum – on which sound wave impinges and causes to move
    - mechanical transducer - consisting of the hammer, anvil and stirrup; it converts acoustical sound wave to mechanical vibrations along the inner ear

- Inner ear :
  - consist of
    - cochea : is a fluid-filled chamber partitioned by the basilar membrane
  - the auditory nerve is connected to the basilar membrane via inner hair cells
  - mechanical vibrations at the entrance to the cochlea create standing waves (of fluid inside the cochlea) causing basilar membrane to vibrate at frequencies commensurate with the input acoustic wave frequencies (formants) and at a place along the basilar membrane that is associated with these frequencies

➢ The basilar membrane function

- Basilar membrane characterized by a set of frequency responses at different points along the membrane
- The cochlea can be modelled as mechanical realization of a bank of filters
- Distributed along the Basilar Membrane is a set of sensors called Inner Hair Cells (IHC) which act as mechanical motion-to-neural activity converters
- Mechanical motion along the BM is sensed by local IHC causing firing activity at nerve fibers that innervate bottom of each IHC
- Each IHC connected to about 10 nerve fibers, each of different diameter
  - thin fibers fire at high motion levels,
  - thick fibers fire at lower motion levels
- 30,000 nerve fibers link IHC to auditory nerve
- Electrical pulses run along auditory nerve, ultimately reach higher levels of auditory processing in brain, perceived as sound.

When we build the auditory model for signal processing we need to model the middle ear, cochlea and hair cell system. There is one such model called Ensemble Interval Histogram (EIH) model.

IDOL

5.3.1 Ensemble Interval Histogram (EIH) model

- EIH is a model of cochlear and hair cell transduction consist of filter bank that models frequency selectivity at points along the basilar membrane, and nonlinear processor for converting filter bank output to neural firing patterns along the auditory nerve

- In EIH model
    - The basilar membrane's mechanical motion is sampled using 165 IHC channels equally spaced on log frequency cell between 150 and 7000 Hz.
    - cochlear filter designs match neural tuning curves for cats whose phase characteristic is minimum phase

- the next step of processing EIH model is shown below

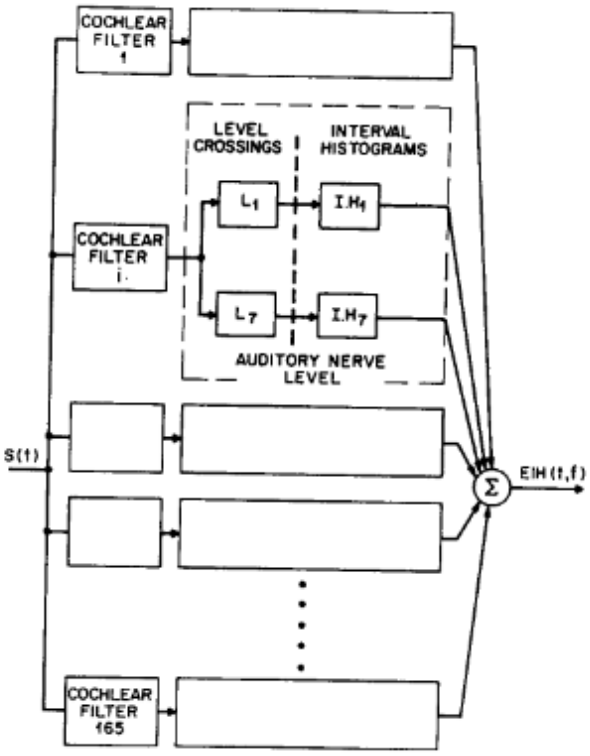


Figure 5.7 Block diagram of the EIH model

- In next step it is array of level crossing detectors that model motion-to-neural activity transduction of the hair cell mechanism.
- Detection levels of each detector are pseudo-randomly distributed to match variability of fiber diameters
- The output of the level crossing detectors represents the discharge activity of the auditory nerve fibers.
- The following figure shows the simulated auditory nerve activity for first 60 msec of vowel /o/ in both time and frequency of IHC channels

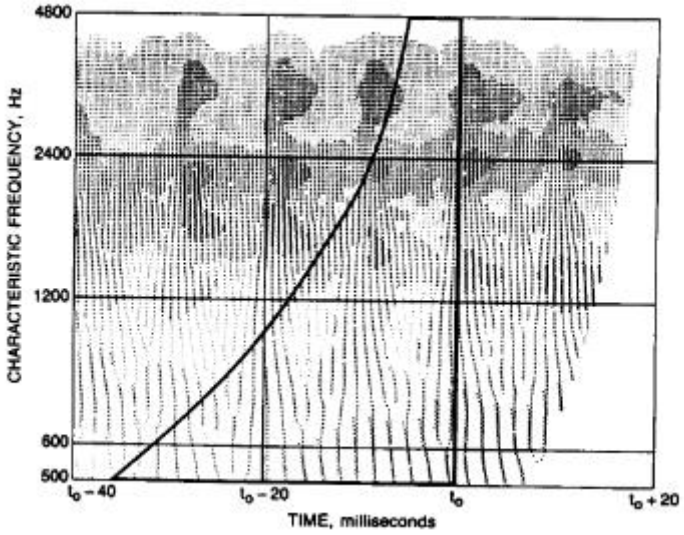


Figure 5.8 magnitude of EIH for vowel /o/ showing the time-frequency resolution [1]

- Level crossing occurrence marked by single dot and each level crossing detector is plotted as a separate trace
- If the magnitude of the filter output is low, one level will be crossed as seen for the very top channel in above figure
- If the magnitude of the filter output is low high many levels crossed as darker region in figure
- the level-crossing patterns represents the auditory nerve activity, which in turn is the input to more central stage of neural processing which gives the overall Ensemble Interval Histogram (EIH)
- EIH is a measure of spatial extent of coherent neural activity across auditory nerve

- It provides estimate of short term probability density function of reciprocal of intervals between successive firings in a characteristic frequency-time zone
- EIH preserves information about the signal's energy.

References and further Reading

[1] L.R. Rabiner , B. H. Juang , "Fundamentals of Speech Recognition", Prentice-Hall, Englewood Cliffs, N.J.

IDOL

CHAPTER 6

PATTERN COMPARISON TECHNIQUES PART -A

6.1 INTRODUCTION

- In pattern based approach to speech recognition the speech is directly represented by the time sequence of spectral vectors obtained from the front end spectral analysis
- It contains the test pattern T and the reference pattern R containing vectors.
- The goal of pattern comparison stage is to determine the dissimilarity of each vector in T to each vector of R to identify reference pattern that has minimum dissimilarity
- To determine the global similarity of T and R we will consider the following problems:
    - T and R generally are of unequal length w.r.t. time duration due to different speaking rates across different talkers
    - T and R need not line up in time in any simple or well prescribed manner this is because different sounds cannot be varied in duration to same degree. Vowels are easily lengthened or shortened but consonants cannot change in duration
    - We need a way to compare a spectral vectors

6.2 SPEECH DETECTION

- Also called as Endpoint detection
- Speech detection separates the speech which is to be processed in continuously recorded signal from the background speech
- To create a pattern or template for automatic speech recognition the speech detection is required to identify the speech of interest.

IDOL

- Speech must be detected to provide the best patterns for recognition. Best patterns means which provides highest recognition accuracy
- Accurate detection of speech in noise-free environment is simple
- But in many cases in noisy environment speech detection is difficult.

➢ Following are factors which makes speech detection difficult difficult

- First Factor
  - during the speech sometimes times the talker produces sound artifacts like heavy breathing, lip smacks, mouth clicks etc.
  - mouth click : mouth click is produced by opening the lips prior to speaking or after speaking. The noise of mouth click can be separate from speech signal
  - heavy breathing : heavy breathing noise is not separated from speech therefore makes accurate speech detection quite difficult

- Second factor
  - Environmental noise
  - quite room with no acoustic noise other than the the sound produce by speaker is the ideal room for talking but such environment is not always practical.
  - we need to consider the speech produced in noisy environment like car horns, road traffic noise, noisy background like machinery sound, fans , tv sound etc.
  - this noisy signals are somewhat like speech signal therefore speech detection become difficult

- Third factor
  - Transmission system
  - The speech signal are sent over the transmission system. The various factors like cross-talk, intermodulation distortion affect the speech

IDOL

signal therefore speech detection becomes difficult

- Following are the methods proposed for speech detection

1. explicit approach
2. implicit approach
3. hybrid approach

1. explicit approach

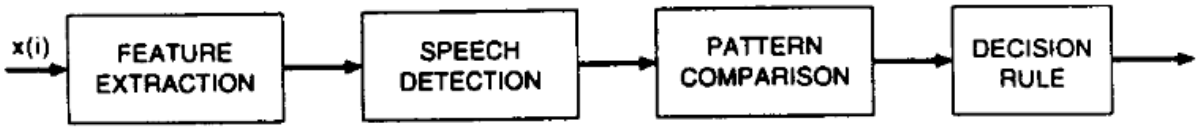


Figure 6.1 block diagram of explicit approach to speech endpoint detection [1]

In above diagram

- firstly speech signals are processed and feature measurement are made
- the speech-detection method is then applied to locate and define speech events
- the detected speech is sent to pattern-comparison algorithm, and finally the decision mechanism chooses the recognized word.

- Advantage
  - this approach produces good detection accuracy for the speech signal with a stationary and low level of background noise

- Disadvantage
  - approach fails when environment is noisy or interference is non-stationary

2. Implicit approach

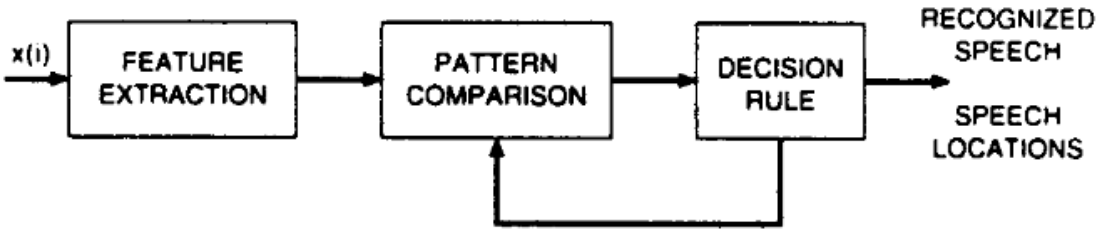


Figure 6.2 Block diagram of implicit approach [1]

- This approach consider speech detection problem simultaneously with pattern-matching and recognition-decision process.
- It recognizes that the speech events are almost always accompanied by a certain acoustic background
- In pattern-matching module all the possible endpoint sets are considered for unmarked signal sequence
- The decision mechanism provides ordered list of the candidate words as well as corresponding speech locations.
- The final result is best candidate and its associated endpoint

- Advantages & disadvantages
  - Requires heavy computations
  - It offers higher detection accuracy than the explicit approach

3. Hybrid approach

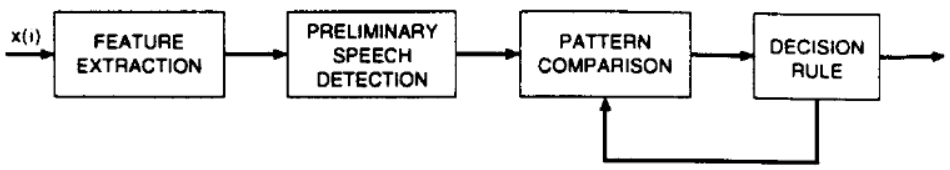


Figure 6.3 block diagram of hybrid approach [1]

IDOL

- This is the combination of both implicit and explicit approaches
- Uses the explicit method to obtain several end points sets for recognition processing and implicit method to choose the alternatives
- The most likely candidate word and the corresponding end points as in implicit approach, are provided by the decision box
- Computational load is equivalent to explicit method
- Accuracy of hybrid approach is comparable to implicit method

The basic algorithm used obtain an estimate of the endpoint involves feed forward processing of the measured short time energy level.

- A block diagram is shown below

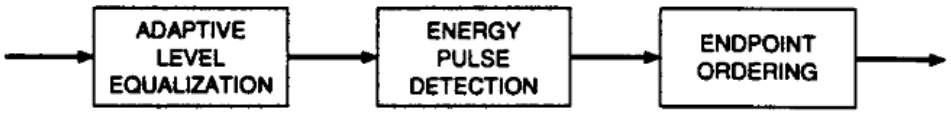


Figure 6.4 block diagram of typical speech activity detection algorithm [1]

- Adaptive level equalization module: estimates the level of the acoustic background and uses the result to equalize the measured energy contour
- Preliminary energy pulses are detected from the equalized energy contour
- Finally, these energy pulse end points are ordered to determine the possible sets of word end point pairs

6.3 DISTORTION MEASURE- MATHEMATICAL CONSIDERATION

- A key component of pattern matching is the measurement of dissimilarity between two feature vectors.
- The measurement of dissimilarity satisfies three metric properties such as Positive definiteness property, Symmetry property and Triangular inequality property

- Assume we have vector space X with two feature vector p and q in it

  - Positive definiteness property

    $0 \le d(p,q) < \infty$ for $p,q \in X$ and $d(p,q)=0$ if and only if $p=q$

  - Symmetry property

    $d(p,q) = d(q,p)$ for $p,q \in X$

  - Triangular inequality property

    $d(p,q) \le d(p,r)+d(q,r)$ for $p,q,r \in X$

- Each metric has three main characteristics such as computational complexity, analytical tractability and feature evaluation reliability
- The distortion measure is one which satisfies only the positive definiteness property of the measurement of Dissimilarity
- Distance in speech recognition means measure of dissimilarity
- For speech processing, an important consideration in choosing a measure of distance is its subjective meaningfulness
- The mathematical measure of distance to be useful in speech processing should consider the lingustic characteristics.
- For example a large difference in the waveform error does not always imply large subjective differences.

6.4 DISTORTION MEASURE – PERCEPTUAL CONSIDERATION

- The choice of an appropriate measure of spectral dissimilarity is the concept of subjective judgment of sound difference or phonetic relevance.
- Spectral changes that keep the sound the same perceptually should be associated with small distances.

IDOL

- Spectral changes that keep the sound the different perceptually should be associated with large distances
- Below we will compare two spectral representation S(w) and S'(w) using distance measure d(S,S').
- The spectral changes that do not fundamentally change the perceived sound include
    1. Spectral tilt

       $S'(w)=S(w).w^{\alpha}$

       α : spectral tilt factor
    2. High pass filtering

       $S'(w)=S(w)|H_{HP}(e^{jw})|^2$

       $H_{HP}(e^{jw})$: filter that is unity in value above some cut off frequency ,that is below the range of first formant frequency and attenuates the signal sharply for frequencies below the cut off frequency
    3. low pass filtering

       $S'(w)=S(w)|H_{LP}(e^{jw})|^2$

       $H_{LP}(e^{jw})$: filter that is unity in value below some cut off frequency ,that is above the range of third formant frequency and attenuates the signal sharply for frequencies above the cut off frequency
    4. notch filtering

       $S'(w)=S(w)|H_{N}(e^{jw})|^2$

       $H_{N}(e^{jw})$: is unity except at a narrow range of frequencies where the signal is greatly attenuated

- For the above cases the spectral content of two signal are phonetically same (same sound) then the distance measure d(S,S') is ideally very small
- Following are the spectral changes due to large phonetic distance include
    - Significant differences in formant locations.

IDOL

- the spectral resonance of S(w) and S'(w) occur at very different frequencies.

▪ Significant differences in formant bandwidths.

- the frequency widths of spectral resonance of S(w) and S'(w) are very different.

- For each of these cases sounds are different so the spectral distance measure d(S,S') is ideally very large
- To relate a physical measure of difference to subjective perceived measure of difference it is important to understand auditory sensitivity to changes in frequencies, bandwidths of the speech spectrum, signal sensitivity and fundamental frequency.
- This sensitivity is presented in the form of just discriminable change – the change in a physical parameter such that the auditory system can reliably detect the change as measured in standard listening test.
- To describe the just discriminable changes includes the difference limen(DL), just noticeable difference(JND) an differential threshold.

6.5 Spectral Distortion Measure

- Measuring the difference between two speech patterns in terms of average or accumulated spectral distortion is reasonable way of comparing patterns, both in terms of its mathematically tractability and its computational efficiency
- Perceived sound differences can be interpreted in terms of differences of spectral features

6.5.1 Log Spectral Distance

- Consider two spectra S(w) and S'(w). The difference between two spectra on a log magnitude versus frequency scale is defined by

$$V(\omega) = \log S(\omega) - \log S'(\omega) \qquad \text{---- (1)}$$

- A distance or distortion measure between S and S' can be defined by

$$d(\mathrm{S},\mathrm{S}')^{\mathrm{p}} = (d_p) = \int_{-\pi}^{\pi} |V(\omega)|^P \frac{d\omega}{2\pi} \qquad \text{---- (2)}$$

Here in above equation (2)

- For P=1 , defines the mean absolute log spectral distortion
- For P=2, defines the rms log spectral distortion that has application in many speech processing systems
- For P approaches to infinity , reduces to the peak log spectral distrotion

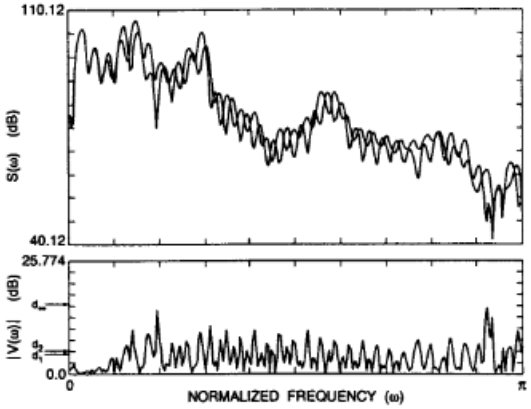
- Since perceived loudness of a signal is approximately logarithmic, the log spectral distance family appears to be closely tied to the subjective assessment of sound differences; hence, it is perceptually relevant distortion measure

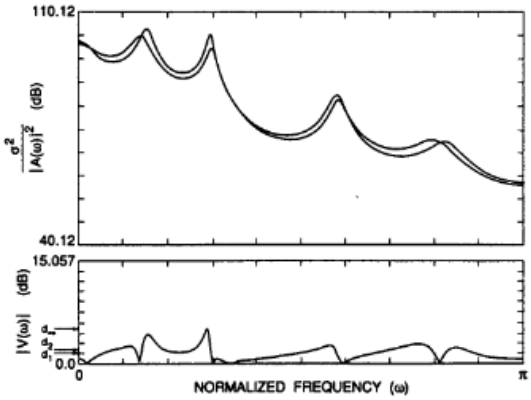
- distortion is calculated using short time DFT power spectra and by LPC model spectra i.e. all pole smoothed model spectra

Below figures shows the example for typical variation between vowel, fricative, and between two grossly different sounds.
Top plot shows overlay of comparing two spectra
Bottom plot shows magnitude of log spectral difference |v(w)|

1. variation between vowel /æ/

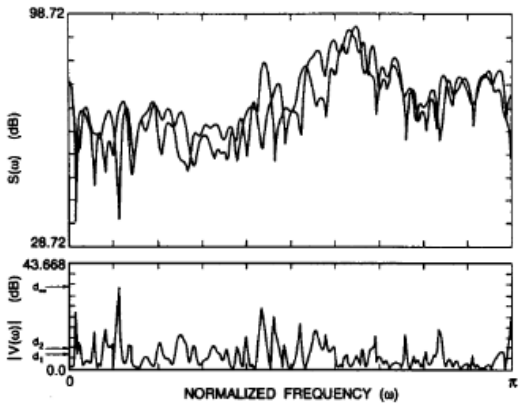
IDOL

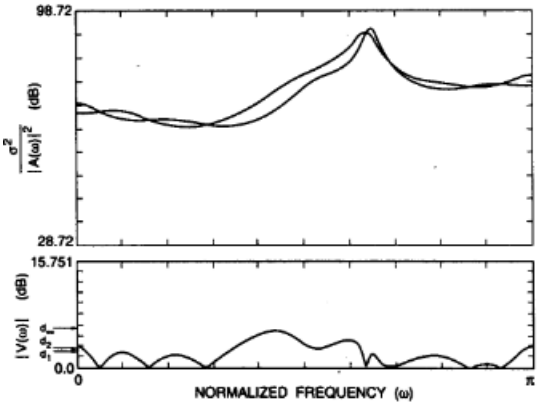




Short time DFT power spectra LPC model power spectra

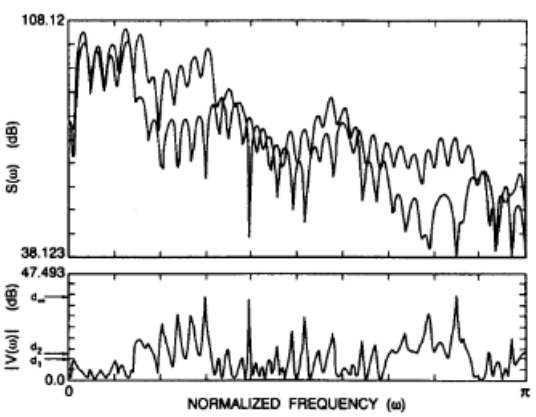
2. variation between fricative /sh/



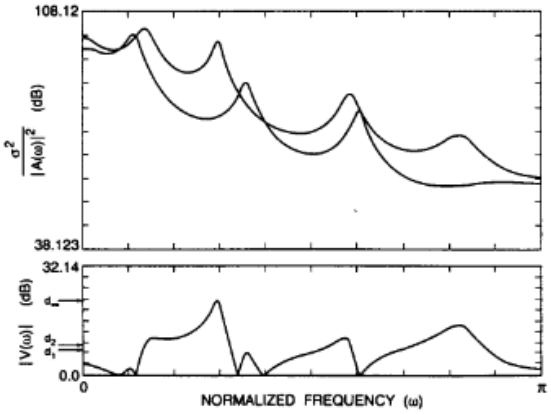


Short time DFT power spectra LPC model power spectra

3. variation between two grossly different sound , /æ/ and /i/



Short time DFT power spectra



LPC model power spectra

- From the above diagram we can see that the log spectral difference |v(w)| as computed from DFT is irregular
- The log spectral difference between the two all pole models of the spectra is much smoother than the DFT
- The smooth spectral difference allows a closer examination of the properties of the distortion measure.

6.5.2 Cepstral Distance

- The complex cepstrum of a signal is defined as the Fourier transform of the log of the signal cepstrum.
- For the Cepstral coefficients we use the rms log spectral distance.
- The fourier series representation of log S(w)

$$\log S(\omega) = \sum_{n=-\infty}^{\infty} c_n e^{-jn\omega} \qquad \text{---- (a)}$$

Where $c_n = c_{-n}$ are real and referred as cepstral coefficient

IDOL

$$c_0 = \int_{-\pi}^{\pi} \log S(w) \frac{dw}{2\pi} \qquad \text{---- (b)}$$

Consider the RMS log spectral distance

$$d_2^{\,2} = \int_{-\pi}^{\pi} \left| \log S(\omega) - \log S'(\omega) \right|^2 \frac{d\omega}{2\pi}$$

$$= \sum_{n=-\infty}^{\infty} (c_n - c_n')^2 \qquad \text{---- (c)}$$

Where c_n and c'_n are the cepstral coefficient of S(w) and S'(w) resp.

- Since the cepstrum is a decaying sequence, the summation in equation (c) does not require an infinite number of terms.
- The LPC models that represent the highly smoothed envelopes of the speech spectra, the equation (c) is truncated to small number of terms
- The number of terms must be no less than p (cepstral coefficients) for truncated distance
- The truncated cepstral distance is defined as

$$d_c^{\,2}(L) = \sum_{n=1}^{L} (c_n - c_n^{'})^2$$

- The truncated cepstral distance is a very efficient method for estimation the rms log spectral distance when the spectrum is represented by all pole model

6.5.3 Weighted cepstral distances and Liftering

- There are other usefulness of cepstral distance beyond the method for estimating the rms log spectral distance

IDOL

- Several other properties of the cepstrum when properly utilized are beneficial for speech recognition applications
- It can be shown that under certain regular conditions, the cepstral coefficients except c_0 have
  - Zero means
  - Variance essentially inversely proportional to the square of the coefficient index, such that

$$E\{c_n^2\} \sim \frac{1}{n^2} \quad \text{---- (a)}$$

- If n^2 factor incorporated into cepstral distance to normalize the contribution from each cepstral term the distance of $d_2^2 = \sum_{n=-\infty}^{\infty} (c_n - c_n')^2$ becomes

$$D^2_{2w} = \sum_{n=-\infty}^{\infty} n^2 (c_n - c_n')^2$$

$$= \sum_{n=-\infty}^{\infty} (nc_n - nc_n')^2 \quad \text{---- (b)}$$

- The distance defined in equation (b) is called “root power sum” distance
- The variability of higher capstral coefficients are more influenced by the inherent artifacts of LPC analysis than that of lower cepstral coefficients.
- For speech recognition, therefore, suppression of higher cepstral coefficients in the calculation of a cepstral distance should lead to a more reliable measurement of spectral differences than otherwise
- The LPC spectrum also includes components that are strong functions of the speaker’s glottal shape and vocal cord duty cycles.
- These components affects mainly the first few cepstral coefficients.
- For speech recognition the phonetic content of the sound is important and not these components so these components are need to be de-emphasized

IDOL

- A cepstral weighting or liftering procedure, w(n) can therefore be designed to control the non information-bearing cepstral variabilities for reliable discrimination of sounds.
- The index weighting as used in equation (b) is the example of the simple form of cepstral weighting

The following figure shows the effect of liftering on the original LPC log spectrum as a function of lifter length L (from L = 8 to 16)

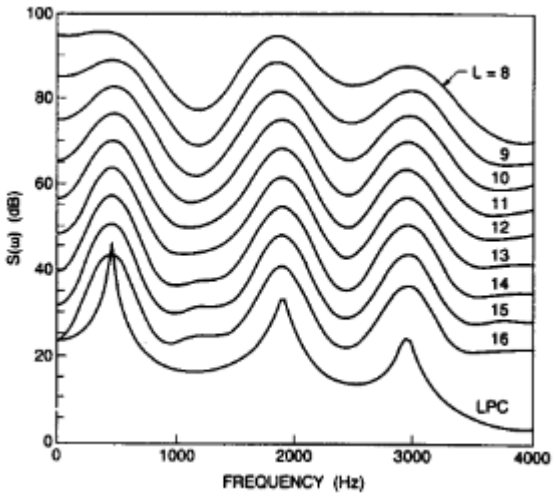


Figure 6.5 effect of cepstral liftering on the log LPC spectrum as a function of lifter length L (from L = 8 to 16) [1]

- The original sharp spectral peaks are highly sensitive to the LPC analysis condition and the resulting peakiness creates unnecessary sensitivity in spectral comparison
- The liftering process tends to reduce the sensitivity without altering the fundamental “formant” structure.
- A useful form of weighted cepstral distance is

$$d_{cw}^2 = \sum_{n=1}^{L} (w(n)c_n - w(n)c_n^{'})^2$$

- Where w(n) is any lifter function.

Itakura and Saito

- The log spectral difference V(w) is defined by V(w) = log S(w) – log S'(w) is the basis of many distortion measures
- The Itakura–Saito distance (or Itakura–Saito divergence) is a measure of the difference between an original spectrum and an approximation of that spectrum.
- The distortion measure proposed by Itakura and Saito in their formulation of linear prediction as an approximate maximum likelihood estimation is

$$d_{IS}(S,S') = \int_{-\pi}^{\pi} \left[e^{V(w)} - V(w) - 1 \right] \frac{dw}{2\pi}$$

$$d_{IS}(S,S') = \int_{-\pi}^{\pi} \frac{S(w)}{S'(w)} \frac{dw}{2\pi} - \log \frac{\sigma_{\infty}^{\ 2}}{\sigma'^{\ 2}_{\infty}} - 1$$

Where $\sigma_{\infty}^{\ 2}$ and $\sigma'^{\ 2}_{\infty}$ are the predication error of S(w) and S'(w) resp. as defined $\sigma_{\infty}^{\ 2} = \exp\left\{ \int_{-\pi}^{\pi} \log S(w) \frac{dw}{2\pi} \right\}$

- The Itakura Saito distortion measure can be used to illustrate the spectral matching properties by replacing S'(w) with the p[th] order all pole spectrum leads to

$$d_{Is}\left(S, \frac{\sigma^2}{\left| A\left(e^{jw}\right) \right|^2} \right) = \frac{1}{\sigma^2} \int_{-\pi}^{\pi} S(w) \left| A\left(e^{jw}\right) \right|^2 \frac{dw}{2\pi} - \log \sigma^2_{\infty} + \log \sigma^2 - 1$$

Where σ^2 is the gain

IDOL

Itakura distortion

- The itakura distortion meaure is

$$d_I\left(\frac{1}{|A_p|^2},\frac{1}{|A|^2}\right)=\log\left\{\int_{-\pi}^{\pi}\frac{|A(e^{jw})|^2}{|A_p(e^{jw})|^2}\frac{dw}{2\pi}\right\}$$

- Which is defined for only two unity gain all-pole spectra.
- The role of gain term is not explicit in Itakura distortion

6.5.4 Likelihood Distortions

- Gain independent distortion measure called likelihood ration distortion can be derived directly from Itakura Saito distortion measure

$$d_I\left(\frac{1}{|A_p|^2},\frac{1}{|A|^2}\right)=d_{LR}\left(\frac{1}{|A_p|^2},\frac{1}{|A|^2}\right)$$

- When the distortion is very small the Itakura distortion measure is not very different from the likelihood distortion measure.

6.5.5 Variations of likelihood distortions

- Unlike cepstral distance likelihood distortions are asymmetric.
- To symmetries the distortion measure there are two methods
  - COSH distortion
  - Weighted likelihood distortion

IDOL

COSH distortion

- COSH distortion is given by

$$d_{COSH} = \int_{-\pi}^{\pi} \cosh\left[\log \frac{S(w)}{S'(w)}\right] \frac{dw}{2\pi} - 1$$

- The COSH distortion is almost identical to twice the log spectral distance for small distortions
- COSH measure has not been extensively used in speech recognition application

Weighted likelihood ratio distortion

- The purpose of weighting is to take the spectral shape into account as a weighting function such that different spectral components along frequency axis can be emphasized or de-emphasized to reflect some of the observed perceptual effects

$$d_{WLR} = \sum \left[\frac{\hat{r}(n)}{\sigma^2} - \frac{\hat{r}'(n)}{\sigma'^2}\right](c_n - c'_n)$$

Where cn and c'n are cepstral coefficient of $\log \frac{1}{|A|^2}$ and $\log \frac{1}{|A'|^2}$

And $\hat{r}(n)$ and $\hat{r}'(n)$ are autocorrelation sequences for

$\frac{\sigma^2}{|A|^2}$ and $\frac{\sigma'^2}{|A'|^2}$ respectively .

References and further Reading

[1] L.R. Rabiner , B. H. Juang , "Fundamentals of Speech Recognition", Prentice-Hall, Englewood Cliffs, N.J.

IDOL

CHAPTER 7

PATTERN COMPARISON TECHNIQUES PART -B

7.1 Incorporation of spectral dynamic feature into distortion measure

Spectral transition play important role in human speech perception, the portion of the utterance where spectral variation was locally maximum contained the most important phonetic information in the syllable.

Therefore it is reasonable that the use of such variational feature of the spectrum in speech comparison should contribute significantly overall recognition performance

Dynamic feature of speech is represented by a time differential log spectrum

A first order differential log spectrum is defined by

$$\frac{\partial \log S(\omega,t)}{\partial t} = \sum_{n=-\infty}^{\infty} \frac{\partial c_n(t)}{\partial t} e^{-jn\omega},$$

Where $c_n(t)$ is the cepstrum at time t.

The time derivative $\partial c_n(t)/\partial t$ is obtained by polynomial approximation.

This is done by fitting the cepstral trajectory with polynomials over a finite of trajectory

Fitting the cepstral trajectory c(t) by a second order polynomial, we Choose h1, h2, h3 such that the fitting error E is minimized.

$$E = \sum_{t=-M}^{M} [c(t) - (h_1 + h_2 t + h_3 t^2)]^2$$

Differentiating E with respect to h1, h2, and h3 and setting to zero results in 3 equations:

$$\sum_{t=-M}^{M} [c(t) - h_1 - h_2 t - h_3 t^2] = 0$$

$$\sum_{t=-M}^{M} [c(t)t - h_1 t - h_2 t^2 - h_3 t^2] = 0$$

$$\sum_{t=-M}^{M} [c(t)t^2 - h_1 t^2 - h_2 t^3 - h_3 t^4] = 0$$

IDOL

The solutions to these equations are:

$$h_2 = \frac{\sum_{t=-M}^{M} tc(t)}{T_M}$$

$$h_3 = \frac{T_M \sum_{t=-M}^{M} c(t) - (2M+1)\sum_{t=-M}^{M} r^2 c(t)}{T_M^2 - (2M+1)\sum_{t=-M}^{M} t^4}$$

And

$$h_1 = \frac{1}{2M+1}\left[\sum_{t=-M}^{M} c(t) - h_3 T_M\right],$$

Where

$$T_M = \sum_{t=-M}^{M} t^2.$$

The first and second time derivatives of c_n can be obtained by differentiating the fitting curve, giving

$$\frac{\partial\, c_n(\tau+t)}{\partial t}\Big|_{t=0} \cong h_2 = \sum_{t=-M}^{M} t\, c_n(\tau+t)/T_M$$

And

$$\frac{\partial^2 c_n(\tau+t)}{\partial t^2}\Big|_{t=0} \cong 2h_3$$

$$= \frac{2\left\{T_M\left[\sum_{t=-M}^{M} c(\tau+t)\right] - (2M+1)\left[\sum_{t=-M}^{M} t^2 c(\tau+t)\right]\right\}}{T_M^2 - (2M+1)\left[\sum_{t=-M}^{M} t^4\right]}$$

A differential spectral distance:

$$d^2_{2\delta^{(1)}} = \int_{-\pi}^{\pi} \left| \frac{\partial \log S(\omega,t)}{\partial t} - \frac{\partial \log S'(\omega,t)}{\partial t} \right|^2 \frac{d\omega}{2\pi}$$

IDOL

$$\cong \sum_{n=-\infty}^{\infty} (\delta_n^{(1)} - \delta'^{(1)}_n)^2 ,$$

A second differential spectral distance:

$$d^2_{2\delta^{(2)}} = \int_{-\pi}^{\pi} \left| \frac{\partial^2 \log S(\omega,t)}{\partial t^2} - \frac{\partial^2 \log S'(\omega,t)}{\partial t^2} \right|^2 \frac{d\omega}{2\pi} \cong \sum_{n=-\infty}^{\infty} (\delta_n^{(2)} - \delta'^{(2)}_n)^2$$

where $\delta_n^{(1)} = \left. \frac{\partial c_n(\tau+t)}{\partial t} \right|_{t=0}$ and $\delta_n^{(2)} = \left. \frac{\partial^2 c_n(\tau+t)}{\partial t^2} \right|_{t=0}$

Combining the first and second differential spectral distances with the Cepstral distance results in:

$$d^2_{2\Delta} = \gamma_1 d_2^2 + \gamma_2 d^2_{2\delta^{(1)}} + \gamma_3 d^2_{2\delta^{(2)}},$$

Where $\gamma_1, \gamma_2 and \gamma_3$ are the weights used to adjust the respective significance of the associated distance components, usually $\gamma_1 + \gamma_2 + \gamma_3 = 1$

The differential spectral distance provide significant additional discriminability for speech pattern comparison

The cepstral weighting or liftering can be applied to the differential cepstrum in much the same manner as it is applied to the cepstrum.

Therefore A weighted differential cepstral distance:

$$d^2_{2w\delta} = \int_{-\pi}^{\pi} \left| \frac{\partial^2 \log S(\omega,t)}{\partial t \, \partial \omega} - \frac{\partial^2 \log S'(\omega,t)}{\partial t \, \partial \omega} \right|^2 \frac{d\omega}{2\pi}$$

$$\cong \sum_{n=-\infty}^{\infty} n^2 (\delta_n^{(1)} - \delta_n'^{(1)})^2 .$$

Linear operation such as scaling and differentiation can be interchanged in order and thus weighted differential cepstral distance is the same as differential cepstral distance defined on liftered spectra.

There are many ways of augmenting the conventional cepstral distances with dynamic spectral feature

IDOL

7.2 Time alignment and normalization

- When comparing different tokens of the same utterance, speech rate variation as well as duration variation should not contribute to the (linguistic) dissimilarity score.
- Thus there is a need to normalize speaking rate fluctuations in order for the utterance comparison to be meaningful before a recognition decision can be made. T
- he need for time alignment arises not only because different utterances of the same word will generally be of different durations, but also because phonemes within words will also be of different durations across the utterance

- Consider two speech patterns X and Y with spectral sequences

  $X = (x_1, x_2, \ldots\ldots, x_{tx})$

  $Y = (y_1, y_2, \ldots\ldots, y_{tx})$

  - x_i and y_i are parameter vectors of the short time acoustic feature.
  - i_x and i_y denotes the time indices of X and Y
  - The duration T_x and T_y need not be identical
  - The dissimilarity between X and Y defined by $d(i_x, i_y)$ where $i_x = 1,2,\ldots,T_x$ and $i_y = 1,2,\ldots,T_y$

- The sequential order of the sounds is critical in the definition of an utterance it is necessary that certain order constraints are satisfied by the indices of spectral being compared
- The interaction between these sequential constraints and the natural speaking rate variation constitutes the time alignment and normalization problem.
- The simplest solution for this problem is linear time normalization.
- In linear time normalization the dissimilarity between X and Y is defined as

IDOL

$$d(x,y) = \sum_{i_x=1}^{T_x} d(i_x, i_y)$$

Where i_x, i_y satisfy

$$i_y = \frac{T_y}{T_x} i_x$$

- In Linear time normalization the speaking rate variation is proportional to the duration of the utterance and is independent of sound being spoken.
- Therefore evaluation of distortion measure takes place as along the straight line of rectangle as shown below

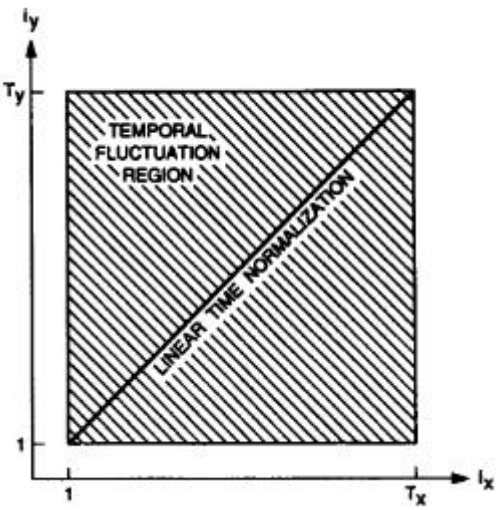


Figure 7.1: linear time alignment for two sequence of different durations[1]

- Here each point in the (ix,iy) plane along the diagonal represents the distance between sectral feature vectors of X and Y at frames ix and iy.

- Time alignment and normalization scheme involves the use of two warping functions ϕ_x and ϕ_y which relate to indices of two speech pattern i_x and i_y to a common "normal" time axis k, i.e.

  $i_x = \phi_x(k)$ k = 1,2,…,T

  And

  $i_y = \phi_y(k)$ k = 1,2,…,T

- The global pattern dissimilarity measure $d_\phi = (X, Y)$ can be defined based on the warping function pair $\phi = (\phi_x, \phi_y)$ as the accumulated distortion over the entire utterance namely ,

  $$d_\phi(\chi, y) = \sum_{k=1}^{T} d(\phi_x(k), \phi_y(k)) m(k) / M_\phi$$

- Where $d(\phi_x(k), \phi_y(k))$ is a short time spectral distoration defined for $x_{\phi_x(k)}$ and $y_{\phi_y(k)}$, m(k) id a nonnegative(path) weighting coefficient and M_ϕ is a (path) normalizing factor
- We need to specify the path $\phi = (\phi_x, \phi_y)$ to complete the definition of dissimilarity measure for the pairs of patterns
- There are large number of possible warping function pairs.
- The key issue which path is then chosen such that overall path dissimilarity can be measured
- One way is To define the dissimilarity d(X,Y) as the minimum of d_ϕ(X,Y) over all possible path such that

  $$d(\chi, y) \overset{\Delta}{=} \min_{\phi} d_\phi(\chi, y),$$

  Where ϕ must satisfy a set of requirements

IDOL

- The main point here is that finding the “best” alignment between a pair of patterns
- Finding the best path requires solving a minimizing problem to evaluate the dissimilarity between two speech patterns
- The specific form of the accumulated distortion suggest the dynamic programming techniques

7.2.1 Dynamic Programming-Basic Considerations

Dynamic programming is tool used in operation research for solving sequential decision problems

Below we will discuss two typical problems in which dynamic programming is used.

1. Optimal path problem
    - For every pair of points (i,j) We define to be a nonnegative cost that represents the cost of moving directly from the i^{th} point to the j^{th} point in one step.
    - According to Bellman:

      An optimal policy has the property that, whatever the initial state and decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision
    - We are interested in obtaining the optimal sequence of moves and the associated minimum cost from any point i to any other point j in As many points as necessary. We have

$$\varphi(i, j) = \min_{\ell} [\varphi(i, \ell) + \varphi(\ell, j)],$$

    - To determine the minimum cost path between points i and j, the following dynamic program is used:

IDOL

$$\varphi_1(i,\ell) = \zeta(i,\ell) \qquad \ell = 1,2,\ldots,N$$

$$\varphi_2(i,\ell) = \min_k (\varphi_1(i,k) + \zeta(k,\ell)), \qquad \begin{matrix} k = 1,2,\ldots,N \\ \ell = 1,2,\ldots,N \end{matrix}$$

$$\varphi_3(i,\ell) = \min_k (\varphi_2(i,k) + \zeta(k,\ell)), \qquad \begin{matrix} k = 1,2,\ldots,N \\ \ell = 1,2,\ldots,N \end{matrix}$$

$$\vdots$$

$$\varphi_s(i,\ell) = \min_k (\varphi_{s-1}(i,k) + \zeta(k,\ell)), \qquad \begin{matrix} k = 1,2,\ldots,N \\ \ell = 1,2,\ldots,N \end{matrix}$$

$$\varphi(i,j) = \min_{1 \le s \le S} \varphi_s(i,j)$$

Where $\varphi_s(i,\ell)$ is the s-step best path from point i to point l and S is the maximum number of steps allowed in path

2. Synchronous decision problem
   - The objective now is to find the optimal sequence of fixed number M of moves starting from point i to ending at point j and the associated minimum cost $\varphi_m(i,\ell)$
   - Suppose that (m+1)th move is to go to point n then $\varphi_{m+1}(i,n)$

$$\varphi_{m+1}(i,n) = \min_{\ell} [\varphi_m(i,\ell) + \zeta(\ell,n)]$$

   - The above equation describes a recursion allowing the optimal path search to be conducted incrementally
   - The algorithm can be summarized as
     1. Initialization :

$$\varphi_1(i,n) = \zeta(i,n)$$
$$\xi_1(n) = i$$
$$for\ n = 1,2,\ldots,N.$$

     2. Recursion :

IDOL

$$\varphi_{m+1}(i,n) = \min_{1\le \ell \le N} [\varphi_m(i,\ell) + \zeta(\ell,n)]$$

$$\xi_{m+1}(n) = \arg \min_{1\le \ell \le N} [\varphi_m(i,\ell) + \zeta(\ell,n)]$$

$$for\ n = 1,2,\ldots,N\ and\ m = 1,2,\ldots,M-2$$

3. Termination :

$$\varphi_M(i,j) = \min_{1\le \ell \le N} [\varphi_{M-1}(i,\ell) + \zeta(\ell,j)]$$

$$\xi_M(j) = \arg \min_{1\le \ell \le N} [\varphi_{M-1}(i,\ell) + \zeta(\ell,j)]$$

4. Path backtracking

$$optimal\ path = (i, i_1, i_2, \ldots, i_{M-1}, j),$$

where

$$i_m = \xi_{m+1}(i_{m+1}), \qquad m = M-1, M-2, \ldots, 1$$

with $i_M = j.$

- The algorithm as a result of optimality principle keeps track of only N paths ending at each of N points at the end of every potential move.
- When reached to destination the optimal path and associated result of the algorithm without having to reexamine any of the previously incurred partial cost

7.2.2 Time Normalization Constraint

To achieve a meaningful alignment in terms of time normalization, some constraints on the warping functions are necessary.

Typical constraints in this context are:

- Endpoint Constraints
- Monotonicity Conditions
- Local Continuity Constraints
- Global Path Constraints
- Slope Weightin

IDOL

End point constraint

- Well-defined endpoints of the two sequences that mark as beginning and endpoint in the (audio) stream are assumed and are fixed together
- End point information is usually derived as the result of speech detection operation
- End points are considered as priori and temporal variation occur within the range defined by endpoint

$$\begin{aligned} &\textit{beginning point} && \phi_x(1) = 1 && \phi_y(1) = 1 \\ &\textit{ending point} && \phi_x(T) = T_x && \phi_y(T) = T_y \end{aligned}$$

- This constraint is usually used in theoretically introducing Dynamic Time Warping, but is relaxed or even removed in practice, depending on the application domain.

Monotonicity Conditions

- The temporal order is important for linguistic meaning.
- To maintain the temporal order while performing time normalization ,therefore a monotonicity condition is introduced of the form:

$$\begin{aligned} \phi_x(k+1) &\geq \phi_x(k) \\ \phi_y(k+1) &\geq \phi_y(k). \end{aligned}$$

- It implies that any path along which $d_\phi(X,Y)$ is evaluated will not have a negative slope.

Local Continuity Constraints

- Time normalization should not result in skipping of any important information-bearing sound segment.

IDOL

- It can take many forms are based on heuristics. an example proposed by Sakoe and Chiba is

$$\phi_x(k+1) - \phi_x(k) \le 1$$
$$\phi_y(k+1) - \phi_y(k) \le 1$$

- Such constraint are quiet complicated and therefore convenient to express them in terms of incremental path changes
- We define a path P as a sequence of moves, each specified by a pair of coordinate increments,

$$\mathrm{P} \rightarrow (p_1, q_1)(p_2, q_2)...(p_T, q_T)$$

- Following figure illustrates the three paths P_1,P_2,P_3 which can be specified by

$P_1 \rightarrow (1,1)(1,0)$, $P_2 \rightarrow (1,1)$ and $P_3 \rightarrow (1,1)(0,1)$

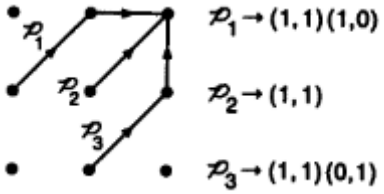


Figure 7.2:an exampe of local continuity constraint expressed oin terms of cordinate increment [1]

- For a path that begins at (1,1), which point we designate k=1, we normally set (as if the path originates from (0,0)) and have:

$$\phi_x(k) = \sum_{i=1}^{k} p_i \qquad \phi_y(k) = \sum_{i=1}^{k} q_i .$$

If paths ends at (T_x,T_y) then

$$\sum_{k=1}^{T} p_k = T_x \qquad \sum_{k=1}^{T} q_k = T_y$$

IDOL

Global Path Constraints

- Due to the local continuity constraints, certain portions of the (ix , iy) plane can not be reached
- For each type of local constraints, the allowable regions can be Defined using the following two parameters:

$$Q_{\max} = \max_{\ell}\left[\sum_{i=1}^{T_\ell} p_i^{(\ell)} \Big/ \sum_{i=1}^{T_\ell} q_i^{(\ell)}\right]$$

$$Q_{\min} = \min_{\ell}\left[\sum_{i=1}^{T_\ell} p_i^{(\ell)} \Big/ \sum_{i=1}^{T_\ell} q_i^{(\ell)}\right],$$

- Where *l* signifies the index of the allowable path P*l*, in the constraint set and T*l* is the total number of moves of moves in P*l*
- We can define the global path constraints as follows:

$$1+\frac{[\phi_x(k)-1]}{Q_{\max}} \le \phi_y(k) \le 1+Q_{\max}[\phi_x(k)-1]$$

---- (a)

$$T_y+Q_{\max}[\phi_x(k)-T_x] \le \phi_y(k) \le T_y+\frac{[\phi_x(k)-T_x]}{Q_{\max}}$$

---- (b)

- Equation (a) specifies the Range of points that can be reached from the beginning.
- Equation (b) specifies the Range of points that have a legal path to the ending.
- The additional global path constraint proposed by the Sakoe and Chiba is

$$|\phi_x(k)-\phi_y(k)| \le T_0$$ --- (c)

- Where T_0 is the maximum allowable absolute time deviation between the two patterns

IDOL

- The above equation (c) are called range-limiting constraints because they limit absolute difference in the warped time scales

Slope Weighting

- Slope weighting add another dimension of control in the search for optimal warping path
- With the local continuity constraints the allowed paths are determined.
- Not all of the allowed paths have equal probability. A slope weighting can weight special paths as more or less desirable
- The weighting function can be designed to implement an optimal discriminant analysis for improved recognition accuracy. Set of four types of slope weighting proposed by Sakoe and Chiba

Type (a) $m(k) = \min[\ \phi_x(k) - \phi_x(k-1), \phi_y(k) - \phi_y(k-1)\ ]$

Type (b) $m(k) = \max[\ \phi_x(k) - \phi_x(k-1), \phi_y(k) - \phi_y(k-1)\ ]$

Type (c) $m(k) = \phi_x(k) - \phi_x(k-1)$

Type (d) $m(k) = \phi_x(k) - \phi_x(k-1) + \phi_y(k) - \phi_y(k-1)$

- It is assumed for initialization purpose that $\phi_x(0) - \phi_y(0) = 0$
- The Accumulated distortion also requires an overall normalization. Customarily:

$$M_\phi = \sum_{k=1}^{T} m(k).$$

- For type (c) and (d) slope weighting the normalizing factors would then become

$$M_\phi^{(c)} = \sum_{k=1}^{T} [\phi_x(k) - \phi_x(k-1)] = \phi_x(T) - \phi_x(0) = T_x$$

And

$$M_\phi^{(d)} = \sum_{k=1}^{T} [\phi_x(k) - \phi_x(k-1) + \phi_x(k) - \phi_x(k-1)]$$
$$= \phi_x(T) - \phi_x(0) + \phi_y(T) - \phi_y(0) = T_x + T_y$$

- Respectively, independent of warping functions and associated constraints

- Typically, for types (a) and (b) slope weightings, we arbitrarily set:

$$M_\phi^{(a)} = M_\phi^{(b)} = T_x$$

7.2.3 Dynamic Time-Warping Solution

- Due to endpoint constraints, we can write equation

$$d(\chi, y) \stackrel{\Delta}{=} \min_{\phi} d_\phi(\chi, y),$$

In terms of Tx and Ty as

$$M_\phi d(x, y) \stackrel{\Delta}{=} D(T_x, T_y)$$

$$= \min_{\phi_x, \phi_y} \sum_{k=1}^{T} d(\phi_x(k), \phi_y(k)) m(k)$$

- Similarly, the minimum partial accumulated distortion along a path Connecting (1,1) and (ix, iy) is:

$$D(i_x, i_y) \stackrel{\Delta}{=} \min_{\phi_x, \phi_y, T'} \sum_{k=1}^{T'} d(\phi_x(k), \phi_y(k)) m(k),$$

where $\phi_x(T') = i_x$ and $\phi_y(T') = i_y$

- The dynamic programming recursion with constraint becomes

$$D(i_x, i_y) = \min_{(i'_x, i'_y)} [D(i'_x, i'_y) + \zeta((i'_x, i'_y), (i_x, i_y))],$$

$$\zeta((i'_x, i'_y), (i_x, i_y)) = \sum_{\ell=0}^{L_s} d(\phi_x(T' - \ell), \phi_y(T' - \ell)) m(T' - \ell)$$

- With L_s being number of moves in path from (i'_x, I'_y) to (i_x, i_y) according to ϕ_x and ϕ_y also

$$\phi_x(T' - L_s) = i'_x \text{ and } \phi_y(T' - L_s) = i'_y$$

- Summarize the dynamic programming implementation for finding the best path through a T_x by T_y grid beginning at (1,1) and ending at (T_x, T_y) as follows

1. Initialization :

$$D_A(1,1) = d(1,1) m(1).$$

IDOL

2. Recursion :

$$For\ 1 \le i_x \le T_x, 1 \le i_y \le T_y$$

$$D_A(i_x, i_y) = \min_{(i'_x, i'_y)} [D_A(i'_x, i'_y) + \zeta((i'_x, i'_y), (i_x, i_y))],$$

3. Termination :

$$d(X, Y) = \frac{D_A(T_x, T_y)}{M_\phi}$$

- The key idea is that the recursion step is done for all local paths that reach (i_x,i_y) in exactly one step using the local path constraint chosen for the implementation.

- The following figure shows the 40 x 40 grid that are used in recursion when set of local path constraint that allow 2 to 1 time expansion and a 2 to 1 time scale contraction.
- There are about (T_x ,T_y)/3 grid points fall within the discrete parallelogram

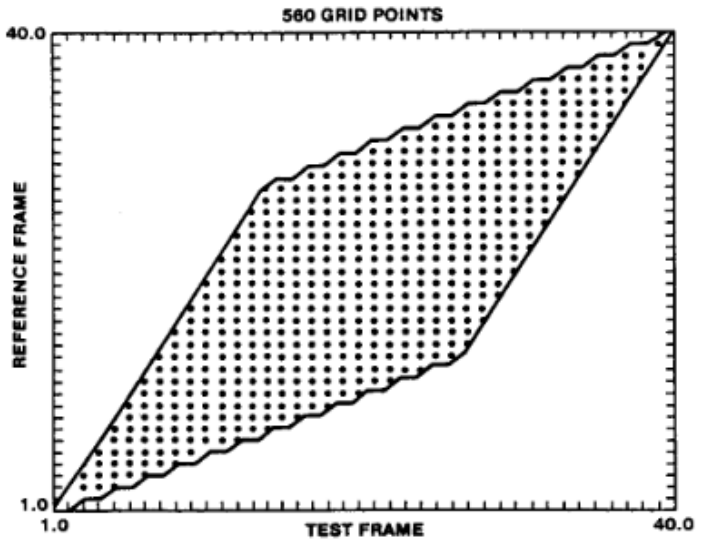


Figure 7.3:set of allowable grid points for dynamic programming implementation of local path expansion and contraction by 2 to 1 [1]

- The number of grid points falling within the parallelogram is also the number of local distance calculations required for implementing the Dynamic time wrapping procedure.
- The ratio of grid allowable grid points to all grid points within T_x x T_y rectangle , when a k-to-1 time scale expansion and contraction is allowed:

$$R = \frac{\left(k - \frac{T_x}{T_y}\right)\left(k - \frac{T_x}{T_y}\right)}{(k^2 - 1)}.$$

- Thus for case k=2 with $T_x = T_y$ we get R = 1/3
- When k = 3 with $T_x = T_y$ we get R = ½
- About half grid points are allowable within a 3:1 scale expansion and contradiction.

Reference and further reading

[1]

IDOL

CHAPTER 8

SPEECH RECOGNITION SYSTEM DESIGN

AND IMPLEMENTATION ISSUES PART - A

8.0 OBJECTIVE

In this chapter we will learn

- How to make the best overall decision for unknown utterance
- What are the appropriate template training methods
- What are the discriminative methods in speech recognition
- What are the ways of recognizing speech in adverse environment

8.1 INTRODUCTION

In last two chapters we have seen pattern recognition approach to speech recognition and issues involved in detecting and comparing the speech pattern. In this chapter we are going to see how to create the referencing pattern and how to make a decision for the unknown word using resulting set of pattern dissimilarity or distortion score.

The problem of how to create referencing pattern is generally known as the training problem. The objective of training is to create pattern representation also called as template based on one or more known pattern. In this chapter we will see the template training.

The problem of how to make a decision for unknown words can be solved using the nearest-neighbour rule. The speech recognizer uses the well-known k-nearest-neighbour rule to make recognition decision. The problem here is that speech utterance that do not have the uniform duration cannot be straightforwardly analyzed in traditional vector space. Therefore we will discuss the new approach to speech recognizer training as “discriminative training” whose primary objective is to maximize the discriminability of reference representation for minimum recognition error performance.

IDOL

8.2 APPLICATION OF SOURCE CODING TECHNIQUE TO RECOGNITION

- Source coding is converting output signal of information into a sequence of binary digits i.e. bits.
- This sequence of binary digits is transmitted over communication channel and at different location or time this sequence of binary digits is used to reproduce original signal with is an acceptable level of distortion.
- The goal of source coding is achieve the minimum possible distortion for a prescribed bit rate in the binary digit sequence
- Vector quantization is efficient source coding technique, which leads to good classifier design
- If source coder is “optimally” designed for particular source, then it will achieve a lower average distortion for signals generated by source than any other code which are not designed for particular source
- This means that there is some degree of discriminability in coder design itself
- Below we will discuss how source coding techniques can be applied to speech recognition problems

8.2.1 Vector quantization and pattern comparison without time alignment

- Consider vector quantization problem
  Let,
  - t : the signal source whose output is observed at regular spaced interval in time
  - x_t: observation sequence at time t
  - C : code words i.e. $C = \{y_i\}_{i=1}^{N}$
  - d(x_t,y_i) : distortion measure between input x_t and code word y_i
  - A vector quantizer encodes $x_t$by the index i of the code word $y_i \in$ C which minimizes d(x_t,y_i)
- The objective of vector quantizer is to find best set of code words C to achieve minimum distortion E{d(x,y_i)} where x is random vector
- The vector quantizer are memoryless that encoding of the current vector $x_t$is independent of $x_{t'}$, where t’ ≠ t.
- The codebook C is designed to minimize

$$D = \frac{1}{T}\sum_{t=1}^{T} d(x_t, \hat{x}_t) \qquad \text{--- eq. 8.2.1(a)}$$

Where

$$\hat{\mathbf{x}}_t = \arg\min_{\mathbf{y}_i \in \mathbf{C}} d(\mathbf{x}_t, \mathbf{y}_i).$$

IDOL

We write D as a function of C i.e. D(C)

- The minimized average distortion is

$$D_{min}=\min_{C} D(C)$$

- If $D_{min}= D(C^0)$ then

$$E\{d(\mathbf{x}_t, \hat{\mathbf{x}}_t^0)\} < E\{d(\mathbf{x}_t, \hat{\mathbf{x}}_t')\}$$

- Where are the closet code words for x_t chosen from the optimal code book C^0 and an arbitrary codebook C' where C^0=C'.

- The concept of vector quantization can be easily applied to speech recognizer
- Suppose there are M utterance classes i.e. words, phrases etc. which is considered as information source
- Then M set of training data $\{x_t^{(i)}\}$ where i=1,2,…,M is the class index
- M codebooks $\{C^i\}_{i=1}^M$ are then designed to minimize the average distortion for M sources
- During the recognition how M codebooks are used to implement M distinct vector quantizer is shown in following figure

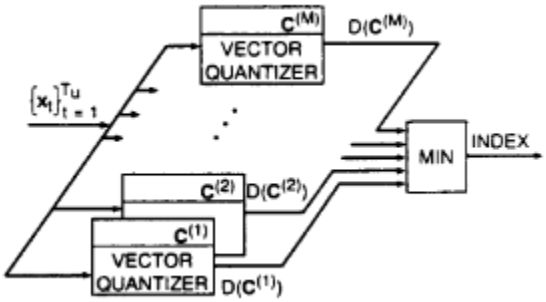


Figure 8.1 A vector quantizer based speech recognition system[1]

- The unknown utterance $\{x_t\}_{t=1}^{T_u}$ is vector quantized by all M quantizers resulting in M average distortion scores $D(C^{(i)})$, i = 1,2,…,M where

$$D(\mathbf{C}^{(i)}) = \frac{1}{T_u}\sum_{t=1}^{T_u} d(\mathbf{x}_t, \hat{\mathbf{x}}_t^{(i)}) \quad \text{---- (a)}$$

With $\hat{x}_t^{(i)} \in C^{(i)}$ satisfying

$$\hat{\mathbf{x}}_t^{(i)} = \arg\min_{\mathbf{y}_j^{(i)} \in \mathbf{C}^{(i)}} d(\mathbf{x}_t, \mathbf{y}_j^{(i)}). \quad \text{---- (b)}$$

- The utterance is recognized as class k if

$$D(\mathbb{C}^{(k)}) = \min_i D(\mathbb{C}^{(i)}).$$

- Here we see that the M codebooks are analogous to M set of reference pattern or templates and the dissimilarity defined according equation(a) and (b) where no explicit time alignment is required
- The above memoryless quantizer is not adequate for recognition performance but it is effective for simple vocabularies such as English digits.

8.2.2 Centroid Computation For VQ Codebook Design

- Vector quantization requires that codebook for particular utterance class be properly designed to minimize the average distortion
- The design algorithm of Lloyd for codebook generation consist the following steps
  - Minimum distortion labelling :

    For each input vector find the entry in given codebook C satisfying minimum distortion; group the training vector according to their associated code word indices

  - Centroid computation :

    For each group of vectors with same index label, compute new centroid which minimizes the average distortion for members of group.
- The above two steps iterate until convergence criterion is met.

- Consider a set of vectors $\{x_i\}_{i=1}^{L}$ and a distortion measure d(x,y)
- The solution to centroid problem is dependent on the choice of the distortion measure
- When x_i and y are vectors $x_i = (x_{i1}, x_{i2}, \ldots, x_{ik})$ and $y=(y_1, y_2, \ldots y_k)$ measured in K dimension space with L2 norm i.e. Euclidean distance, the centroid is the mean of the vector set

$$\overline{y} = \frac{1}{L}\sum_{i=1}^{k} x_i$$

- The above equation also implies to weighted Euclidean distance. The centroid $\overline{y}$ is median vector of $\{x_i\}_{i=1}^{L}$

IDOL

- The spectral distortion measures introduced in previous chapter are not straight forward as L_p norm.
- Below we will elaborate centroid solution for popular spectral distortion measure

8.2.2.1 Likelihood Distortion

- Likelihood distortion includes Itakura-Saito, Itakura and Likelihood distortion from chapter 6
- Let $\{s_i(w)\}_{i=1}^{L}$ be the spectra for which centroid has to be found

- Itakura-Saito distortion

  - The average Itakura-Saito distortion from each of the given spectral to an arbitrary p^{th} order all-pole model spectrum $\sigma^2/A(e^{jw})/^2$ is

$$D_{IS} = \frac{1}{L}\sum_{i=1}^{L} d_{IS}\left(S_i, \frac{\sigma^2}{|A|^2}\right) = \frac{1}{\sigma^2}\int_{-\pi}^{\pi}\left[\frac{1}{L}\sum_{i=1}^{L} S_i(\omega)\right]|A(e^{j\omega})|^2 \frac{d\omega}{2\pi} - \frac{1}{L}\sum_{i=1}^{L}\log \sigma_{\infty,i}^2 + \log \sigma^2 - 1$$

  Where σ_∞^2 is the one-step prediction error of $S_i(w)$.

$$\frac{1}{\sigma^2}\int_{-\pi}^{\pi}\left[\frac{1}{L}\sum_{i=1}^{L} S_i(\omega)\right]|A(e^{j\omega})|^2 \frac{d\omega}{2\pi} = \mathbf{a}^t\left[\frac{1}{L}\sum_{i=1}^{L}\mathbf{R}_i\right]\mathbf{a}/\sigma^2$$

  Where R_i is a (p+1) X (p+1) autocorrelation matrix, a is the coefficient vector of A(z) and t denotes the matrix transpose.

  - The centroid model spectrum has a filter coefficient vector $\bar{a}$ obtained as

$$\bar{\mathbf{a}} = \arg\min_{\mathbf{a}}\left(\mathbf{a}^t\left[\frac{1}{L}\sum_{i=1}^{L}\mathbf{R}_i\right]\mathbf{a}\right)$$

  - That is the solution for $\bar{a}$ is equivalent to solving LPC normal equations defined by the average autocorrelation $\sum R_i/L$
  - The gain term of the centroid model spectrum is

$$\bar{\sigma}^2 = \min_{\mathbf{a}} \left(\mathbf{a}^t \left[\frac{1}{L} \sum_{i=1}^{L} \mathbf{R}_i \right] \mathbf{a} \right)$$
$$= \bar{\mathbf{a}}^t \left[\frac{1}{L} \sum_{i=1}^{L} \mathbf{R}_i \right] \bar{\mathbf{a}}.$$

- The average distortion is minimized by the centroid model spectrum $\overline{\sigma^2}/|\bar{A}|^2$ with

$$(D_{\mathrm{IS}})_{\min} = \frac{1}{L} \sum_{i=1}^{L} d_{\mathrm{IS}} \left(S_i, \bar{\sigma}^2 / |\bar{A}|^2 \right)$$
$$= -\frac{1}{L} \sum_{i=1}^{L} \log \sigma_{\infty,i}^2 + \log \bar{\sigma}^2.$$

- <u>likelihood ratio distortion</u>

- The average likelihood ratio distortion from a set of p[th]order unity gain model spectra $\{1/A_i(e^{jw})|^2\}_{i=1}^{L}$ to an arbitrary unity gain all pole model spectrum $1/|A(e^{jw})|^2$ is defined by

$$D_{\mathrm{LR}} = \frac{1}{L} \sum_{i=1}^{L} d_{\mathrm{LR}} \left(\frac{1}{|A_i|^2}, \frac{1}{|A|^2} \right)$$
$$= \frac{1}{L} \sum_{i=1}^{L} \frac{\mathbf{a}^t \mathbf{R}_i \mathbf{a}}{\sigma_i^2} - 1$$

- Where R_i and a are defined as before and σ_i^2 is the minimum p[th]order residual energy associated with R_i
- The evaluation of the centroid model spectrum $1/|\bar{A}|^2$ becomes finding LPC solution to the set of normal equation defined by the residual-normalized autocorrelation

$$\bar{\mathbf{a}} = \arg\min_{\mathbf{a}} \left(\frac{1}{L} \sum_{i=1}^{L} \frac{\mathbf{a}^t \mathbf{R}_i \mathbf{a}}{\sigma_i^2} \right)$$

And

$$(D_{\mathrm{LR}})_{\min} = \bar{\mathbf{a}}^t \left[\frac{1}{L} \sum_{i=1}^{L} \frac{\mathbf{R}_i}{\sigma_i^2} \right] \bar{\mathbf{a}} - 1.$$

- <u>Itakura distortion</u>

- The centroid solution for the Itakura distortion is difficult to find

- The average distortion in this case is defined by

$$D_1 = \frac{1}{L}\sum_{i=1}^{L} d_1(S_i,\ \alpha_i/|A|^2)$$
$$= \frac{1}{L}\sum_{i=1}^{L} \log\left[\frac{1}{\sigma_i^2}\int_{-\pi}^{\pi} S_i(\omega)\left|A(e^{j\omega})\right|^2 \frac{d\omega}{2\pi}\right]$$
$$= \frac{1}{L}\sum_{i=1}^{L} \log\left(\frac{\alpha_i}{\sigma_i^2}\right)$$

  Where

$$\alpha_i = \int_{-\pi}^{\pi} S_i(\omega)\left|A(e^{j\omega})\right|^2 \frac{d\omega}{2\pi}$$

  And σ_i^2 is the minimum p[th]order residual energy of $S_i(w)$

8.2.2.2 COSH Distortion

- The COSH Distortion is defined as

$$d_{\text{COSH}}(S_i, S) = \frac{1}{2}\int_{-\pi}^{\pi}\left[\frac{S_i(\omega)}{S(\omega)} + \frac{S(\omega)}{S_i(\omega)}\right]\frac{d\omega}{2\pi} - 1.$$

- The averageCOSH distortion is

$$D_{\text{COSH}} = \frac{1}{L}\sum_{i=1}^{L} d_{\text{COSH}}(S_i, S)$$
$$= \frac{1}{2}\int_{-\pi}^{\pi}\left[\frac{\frac{1}{L}\sum_{i=1}^{L} S_i(\omega)}{S(\omega)} + \frac{1}{L}\sum_{i=1}^{L}\frac{S(\omega)}{S_i(\omega)}\right]\frac{d\omega}{2\pi} - 1.$$

- By taking derivative of D_{COSH} with respect to S(w) and setting it to zero we obtain

$$\frac{dD_{\text{COSH}}}{dS(\omega)} = \frac{1}{2}\int_{-\pi}^{\pi}\left[-\frac{\frac{1}{L}\sum_{i=1}^{L} S_i(\omega)}{S^2(\omega)} + \frac{1}{L}\sum_{i=1}^{L}\frac{1}{S_i(\omega)}\right]\frac{d\omega}{2\pi} = 0,$$

  Which has a solution at? $\bar{s}(w)$

$$\bar{S}(\omega) = \left\{\left[\frac{1}{L}\sum_{i=1}^{L} S_i(\omega)\right] \Big/ \left[\frac{1}{L}\sum_{i=1}^{L} S_i^{-1}(\omega)\right]\right\}^{1/2}.$$

- S(w) is power spectrum hence we retain with the positive solution.
- This is the centroid solution for the COSH distortion case with

$$(D_{\text{COSH}})_{\min} = \int_{-\pi}^{\pi} \left\{ \left[\frac{1}{L} \sum_{i=1}^{L} S_i(\omega) \right] \left[\frac{1}{L} \sum_{i=1}^{L} S_i^{-1}(\omega) \right] \right\}^{1/2} \frac{d\omega}{2\pi} - 1.$$

8.2.2.3 Cepstral Distance

- The centroid spectrum for untruncatedcepstral distance has a particular frequency domain interpretation that is important
- The cepstral distance is defined by

$$d_2^2(S_i, S) = \int_{-\pi}^{\pi} |\log S_i(\omega) - \log S(\omega)|^2 \frac{d\omega}{2\pi}.$$

- The average cepstral distance is

$$D_2 = \frac{1}{L} \sum_{i=1}^{L} d_2^2(S_i, S)$$
$$= \int_{-\pi}^{\pi} \frac{1}{L} \sum_{i=1}^{L} |\log S_i(\omega) - \log S(\omega)|^2 \frac{d\omega}{2\pi}$$

- Which is minimized by $\bar{s}(w)$

$$\bar{S}(\omega) = \left[\prod_{i=1}^{L} S_i(\omega) \right]^{1/L}$$

8.2.3 Vector Quantizer With Memory

Memoryless vector quatizer are effective fot the limited duration utterence like words

When words are long in duration the simple memoryless vector quatizer is not adequate

So the remedy for this is to use the vector quantizer with memory.

Matrix quantizer

- Encodes the several vectors, it is the extension memoryless vector quantizer
- It can be designed using Lloyd algorithm
- If n spectra encoded at same time then codebook C is designed to minimize

$$D = \frac{1}{T - n + 1} \sum_{t=1}^{T-n+1} d'(\mathbf{X}_t, \hat{\mathbf{X}}_t)$$

IDOL

- Where $x_t = (x_t, x_{t+1}, \ldots, x_{t+n-1})$ which is sequence of spectral vectors and
- The encoding sequence of spctral vector impiles that the cord word Y_i have embeded block memory constraints.
- The iterative procedure of Lloyd is readily applicable to matrix quantizer.
- By eq 5.2.1(a) the difference with memoryless vector quantizer is, it now involves finding of n separate spectral vector

Trellis vector

- Is another import class vector quatizer with memory
- It is a finite state vector quantizer
- It is specified by
    - Finite state spece Q
    - Initial state q_0
    - Three function
        1. Encoder :- $\alpha : A \times Q \rightarrow N$
           Where A : the space of spectral observation
           N : index set
        2. Transition :- $f : Q \times N \rightarrow Q$
        3. Decoder :- $\beta : Q \times N \rightarrow \hat{A}$
           Where $\hat{A}$: space of reproduction spectral vector

- During encoding, the encoder α assigns the input $x_t \in A$ to code word with index $u_t \in N$ based on current state q; i.e. $u_t = \alpha(x_t, q_t)$
- The state advances according to $q_{t+1} = f(q_t, u_t)$
- The decoder β upon receiving u_t reconstructs x_t by $\hat{x}_t$ based on the function $\hat{x}_t = \beta(q_t, u_t)$

8.2.4 Segmental vector quantization

- The standard memoryless quantization approach that uses a single vector quantizer for the entire duration of the utterance class for each class is not designed to preserve the sequential characteristic of the utterance class
- This lack of explicit characterization of the sequential behaviour can be remedied by treating each utterance class as concatenation of several information subsources each of which is represented by a VQ codebook
- We call this segment-specific VQ approach “Segmental vector quantization”

IDOL

- For an utterance $\{xt\}_{t=1}^{T}$ the simplest way to decompose it into a concatenation of N_s information subsources is to equally divide the utterance into N_s-segments$\{xt\}_{t=1}^{T/N_s}$,$\{xt\}_{t/N_{s+1}}^{T/N_s}$ and so on.
- The simple linear segmentation scheme is shown below

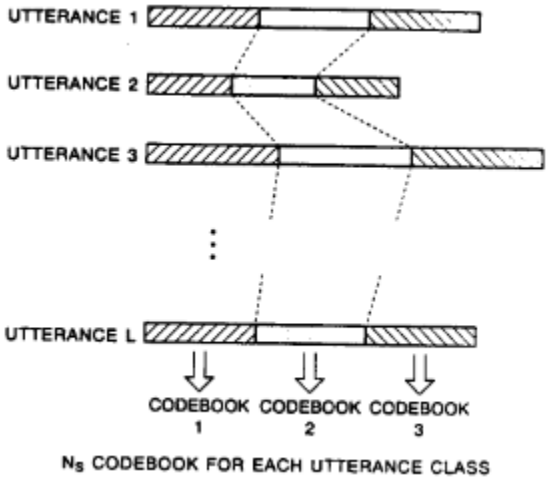


Figure 8.2 codebook training for segmental vector quantization[1]

- Segmental vector quatization requires the same computational complexity as the previous utterance based VQ
- The complexity increases is in the codebook storage

8.3 TEMPLATE TRAINING METHOD

Training problem is refer to as creating a reference pattern for speech recognition. Here we will discuss how to create a refence patterns.

8.3.1 Casual Training

- When there is no large vocabulary of utternace class and the system is designed for specific person, a simple template training procedure is used which is called casual training
- Each utterance class i.e. words, pharases etc. is represented by multiplicity of token words hence a multiplied of reference patterns
- Speaker should produce a consistent set of reference pattern to be the system useful

- Pros of casual training

- Simple training procedure

- Cons of casual training
  - System does not predict the pattern variability hence the system could easily fail when the words in vocabulary are confusing
  - The errors committed while training such as mispronunciation, mishandling of handset , acoustic noise or other are considered as valid reference patterns without any correction this leads to poor performance under some condition

8.3.2 Robust Training

- Each utterance class is spoken multiply until consistent pair of token is obtained.
- Resulting reference pattern is calculating as average of pair of consistent token
- Woking of training procedure is as follows
  - Let,
  - $X_1 = (x_{11},x_{12},x_{13},\ldots,x_{1T1})$: first spoken token
  - $X_2 = (x_{21},x_{22},x_{23},\ldots,x_{2T2})$: another spoken token
  - Two patterns compared via Dynamic Time Wrapping process resulting in DTW distortion score $d(X_1,X_2)$

$$d(\mathcal{X}_1, \mathcal{X}_2) \triangleq d_{\phi}(\mathcal{X}_1, \mathcal{X}_2) = \min_{\phi'} d_{\phi'}(\mathcal{X}_1, \mathcal{X}_2)$$

Where

$$d_{\phi'}(\mathcal{X}_1, \mathcal{X}_2) \triangleq \sum_{k=1}^{T_y} d(\phi_1'(k), \phi_2'(k)) m(k)/M_{\phi'}$$

Set of wrapping function $i_1 = \phi_1'(k)$ and $i_2 = \phi_2'(k)$

  - If $d(X_1,X_2)$ is smaller than prescribed threshold ε, the pair of token consider to be consistent
  - The reference pattern $Y=(y_1,y_2,\ldots,y_{ty})$ is then computed as a warped average of X_1 and X_2 as

$$\mathbf{y}_k = \frac{1}{2}(\mathbf{x}_{1\phi_1(k)} + \mathbf{x}_{2\phi_2(k)}), \qquad k = 1, 2, \ldots, T_y.$$

  - If $d(X_1,X_2) > \varepsilon$ talker asked to speak another training token X_3
  - Distortion score $d(X_1,X_3)$ and $d(X_2,X_3)$ are computed and compared against ε
  - The smaller of two scores falls within ε, a consistent pair is declared and reference pattern Y is calculated
  - Otherwise procedure repeats with another new training token until consistent pair of token obtained

8.3.3 Clustering

- High word recognition accuracy is achieved by the template training by clustering for speaker independent speech recognition
- Here L speech patterns each of which is a realization of particular utterance class is to be recognized.
- The task is to cluster (grouping the similar things) the L patterns into N clusters such that within each cluster the pattern are highly similar under the specific pattern dissimilarity measure, hence can be represented by typical template
- Here N templates are created from set of L training patterns for each utterance class

- Let Ω be a set of L training patterns, $\Omega = \{X_1, X_2, \ldots, X_L\}$ where each pattern X_i is a realization of one specific utterance class
- An LxL dissimilarity or distance matrix D can be defined with ij^{th} entry, d_{ij} calculated as

$$d_{ij} = \frac{1}{2}[d(\mathcal{X}_i, \mathcal{X}_j) + d(\mathcal{X}_j, \mathcal{X}_i)] = \delta(\mathcal{X}_i, \mathcal{X}_j).$$

- Because of the symmetry we need to save only L(L-1)/2 terms of $d_{ij.}$ This distance matrix is the basis of several clustering algorithm.
- The objective of algorithm is to cluster the training set Ω into N disjoiint cluster $\{w_i, i=1,2,\ldots,N\}$ such that

$$\Omega = \bigcup_{i=1}^{N} \omega_i$$

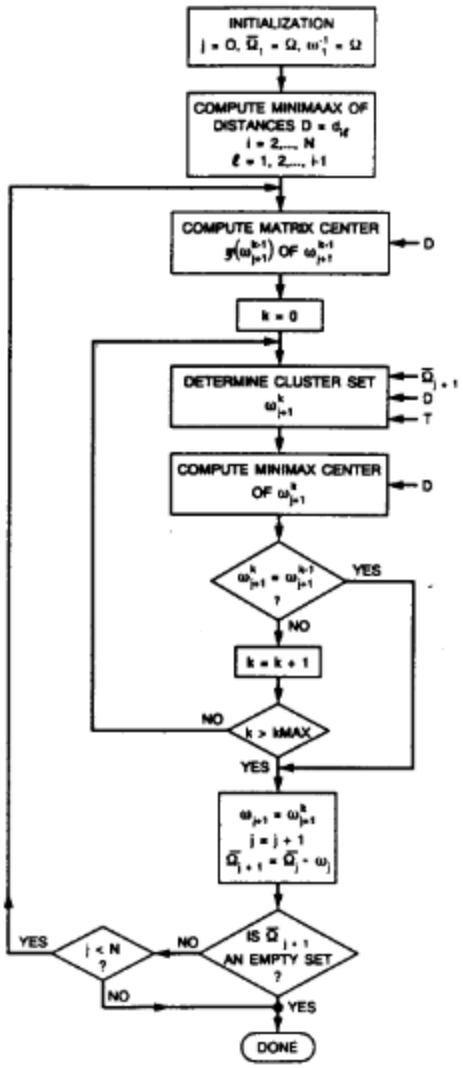
- And such that speech patterns in the same cluster are “close” to each other.

8.3.3.1 Unsupervised clustering without Averaging (UWA)

- The idea behind the supervise clustering is ‘find all training pattern close to the “center” of this cluster exclude this pattern from training set and recluster the remaining pattern’.

Diagram

- Flow diagram of UWA algorithm is shown below



- Let us denote,
  - Ω_j : as partial coveage set which incudes all trianing patterns in first j cluster

$$\Omega_j = \bigcup_{i=1}^{j} \omega_i = \Omega_{j-1} + \omega_j.$$

  - $\overline{\Omega_j}$: the complement set

$$\overline{\Omega_j} = \Omega - \Omega_j$$

  - Thus set of all remaining training patterns after j^{th} cluster is formed

    k : as iterative index

    w_j^k: as the set of training patterns in the j^{th}cluster at k^{th}iteration

  - Each cluster w, we have minimax “center” Y(w)

$$y(w) = X_{i_c} \in w$$

Such that

$$\max_m d_{i_c,m} = \min_i \max_m d_{i,m}$$

For all $X_i \in w$

Algorithm

- The UWA algorithm can be stated as follows

1. Initialization :
   j=0,k=0,$\bar{\Omega}_i = \Omega, w_1^{-1} = \Omega$
2. Determine Y(w_{j+1}^k) according to above equation by making use of distortion matrix
3. Form w_{j+1}^k by

$$\omega_{j+1}^k = \bigcup_m \mathcal{X}_m$$

   Where

$$\mathcal{X}_m \in \bar{\Omega}_{j+1}$$

   And

$$\delta\left(\mathcal{Y}(\omega_{j+1}^{k-1}), \mathcal{X}_m\right) \leq \delta_{th},$$

   Including within w_{j+1}^k all patterns $\bar{\Omega}_{j+1}$ that within distance threshold of minimax center Y(w_{j+1}^k)
4. Determine Y(w_{j+1}^k) the new minimax center of cluster w_{j+1}^k
5. If $w_{j+1}^k = w_{j+1}^{k-1}$ convergence is obtained and $w_{j+1} \triangleq w_{j+1}^k$, increment j, and form new partial training set ,$\bar{\Omega}_{j+1}$

$$\bar{\Omega}_{j+1} = \bar{\Omega}_j - w_j$$

   If $\bar{\Omega}_{j+1}$ is not an empty set and j is smaller than maximum number of cluster allowed then go back to step 2 and iterate else stop

In state 5 above, when the iteration index k reaches the maximum allowable numberof iteration, it is treated as if convergence were obtained

- Problems associated with UWA are

IDOL

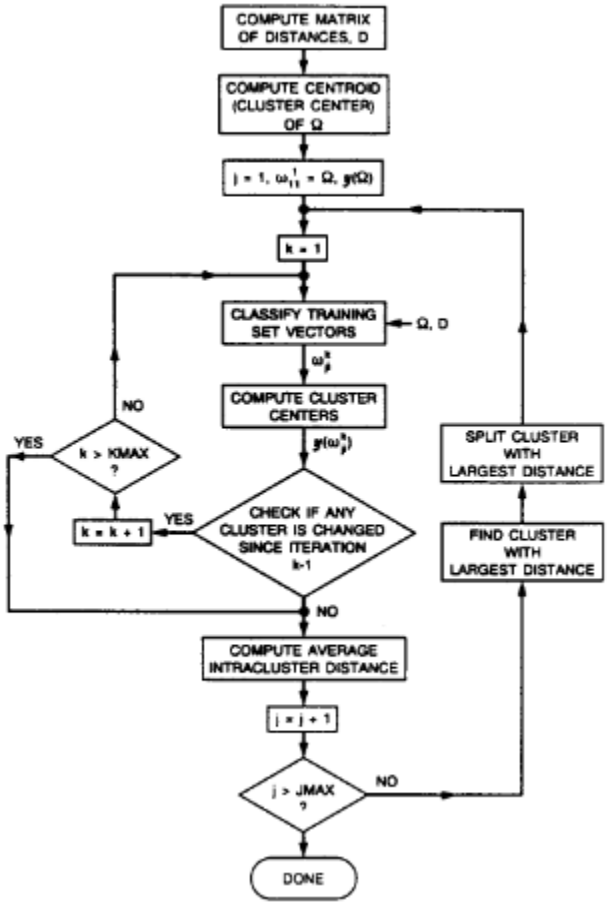
1. In step 3 user have to define the distance threshold, it is not defined by the analytical performance. Different utterance classes spoken by different talker require different threshold
2. Procedure does not guarantee coverage of the entire training set.

8.3.3.2 Modified K-means algorithm

- Here iteratively refining the clusters and cluster centroid such that some optimality criterion is met
- The difference between the Lloyd and MKM is that MKM deals with temporal sequence of spectral vectors rather than single vector

Diagram

- Below is the flow diagram of MKM algorithm



- We denote

- i^{th} cluster of a j^{th} cluster set at the k^{th} iteration as where i = 1, 2,...,j. and k = 1,.., $k_{max}$with , k_{max} being maximum allowable iteration count
- Y(w) is the representative pattern for cluster w and defined as the centroid.
- The algorithm finds the j clusters incrementally from j=1 to $j=j_{max}$ where j_{max} is maximum number of clusters

Algorithm

- Algorithm is as follows to compute the distance matrix D
    1. Initialization :

        Set j = 1, k = 1, i = 1; set$w_{1,1}^{1}$= Ω and compute the cluster centery(Ω) of Ω, the entire training set.

    2. Optimal classification :

        Label each pattern X_l, l = 1,2,…,L in Ω by index i according to minimum distance principle :

        $$\mathcal{X}_\ell \in \omega_{j,i}^k \quad \text{iff} \quad \delta(\mathcal{X}_\ell, \mathcal{Y}(\omega_{j,i}^k)) = \min_{i'} \delta(\mathcal{X}_\ell, \mathcal{Y}(\omega_{j,i'}^k)).$$

        Accumulate the total intracluster distance for each cluster $w_{j,i}^{k}$ defined by

        $$\Delta_i^k = \sum \delta(\mathcal{X}_\ell, \mathcal{Y}(\omega_{j,i}^k))$$

        Where the summation is overall $x_l \in w_{j,i}^{k}$

    3. Revision of cluster and cluster center :

        find new cluster center$w_{j,i}^{k+1}$, i=1,2,…,j

    4. Convergence check :

        Goto step 5 if following constion met

        i. $w_{j,i}^{k+1} = w_{j,i}^{k}$ for all i=1,2,…,j
        ii. $k = k_{max}$ a preset maximum allowable number of iteration
        iii. The change in average distance

IDOL

$$\left(\sum_{i=1}^{j}\Delta_i^k - \sum_{i=1}^{j}\Delta_i^{k-1}\right)\Big/\sum_{i=1}^{j}\Delta_i^{k-1} < \Delta_{th}.$$

Otherwise increment k ➔k+1and repeat step 2 through 4

5. Record the j-cluster solution:

   When convergence condition is met, the resultant clusters and cluster center$w_{j,i}^{k+1}$ and y($w_{j,i}^{k+1}$), i=1,2,…,j are the j-cluster solution for training set Ω.

   If j = j_{max}, j_{max} being the maximum number of cluster and the entire cluster processing complete else continue to step 6

6. Cluster splitting :

   Split the cluster that has the largest intracluster distance into two.

   There are two possibilities

   i. Based on largest total intracluster distance
   ii. Largest average intracluster distance

   Increment j ➔ j+1, reset k=1 and repeat steps 2-5

- The procedure guarantees the coverage of the training set Ω and set of references template ranging from 1 to j_{max} can be created

Now we discuss the computation of a cluster center

Minimaxcenter

- One definition of cluster center is theMinimaxcenter used in the UWA algorithm.
- It is member pattern of the cluster set
- The minimaxcenter can be easily computed by looking up the distance values in the matrix directly

*Pseudoaverage*center

- Another way of defining cluster to is find the particular pattern in cluster that has the largest population of patterns whose distance to particular pattern falls within threshold.

IDOL

- If several patterns have the largest count of patterns then the pattern that has the smallest distance to all pattern in subcluster is chosen as cluster center. This pattern is called *pseudoaverage*center

Average cluster center

- Another possibility is to perform certain *averaging* patterns in cluster.
- This is done after minimaxcenter or the pseudoaveragecenter.
- Since the patterns have temporal variations the averaging process involves the time alignment for robust training
- Those vectors are aligned to the same index i are then averaged tp produce an average spectrum. The resultant pattern sequence with vectors indexed from 1 to T_y is then desired *average* cluster center.

Above three types of cluster are not specifically designed to minimize the average intracluster distance because temporal variation inherent in the patterns and it is not straight forward to find pattern that minimizes intracluster distance.

References and further Reading

[1] L.R. Rabiner , B. H. Juang , "Fundamentals of Speech Recognition", Prentice-Hall, Englewood Cliffs, N.J.

IDOL

CHAPTER 9

SPEECH RECOGNITION SYSTEM DESIGN AND IMPLEMENTATION ISSUES PART - B

9.1 PERFORMANCE ANALYSIS AND RECOGNITION ENHANCEMENT

The performance of the system is depend upon the several factors such are choice of distortion measure, the way reference pattern created, the analysis condition

Choice of distortion measure

The several studies[4] has been carried out to point out several important issues in defining distortion measure.

1. The incorporation of energy in pattern comparison requires careful design. Like Itakura-saito distortion measure
2. It is important to design a distortion measure that takes into account the inherent speaker variation and works well for different speakers
3. It is not easy to incorporate our empirical knowledge of sound perception in distortion measure. The weighted slope measure and the weighted likelihood measure both failed to improve performance but slope measure works best when unweighted

Choice of clustering Methods and kNN decision Rule

- Template training by clustering is important because the acoustic variability of word utterance across different talkers that require multiple representation for an utterance class
- Because multiple reference templates we consider use of kNN decision rule to provide both robust class decision and improved recognition performance
- If we use N reference pattern to represent each word then for every unknown input pattern N distortion scores are calculated.
- We denote N distortion scores by d^{ij} , j = 1,2,…,N where I indicates the word index ranging from 1 to M the vocabulary size
- The distortion scores for particular word i can be recorded such that

$$d^{i(1)} \leq d^{i(2)} \leq \ldots. \leq d^{i(N)}$$

where parenthesis are used to indicate ordered indices.

- For kNN rule we compute the average of the k smallest distortion

$$d^i = \frac{1}{k}\sum_{j=1}^{k} d^{i(j)}.$$

- The index of recognized word using kNN rule is determined as

$$i^* = \arg\min_i d^i.$$

- The UWA algorithm was found to perform worse than MKM algorithm
- If observed for one template per word then MKM template yielded a 10% lower error rate than UWA template
- If more template per word is used then MKM template provides 2% reduction in error rate over obtained using UWA templates
- MKM is more analytically consistent than UWA.

Incorporation of Energy Information

- It is generally energy contour of an utterance contains important information about phonetic identity of sound within utterance
- Fricatives are much lower in energy than vowels
- Proper use of such energy information can therefore be helpful in distinguishing one word utterance from another
- The energy information appears in two forms

1. Absolute power of the power spectrum S(w)

$$E = \int_{-\pi}^{\pi} S(\omega)\frac{d\omega}{2\pi},$$

2. Gain σ^2 when S(w) is modelled by an all-pole spectrum $\sigma^2/|A(e^{jw})|^2$

$$\sigma^2 = \min_A \int_{-\pi}^{\pi} |A(e^{j\omega})|^2 S(\omega)\frac{d\omega}{2\pi}.$$

- The Itakura-Saito distortion measure includes the gain term in fixed and inflexible manner
- The way to add energy information i.e. absolute power of the gain is to add an energy distance term to any of the spectral distortion

IDOL

- The energy information is the temporal pattern sequence can be useful for improving the recognition accuracy if it is properly normalized

Effects of signal Analysis Parameters

- The analysis parameter using template-based system using LPC analysis are

1. LPC analysis order denoted by p
2. Length of analysis window denoted by N_w
3. Frame rate in terms of shift interval N_s

- The experiment with 4 talker 39 word vocabulary database was conducted in speaker trained recognition mode
- Five tokens per word is used in training and 10 tokens per word is used in testing
- The speech material was recorded over a dialed-up telephone line and sampled at 6.67 kHz.
- The values of p , N_w and N_s tested were p =6,8,10,12 ; N_w = 33,67,100,133,200,300 and N_s = 33,67,100,200,300 resp.
- The reference patterns generated by clustering method
- The performance at $N_w = 3N_s$ is found best for a fixed N_s
- Larger the N_s meand less computation, When N_w goes beyond 300 that is analysis window longer than 45 ms, the analysis window encompasses nonstationary segments of speech, reducing precision of signal representation
- The reasonable choice is N_w=300 and N_s=100. For a 6.67 kHz sampling rate these are equivalent to 45 ms analysis window and 15 ms window shift resp.

Performance of Isolated word-recognition System

- The isolated word-recognition system usually perform well with proper choice system parameters, the distortion measure and the method to generate the reference patterns

TABLE 5.4. Performance of Isolated Word-Recognition Systems

	Vocabulary	Mode	Error Rate (%)	Comments
10	Digits	SI	0	400 Talkers
37	Dialer Words	SD	0	10 Talkers
39	Alphadigits	SD	4.5	4 Talkers
		MS	7.0	100 Talkers
54	Computer Terms	SI	3.5	20 Talkers
129	Airline Words	SI	2.9	20 Talkers
200	Japanese Cities	SD	2.7	1 Talker
1109	Basic English	SD	4.3	3 Talkers

IDOL

Figure [1]

- Below table shows the summarization of laboratory performance of a number of isolated word system evaluated on different vocabularies in different modes

  SI : speaker independent mode

  SD : speaker dependent mode

  MS : multispeaker mode in which training data and testing data are from same set of talkers

- In comment column the number of talker is listed
- The size of vocabulary does not affect the recognition performance
- These performance show that isolated word-recognition technologies are mature enough to be of practical use for many application

9.2 Template adaption to new talkers

- it is a technique that adapts the values of parameters of reference pattern to changes in input signal so that recognizer can properly deal with input signal that are somewhat different from training pattern.
- Effects that cause changes to input signals are background noise difference in transducer , new talker accents different from those used in training set.
- In speech recognition task as long as the sufficient data is available to train the speaker dependent template the speaker dependent system performs better than speaker independent system
- If data is limited to train the speaker dependent template the superiority of speaker dependent system over speaker independent system does not guaranteed because lack of reliability in the reference parameters.
- To improve performance of recognizer under these conditions is to make use of existing knowledge as represented in speaker independent template set. This technique is referred to as speaker adaption

- By the following way speaker adaption of reference patterns can be formulated as shown in following figure

IDOL

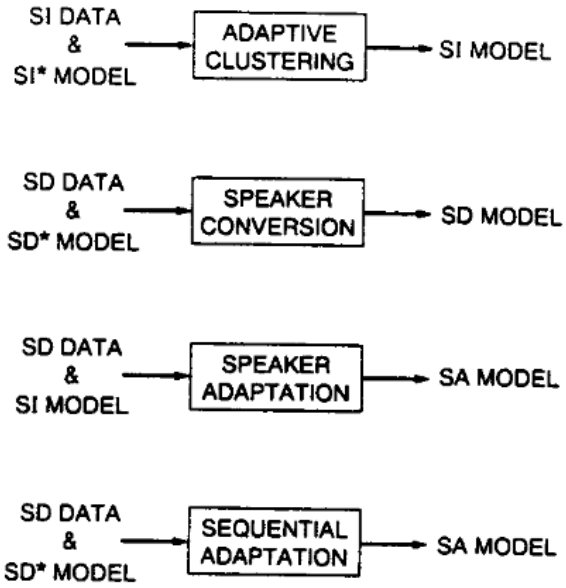


Fig:9.1 SA –SD model

- Adaptive clustering : modify the existing speaker independent reference set based on speaker independent training data so that new talkers characteristic are properly captured in adopted template set
- Speaker conversion : well trained template set for one particular talker is converted to another template set for new talker based on limited amount of training data from new talker
- Speaker adaption : modify the speaker independent template set using new training data from single talker and provide good set of speaker adapted templates

Sequential adaption : training data from a given talker are acquired over time and the reference patterns are sequentially adjusted every time new training data is available.

Spectral transformation

- Adaption of the reference set of templates involves the transformation of one spectrum to another that matches to new talkers spectral characteristic
- Suppose

  $S_A(w)$ speech produce by talker A

  f : Spectral transformation mapping converts $S_A(w)$ into $S_B(w)$ so,

  $S_B(w) = f(S_A(w))$

Which is likely to produced by talker B

- One approach to spectral transformation is based on physical vocal tract model
- Estimate the vocal cord spectrum and vocal cord dimension for individual talker and perform the spectral transformation defined by estimated articulatory parameter
- The transformation is paramerterized by Λ_{AB}.

  $S_B(w) = f(S_A(w), \Lambda_{AB})$

- Another approach is establish a correspondence between pairs of “typical” spectra from two talkers based on their occurrence in same context speech
- The context could be word as represented by a sequence of vector quantization codebook entries
- Use of VQ code words permits nonparametric mapping for a finite set of “typical” spectra.
- It offers a more flexible, effective mapping than universal parametric transformation $f(.\Lambda_{AB})$ because it does not rely on existence of such function for arbitrary spectrum
- The VQ based spectral transformation maps an arbitrary spectrum of one talker to another spectrum that is member of a finite collection of typical spectra of another talker.
- This type of transformation has proven effective in speech recognition

Hierarchical spectral clustering

- It is adaptive clustering technique performs speaker adaption in an automatic manner
- Speaker adaption is achieved by adapting code book entries from VQ codebook to a particular talker while keeping the index sequence intact
- The idea behind this is hierarchically cluster the spectra in new training set in correspondence with those in original VQ codebook
- The correspondence between centroid of new cluster and the original code word is established by way of deviation error

Hierarchical clustering method works as follows

1. Universal codebook is produced by clustering the spectral vectors collected from set of talkers. VQ based training clustering then used to create a word dictionary based on training utterance and the universal codebook
2. Another training set from new talker is given . the procedure starts by calculating centroid of all the training spectra from new talker and the centroid of all the N code spectra in universal codebook
3. We denote these two centroids as u_{11} and V_{11}.
4. Deviation error can be calculated as $z_{11} = U_{11}-v_{11}$
5. Next split the new training set into two clusters and calculate the centroid of each cluster
6. We denote these two centroids by U_{2i} , i=1,2.
7. Each of the shifted code word vectors y_i+z_{11}, i=1,2,…N is then assigned to one of two centroids according to nearest neighbour principle generating two new code word clusters and the corresponding centroids we denote it by v_{2i}, i=1,2.
8. We define two deviation vectors $z_{2i}=u_{2i}-v_{2i,}$ i=1,2 and use them to further shift the already shift code word vectors according to following procedure
9. Shift procedure for the code-word vectors takes into account the relative distances from particular shifted code word vector to all the new centroids
10. Let us assume y_i has been shifted to y'_i m pairs of centroids (u_{mi},v_{mi}) i=1,2,…,m are created
11. The deviation vector calculated as $z_{mi} = u_{mi}-v_{mi}$, i=1,2,…,m
12. S_j is calculated as weighted sum of deviation vector z_{mi}

$$s_j = \left(\sum_{i=1}^{m} w_{ji}\, z_{mi}\right) \Big/ \left(\sum_{i=1}^{m} w_{ji}\right)$$

Where

$$w_{ji} = 1/[d(\mathbf{y}'_j, \mathbf{u}_{mi})]^{\alpha}.$$

- In above equation α is constant typically between 0.5 and 1 and d(.,.) is a Euclidean distance
- The shift vector s_j is then used to adjust y'_j

$$y'_j \longrightarrow y'_j + s_j$$

- Hierarchical procedure is progressive that the number of clusters grows from one to the desired limit.
- The hierarchical VQ clustering method relies on the spectral dissimilarity between the code words and new cluster centers and hence is essentially

independent of the contextual environment in which particular spectrum or code words appears

9.3 dicriminative methods in speech recognition

- In speech recogition phonetically similar words are source of recognition error
- The large vocabulary size contains no phonetically similar words whereas small vocabulary size contains many similar words
- Many phonetically similar sound can be differentiate on the basis of critical portion of the utterance
- A simple distortion score over entire utterance duration is not sufficient for critical parts of utterance
- The average distortion along the optimal warping path is define as

$$d(\mathcal{X}, \mathcal{Y}_i) = \frac{1}{T} \sum_{k=1}^{T} d(\phi_x(k), \phi_{y_i}(k))$$

- But if the critical region is short compared to length with entire utterance, the above equation of average distortion can be unreliable

discriminative weighting

- Using the global discriminative weighting in the calculation of pattern dissimilarity, the dissimilarity between two speech pattern can be modified so that critical region of the utterance receive proper emphasis for recognition
- Global discriminative weighting defined as

$$d(\mathcal{X}, \mathcal{Y}_i) = \frac{\sum_{k=1}^{T} w_i(k)\, d(\phi_x(k), \phi_{y_i}(k))}{\sum_{k=1}^{T} w_i(k)}$$

- Where $w_i(k)$ is a weighting function to be determine
- In global dissimilarity measure the local slope weighting is defined according the local constraint for all vocabulary words
- The discriminative weight $w_i(k)$ is a template specific and provide better recognition decision

discriminative template

- Alternative to use of discriminative analysis is design reference template.
- The parameter values in template are preweighted so that it is able discriminate difference with all other words
- Procedure allows to optimize the template parameter values for best discrimination among the acoustically similar words

- Discriminative weighting and discriminative template optimization are component of the Discriminative training
- We will discuss several key aspect of discriminative training

Determination of word equivalence classes

- Form a set of equivalence classes such that the words belonging to same class are similar to each other acoustically and phonetically.
- To form equivalence classes for words it is necessary to define "word-by-word" dissimilarity
- Two approaches to this problem are
  1. Phonetically based
  2. Acoustically based

1. Phonetically based

- Word by word dissimilarity is on the basis of phonetic description of words
- For this define the "phonetic distance" matrix in which elements are distances of phoneme pairs and denoted as $d_p(.,.)$.
- The way obtain the phoneme distances are
  - By the count of the number of phonetic feature that have to be changed to convert one phoneme into another
  - Manually segment the words according to their phonetic description and calculate the distortion directly from the phoneme segment
- We need to define the distance cost
  - For inserting a phoneme d_I
  - For deleting a phoneme d_D
- The total word-by-word dissimilarity is defined by a dynamic time-warp match between the word with,
  - Vertical step in warping path represent an insertion

- Horizontal step in warping path represent an deletion

➢ The following diagram part(a) represents the phonetically based procedure for digit word “eight” and letter “j” and Part(b) for the words “one” and “nine”

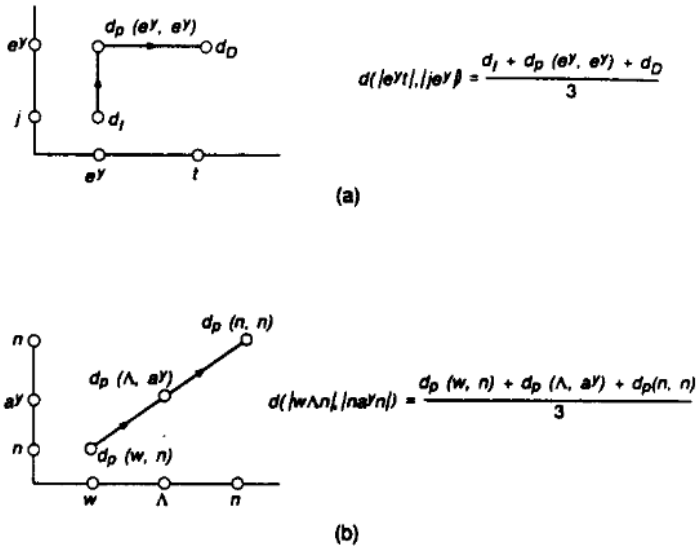


Figure 9.2 examples illustrating “word” alignment based on dynamic “phone” warping for word equivalence class [1]

- For words “eight” (/e^{y}t/) and ”J”(/je^y/) The alignment path is for an insertion of j, a match between the /e^y/ in eight and the /e^y/ in J and deletion of /t/.

$$d(/e^yt/, /je^y/) = \frac{d_I + d_p(e^y, e^y) + d_D}{3}$$
$$= \frac{d_I + d_D}{3}$$

Where we assume $d_p(e^y, e^y)=0$

- For words “one” (/w ∧ n/) and “nine” (/n a^y n/) the alignment path is giving

$$d(/w\ \Lambda\ n/,\ /na^y n/) = \frac{d_p(w,n) + d_p(\Lambda, a^y) + d_p(n,n)}{3}$$
$$= \frac{d_p(w,n) + d_p(\Lambda, a^y)}{3}$$

Where we assume $d_p(n,n)=0$

- The phonetic based approach provide a simple method of generating the equivalence class

2. Acoustically based

- To obtain word-by-word dissimilarity is to use real token of words and perform actual DTW on spectral patterns to calculate word distance
- Equivalence class can be obtained using clustering procedure.
- Here the vocabulary words are grouped into clusters based on pair wise distance
- Having obtained equivalence classes further define pattern-to-class distance for an unknown utterance pattern in order to provide decision that determines the equivalence set of unknown pattern before final recognition result is computed
- The pattern-to-class can be distance can be determined by :
- The minimum of the pattern-to-word distance for all the words in equivalence class ie for unknown pattern x

$$d(\mathcal{X}, \phi_j) = \min_{\mathcal{Y}_i \in \phi_j} d(\mathcal{X}, \mathcal{Y}_i)$$

Where $y_i = \phi_j$ means the word y_i which represents is in equivalence class ϕ_j

- Another way is apply k-nearest neighbour rule to all reference template of equivalence and use k-nearest neighbour average distance as pattern-to-class distance
- It does not lead the computational saving because pattern comparison is performed over entire set of reference template
- Another method is explicitly define the "Class-reference" template and calculate the distance from unknown pattern directly using class reference template
- Class reference template obtain using clustering method if there are multiple templates per class then k-nearest neighbour can be used
- A ranking based on $d(X,\phi_j)$, j=1,2,..M for each unknown pattern X can created for further discriminative processing

Number of distance calculation required for pattern-to-class distance is smaller than for word template, because number of word classes is smaller than number of words

Discriminative weighting function

- Discriminative weighing can be applied to maximize the likelihood of correct recognition after determining equivalence class of confusing word

- The question is how to choose the weighting function $w_i(k)$.
- Methods for designing a weighting function is proposed based on classical linear discriminant analysis, here we will see an approach to design the discriminative weights using the concept of Fisher's linear discriminant
- Let we have training set divided into two subset $\{x_j^{(+i)}\}$ and $x_j^{(-i)}$ with $N^{(+i)}$ and $N^{(-i)}$ utterance.
    - (+i) : corresponds to template y_i
    - (-i) : not corresponds to template y_i
    - We use vector notation

$$\mathbf{w}_i = (w_i(1), w_i(2), \ldots, w_i(T))^t$$

And

$$\mathbf{d}_{ij}^{(*)} = \left(d_{ij}^{(*)}(1), d_{ij}^{(*)}(2), \ldots, d_{ij}^{(*)}(T)\right)^t, \quad * = \pm i.$$

- When t as usual indicates matrix(vector) transpose and

$$d_{ij}^{(*)}(k) = d\left(\phi_{x_j}^{(*)}(k), \phi_{y_i}(k)\right), \quad k = 1, 2, \ldots, T, \ * = \pm i$$

With $\phi_{x_j}^{(*)}(k)$: the warping path

- $d_{ij}^{(+i)}$ is distance vector with elements obtained from matching two utterance of same word
- $d_{ij}^{(-i)}$ is distance vector with elements obtained from nonmatch utterance $x_j^{(-i)}$
- w_i has been modified so that the modified discriminative measure is

$$d(\mathcal{X}_j^{(+i)}, \mathcal{Y}_i) = \mathbf{w}_i^t \mathbf{d}_{ij}^{(+i)}$$

- Let us consider following sample means of the distance vector

$$\mu^{(+i)} \triangleq \left(\mu^{(+i)}(1), \mu^{(+i)}(2), \ldots, \mu^{(+i)}(T)\right)^t \triangleq \frac{1}{N^{(+i)}} \sum_{j=1}^{N^{(+i)}} \mathbf{d}_{ij}^{(+i)}$$

and

$$\mu^{(-i)} \triangleq \left(\mu^{(-i)}(1), \mu^{(-i)}(2), \ldots, \mu^{(-i)}(T)\right)^t = \frac{1}{N^{(-i)}} \sum_{j=1}^{N^{(-i)}} \mathbf{d}_{ij}^{(-i)}.$$

IDCL

After projection we have two mean values for two subsets of training data

$$\mu^{(+i)} = \mathbf{w}_i^t \mu^{(+i)} = \frac{1}{N^{(+i)}} \sum_{j=1}^{N^{(+i)}} \mathbf{w}_i^t \mathbf{d}_{ij}^{(+i)}$$

And

$$\mu^{(-i)} = \mathbf{w}_i^t \mu^{(-i)} = \frac{1}{N^{(-i)}} \sum_{j=1}^{N^{(-i)}} \mathbf{w}_i^t \mathbf{d}_{ij}^{(-i)}.$$

We define the scatter for projected distance vectors by

$$s^{(*)} = \sum_{j=1}^{N^{(*)}} \left(\mu^{(*)} - \mathbf{w}_i^t \mathbf{d}_{ij}^{(*)}\right)^2, \qquad * = \pm i.$$

The quantity $s^{(+i)} + s^{(-i)}$ is called within class scatter

- A measure of separation can be defined as

$$\theta(\mathbf{w}_i) = \frac{(\mu^{(+i)} - \mu^{(-i)})^2}{s^{(+i)} + s^{(-i)}}.$$

- In above equation the separation measure is expressed as a function of w_i

  And can be maximize by choosing

$$\mathbf{w}_i = \mathbf{S}^{-1}(-\mu^{(+i)} + \mu^{(-i)})$$

- The above equation is called Fisher Linear discriminant
- The variation of weighting function has the form

$$w_i(k) = \left|\mu^{(+i)}(k) - \mu^{(-i)}(k)\right| / \left[(\sigma^{(+i)}(k))^2 + (\sigma^{(-i)}(k))^2\right]^{1/2}$$

Discriminative training for Minimum recognition error

- The classical pattern classifier or recognizer is based on two fundamental steps
  - Definition of an appropriate discriminant function $g_i(x;\Lambda)$ parameterized on Λ for each class i

- Use of the discriminant function in implementing decision rule which stated as

$$C(\mathbf{x}) = C^i \quad \text{iff } g_i(\mathbf{x}; \Lambda) = \max_j g_j(\mathbf{x}; \Lambda),$$

Where C(x) denotes the recognition decision on x and c^i denotes class i

- When the decision is based on minimum distance the maximum in decision rule of above equation is replaced by minimum
- The goal of the discriminative training is to obtain the values in template parameter set ∧, for "best" recognition result
- generally speech recognizer are evaluated by their recognition error rate hence it is desirable to use recognition error as the minimization criterion
- The following three step procedure shows a resonable way to define the parametrically smoothed recognition error function
  1. Define set of discriminant functions

     $g_i(X,\Lambda)$, i= 1,2,…,M

  2. Define a misrecognition/misclassification measure

$$h_i(\mathcal{X}; \Lambda) = -g_i(\mathcal{X}, \Lambda) + \left\{ \frac{1}{M-1} \sum_{j, j \neq i} g_j(\mathcal{X}; \Lambda)^\eta \right\}^{1/\eta}$$

     Where n is the smoothing parameter

  3. Define a smooth 0-1 loss function $l_i(X;\Lambda)$ based on misrecognition measure i.e.

$$l_i(X; \Lambda) = l_i\big(h_i(X; \Lambda)\big)$$

     Example of smooth 0-1 function includes

     a. Sigmoid function

$$\ell(h) = \frac{1}{1 + e^{-\xi(h+\alpha)}}, \qquad \xi > 0$$

     b. Hypertangent function

$$\ell(h) = \tanh(h).$$

- The smooth 0-1 function then converts the enumerated recognition operation into smoothed error count
- The recognizer performance can be defined as the expected value of the above loss function

$$\mathcal{L}(\Lambda) = E\{; (X; \Lambda)\}$$

IDOL

- The performance criterion can be minimized by adaptive gradient descent scheme in which parameter set ∧ can be adjusted by a small amount
- Let $\wedge_{j+1}$ be the parameter set after X_j is applied. The adjustment role for obtaining $\wedge_{j+1}$

$$\wedge_{j+1} = \wedge_j + \Delta\wedge j$$

where

$$\Delta\wedge j = -\in_j \bar{U} \nabla l(X_j; \wedge_j)$$

$\in_j$ Where is a small positive number satisfying certain stochastic convergence constraint
$\bar{U}$ Is a positive definite matrix
ΔIs a gradient operator

- We need to make explicit definition involved in above three step procedure for discriminative training of DTW templates and weighting function
- The $l(X_j; \wedge)$ evaluates the error probability achieved by recognizer
- The gradient $\nabla l(X_j; \wedge)$determines the direction in which the recognizer parameter should be changed to increase or decrease this error probability
- The adaption amount $\nabla\wedge$ ensure that the error probability will decrease in probabilistic sense for the next token X_{j+1}
- As the more training data is presented the recognizer parameter set eventually converges to a solution which is at least a local optimum.

9.4 SPEECH RECOGNITION IN ADVERSE ENVIRONMENT

- The speech system which are designed under the low noise background degrades its performance in presence of noise and distortion
- Adverse environment means the differences in environment between training and testing.
- The environment while testing the system is mismatched with the environment condition while training the system.

IDOL

9.4.1 Adverse condition in speech recognition

1. Noise

   - Noise signals are additive. Means the speech signal consist of speech with addition of noise.
   - Example:
     - In office, noise includes office machinery such as typewriter, computer , printer etc. adds enough level of noise which degrades performance of the speech recognizer
     - In automobile noise from engine, wind, cooling fan, tyre and road etc.
     - Other noise is electronic noise which is present in any electronic speech recognition system.
     - Noise due to transmission and switching equipment in telephone network

2. Distortion

   - Speech signal undergoes in series of spectral distortion before being recorded and processed
   - The room in which speech recognizer system is deployed consist of some degree of distortion.
   - The microphone depending on its type and mounting position also can significantly distort the speech spectrum
   - When transducer configuration is used for training is different from used for testing then there is mismatch in spectral distortion which is a major problem.
   - It shows 85% of accuracy if matched transducer is used and less than 19% accuracy if different microphone is used during testing
   - If automatic speech recognizer is deployed in telephone network the telephone network through which speech signal travels can cause the distortion

3. Articulation effect

   - Many factors affect the speaking manner of individual talker.
   - Characteristic changes in articulation due to environmental influence is known as Lombard effect.
   - When talker speaks in environment with a noise that the first formant of vowel increases while second formant decreases.

IDOL

- These characteristic changes affect the performance of an automatic speech recognizer
- Speaker dependent word recognizer have accuracy better than 92% when training and testing done in clean environment but in when test utterance contained the Lombard effect then it achieve accuracy of 61%

Dealing with adverse condition

Signal Enhancement Preprocessing

- We can apply single enhancement method to supress the noise before applying recognition algorithm
- Adaptive noise cancellation using two signals sources is widely used signal enhancement method.
- It adaptively adjusts the set of filters so that subtracting the filtered noise from input leads to an output signal with minimum energy
- It has seen that in car if two microphones are placed at a distance greater than of 50 cm. the only noise component is engine noise, to cancel 90% of noise energy the two microphones cannot be placed more than 5cm
- Other signal enhancement method is separate noise reference
- These techniques use some estimate of the noise characteristics such as noise power or signal to noise ratio to obtain spectral models of speech from noise corrupted signals.

Special transducer Arrangements

- Noise cancelling microphone
- If talker position is fixed then noise cancelling microphone can be effective in supressing low frequency noise in automobile or cockpit
- It is specially designed dynamic microphone in which both side of diaphragm are exposed to sound fields
- So that the sound coming from large distance cancelled because sound pressure causes virtually no net force on the diaphragm

IDOL

- The sound source which is close to microphone, the back of diaphragm is shaded from sound field and sound pressure is perceived only by the front of the diaphragm.

Two sensor input

- Other type of microphone to be effective if the sensor location was fixed and optimized
- The noise cancellation microphone is not effective when used in cockpit instead of that suggested to use two sensor input that combines an accelerometer output for low frequency and gradient microphone output for high frequency .
- The accelerometer is attached to the skin of talker i.e. near to the throat

Noise masking and Adaptive model

- In broadband noise certain region of speech spectrum below the lower level will be affected by the noise. This makes spectral distortion calculation more difficult because corrupted region represent less reliable spectral measurement
- Noise masking in conjunction with a filter bank analyzer.
- Choose the masking noise level, for each channel of filter bank output, greater than noise level in reference signal and testing signal and then replace that channel output by the mask value if it is below the corresponding mask level.
- This helps to prevent distortion accumulation because the channels which are corrupted by noise will have the same spectral value in both training and testing
- Limitations of this is when two patterns being compared that have very different noise levels. When test token have a high level of noise, all the reference patterns that are of lower level than the noise would result in equally small distance making the comparison meaningless

A separate technique employed by Roe is to adapt the spectral prototypes to the noise condition in autocorrelation domain.

This technique is reminiscent of spectral transformation of VQ codebook

The assumption here is that the power spectra of speech and noise are additive and so are the autocorrelation

Stress compensation

- Purpose of stress compensation is to provide offset for the spectral distortion caused by extraordinary speaking effort due to the talkers reaction to ambient conditions
- Template based system that incorporate VQ codebook and Dynamic time wrapping
    - In this, spectral pattern was first vector quantized and replaced by closest code word in VQ codebook
    - Each stressed speech utterance was time aligned without spectral quantization to correct reference template
    - The autocorrelation vector of the stressed speech frames were then grouped according to the VQ indices in the reference template and average to yield the stress-compensated prototypes

- Cepstral domain
    - The spectral distortion induced by unusual speaking efforts could be compensated by simple linear transformation of the cepstrum
    - The statistics of cepstral vectors did display some systematic modification in various speaking styles hence possibility of spectral compensation in the cepstral domain become feasible

IDOL

References and further Reading

[1] L.R. Rabiner , B. H. Juang , "Fundamentals of Speech Recognition", Prentice-Hall, Englewood Cliffs, N.J.

CHAPTER 10

THEORY AND IMPLEMENTATION OF HIDDEN MARKOV MODELS PART-A

10.0 Objectives:

- Discrete time Markov processes
- Extensions to hidden Markov Models
- The three basic problems for HMMs

10.1 INTRODUCTION:

HMM is a statistical method of characterizing the spectral properties of frames.
The underlying assumption of the HMM is that the speech signal can be characterized as a parametric random process and that the parameters can be estimated in a precise and well defined manner.

The basic theory of HMM was published in a series of classic papers by Baum and his colleagues.

10.2 DISCRETE TIME MARKOV PROCESSES:

Consider a system that can be described at any time in one of a set of N distinct states indexes by {1, 2, N}.

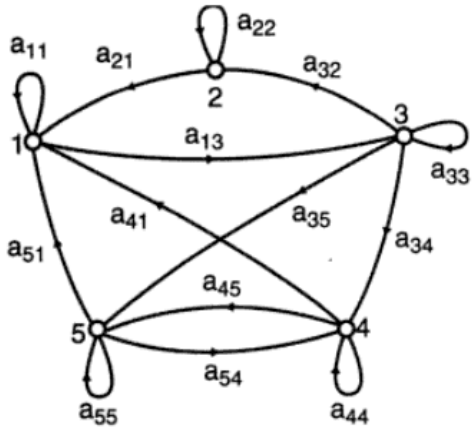


Figure 6.1 A Markov chain with five states (labeled 1 to 5) with selected state transitions.

Figure: 10.1 A Markov chain with five states (labeled 1 to 5) with selected state transition.

At regular or discrete times, the system undergoes a change of state according to a set of possibilities associated with states; it might possibly go back to the same state.
We denote the time instants associated with state changes as $t = 1, 2,$
Actual state at time t *as* q_t

There are 5 states named as 1, 2 … 5.To full describe this system requires, specification of current state at time *t* as well as all the predecessor states

The first order, markov chain, the probabilistic dependence is truncated to just the preceding state that is:
$P[qt = j \mid q_{t-1} = i , q_{t-2} = k, \ldots] = P[q_t = j \mid q_{t-1} = i]$

Furthermore, by considering only right hand side of above equation, it leads to set of state transition probabilities a_{ij} of the form

$$a_{ij} = P[q_t = j \mid q_{t-1} = i]$$

With following properties:

$$a_{ij} \geq 0 \qquad \forall j,i$$

$$\sum_{j=1}^{N} a_{ij} = 1 \qquad \forall i$$

The above stochastic (having a random probability distribution or pattern that may be analyzed statistically but may not be predicted precisely) process is called Observable Markov model. Because each state of process correspondence to an observable event, at each instant of time.

Let's consider an example of Discrete Markov Model:

The model shown in the figure below is a three state Markov Model of weather.

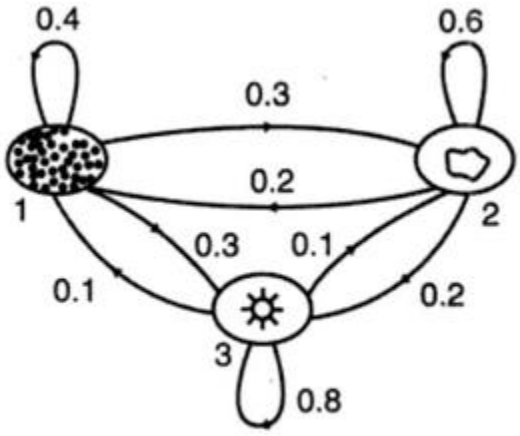


Figure 6.2 Markov model of the weather.

Figure: 10.2 Markov model of the weather

The model has three states:
State 1: Rain or Snow
State 2: Cloudy
State 3: Sunny

Let the weather on day *t* to be denoted by one of the states above.
The state transition probabilities matrix is as follows:

$$A = \{a_{ij}\} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}.$$

Let's solve problems on above weather model.

Problem 1:

What is the probability that the weather for eight consecutive days is "sunny-sunny-sunny-rainy-rainy-sunny-cloudy-sunny"?

Solution:

First define the observation sequence, O as

O = (sunny, sunny, sunny, rainy, rainy, sunny, cloudy, sunny)
= (3,3, 3, 1, 1, 3, 2, 3)
Day 1 2 3 4 5 6 7 8

Now we have to calculate the probability of the above observation sequence, P(O|Model) as:

$$
\begin{aligned}
P(\mathbf{O}|\text{Model}) &= P[3,3,3,1,1,3,2,3|\text{Model}] \\
&= P[3]P[3|3]^2P[1|3]P[1|1] \\
&\quad P[3|1]P[2|3]P[3|2] \\
&= \pi_3 \cdot (a_{33})^2 a_{31} a_{11} a_{13} a_{32} a_{23} \\
&= (1.0)(0.8)^2(0.1)(0.4)(0.3)(0.1)(0.2) \\
&= 1.536 \times 10^{-4}
\end{aligned}
$$

Where the initial state is denoted by

$$\pi_i = P[q_1 = i], \qquad 1 \le i \le N$$

Problem 2:
Given that the system is in a known state, what is the probability that it stays in that state for exactly d days?

Solution:
Here we have to calculate the probability of say state *i* stays consecutively for d days.

Hence the observation sequence would be:

O = (*I*, *I*, *I*,.... *I*, *j*) ..... j is a state other than i
Day = 1 2 3..... d d+1

Now let's calculate, P(O!Model) as

IDOL

$$P(\mathbf{O}|\text{Model}, q_1 = i) = P(\mathbf{O}, q_1 = i|\text{Model}) / P(q_1 = i)$$
$$= \pi_i (a_{ii})^{d-1}(1-a_{ii}) / \pi_i \qquad (6.5)$$
$$= (a_{ii})^{d-1}(1-a_{ii})$$
$$= p_i(d)$$

$P_i(d)$ is the probability distribution function of duration d in state *i.*

Based on $P_i(d)$, we can calculate the expected number of observations in a state, conditioned on starting in that state as

$$\bar{d}_i = \sum_{d=1}^{\infty} d p_i(d) \qquad (6.6a)$$

$$= \sum_{d=1}^{\infty} d(a_{ii})^{d-1}(1-a_{ii}) = \frac{1}{1-a_{ii}}. \qquad (6.6b)$$

Hence

P(O!Model) = $1/(1 - a_{ij})$

Let State *I* be 2 (sunny), then the expected number of days of sunny weather, is
P(O!Model) = $1/(1 - a_{ij})$

= 1/(1-0.8) = 1/0.2
= 5 days

Problem 3:
Derive the expression for the mean of $p_i(d)$.

Solution:

IDOL

$$\bar{d}_i = \sum_{d=1}^{\infty} d p_i(d)$$

$$= \sum_{d=1}^{\infty} d(a_{ii})^{d-1}(1-a_{ii})$$

$$= (1-a_{ii}) \frac{\partial}{\partial a_{ii}} \left[\sum_{d=1}^{\infty} a_{ii}^{d} \right]$$

$$= (1-a_{ii}) \frac{\partial}{\partial a_{ii}} \left(\frac{a_{ii}}{1-a_{ii}} \right)$$

$$= \frac{1}{1-a_{ii}}.$$

10.3 EXTENSION TO HIDDEN MARKOV MODELS

In the discrete markov model, each state corresponds to a deterministic observable event, hence the output is not random and it can be determined. That's why this model is restrictive and is not applicable to many problems.

To overcome the limitation of it, another approach to markov model is implemented, in which the observation of a state is a probabilistic function. And this extended approach is called Hidden Markov Model.

This is called hidden because the state of the model is not directly observable (i.e. hidden) but it can be observed only through another set of process which can be a sequence of observations.

Before going to some simple examples of Hidden Markov Model, let's first review some basic concepts of probability through following exercise:

Exercise:

Given a single fair coin i.e. P(Heads) = P(Tails) = 0.5, which you toss once and observe Tails,

Problem 1:
What is the probability that the next 10 tosses will provide the sequence (HHTHTTHTTH)?

IDOL

The probability of any specific observation of length 10 is $(1/2)^{10}$ since there are 2^{10} such sequence and all are equally probable.
Thus

$$P(HHTHTTHTTH) = \left(\frac{1}{2}\right)^{10}.$$

Problem 2:
What is the probability that the next 10 tosses will provide the sequence (HHHHHHHHHH)?

Since P(H)=P(T),
The specified run length of H for 10 is same as specified run of interlaced H and T.

$$P(HHHHHHHHHH) = \left(\frac{1}{2}\right)^{10}.$$

Problem 3:
What is the probability that 5 of the next 10 tosses will be tails? What is the expected number of tails over the next 10 tosses?

The probability of 5 tails in next 10 tosses is:

$$P(5H, 5T) = \binom{10}{5}\left(\frac{1}{2}\right)^{10} = \frac{252}{1024} \cong 0.25$$

The probability would be same for getting 5 of the next 10 tosses will be Heads, as both H and T have equal chances.

The expected number of tails in 10 tosses is:

$$E(T \text{ in 10 tosses}) = \sum_{d=0}^{10} d\binom{10}{d}\left(\frac{1}{2}\right)^{10} = 5.$$

Hence, on an average, there can be 5T and 5H in 10 tosses, but the probability of exactly 5T or 5H is only 0.25.

IEOL

10.4 COIN - TOSS MODELS:

Consider a scenario in which you are on one side of a curtain and on other side, there is a person performing a coin tossing experiment with one or more coins.

The person just provide the result of coin each coin toss, but which coin was tossed is not been told.

Thus a sequence of hidden coin tossing experiment is performed and the observation sequence is a series of Heads and Tails as:

$O = (o_1 o_2 o_3 \ldots\ldots o_r)$

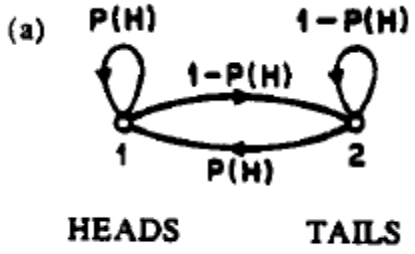
$= (H H T T T H T T H \ldots. H)$

Here in coin tossing experiment, we face following three problems:

1. What are the states in the model?
2. How many states should be in the model?
3. What are the state transition probabilities?

Consider following three cases to understand the HMM.

Case 1:
Here we consider that there is only a single biased coin is tossed. The model is depicted in the following figure:



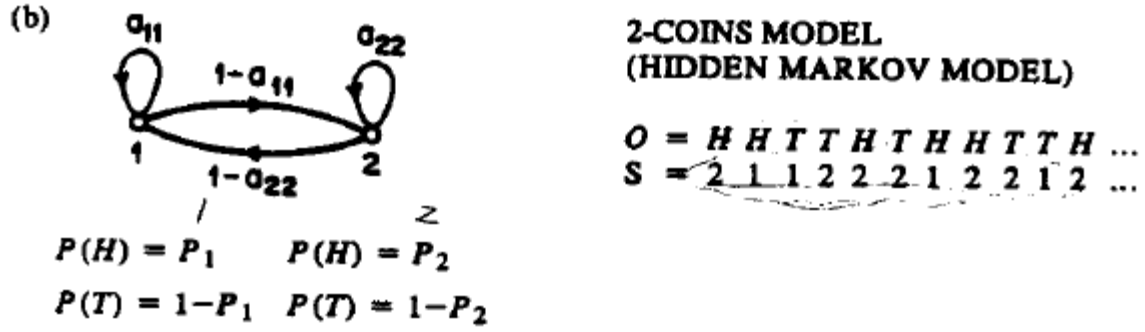
1-COIN MODEL
(OBSERVABLE MARKOV MODEL)

$O = H H T T H T H H T T H \ldots$
$S = 1 1 2 2 1 2 1 1 2 2 1 \ldots$

We could model the situation in two state model, where each state corresponds to previous coin toss.
This morkov model is observable and the only problem is how to decide the best value for the single parameter of the model, i.e.the probability of say Heads.

Case 2:

In this case, again we consider a model of two state. Each state corresponds to different biased coins tossed. Each state is characterized by probability distribution of Heads and Tails, and the transition between those states is characterized by a state transition matrix.
This model is depicted below:



Case 3:
In this model, we consider 3 different biased coins are tossed. The coin to toss is choosing based on some probability event.
This model is depicted below:

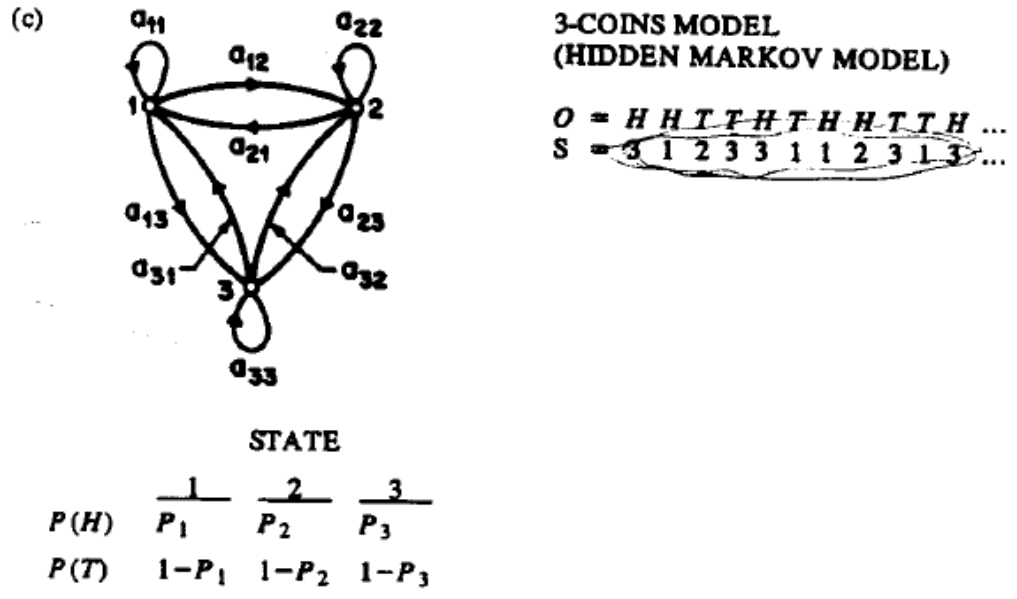


Figure 6.3 Three possible Markov models that can account for the results of hidden coin-tossing experiments. (a) one-coin model, (b) two-coins model, (c) three-coins model.

Figure: 10.3 Three possible Markov models

From the above three model, we can see that, case1 model has only single unknown parameter as it has only one coin hence (1^2). Case 2 model has 4 unknown parameters as it has two coins hence (2^2) and case 3 has 9 unknown parameters which uses 3 coins (3^2).

Thus the larger HMMs seem inherently more capable of modeling as series of coin-toss than smaller HMM model.

10.4 THE URN AND BALL MODEL:

After the coin toss model, now let us understand some more complicated situation to model through HMM.

Consider urn and ball system shown below:

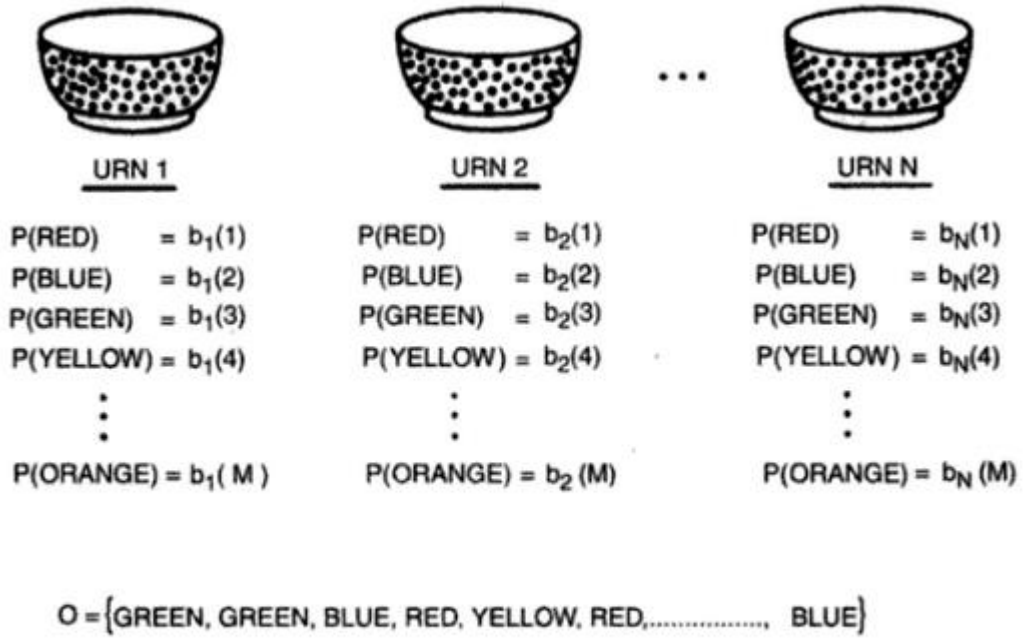


Figure 6.4 An *N*-state urn-and-ball model illustrating the general case of a discrete symbol HMM.

Figure 10.4: An N-state urn-and-ball model

Assume in a room there is N number of glass urns. Every urn is filled with large quantity of coloring balls. Let there be M number of distinct coloring balls.
Now we form the observation series as follows.

A machine chooses an initial urn according to some random procedure and then randomly chooses a ball from that initial urn. The color of the chosen ball is recorded as observation.

A new urn is selected by the machine according to some random procedure associated with the previous urn; the ball selection process is repeated.

This entire process is repeatedfinite number of times to generate an observation sequence which we like to model it as output of HMM.

It should be noted that the urns may have same colors of ball in it and so from a specific color ball does not tell us from which urn it has been chosen.

The simple HMM in urn and ball process is to consider each state corresponds to a specific earn, and the color of the ball is defined by some probability. As well the choice of urn is according to the state transition matrix.

10.5 ELEMENTS OF A HIDDEN MARKOV MODEL:

An HMM for discrete symbol observations is characterized by following:

1. **Number of states in the model, N:**

   Even though states are hidden, in practical applications there is some physical significance attached to the states.
   In coin toss model, each state represents a distinct biased coin.
   In urn and ball model, the states correspond to urns.
   Usually states are interconnected, so that state can transit from one to another.
   The individual states are labeled as {1, 2, 3,…. , N} and denote the state at time *t* as q_t.

2. **The number of distinct observation symbols per state, M:**
   M is the discrete alphabet size.
   This symbol corresponds to the physical output of system model.
   For coin toss model, observation symbol were Heads and Tails.
   For urns and ball model, they were the colors of balls.
   The individual symbols is labeled as V = { V_1, V_2, V_3, ….. V_M}.

3. **The state transition probability distribution A = {a_{ij}}** where

$$a_{ij} = P\left[q_{t+1} = j \middle| q_t = i\right], \qquad 1 \le i, j \le N. \tag{6.7}$$

   For a special case in which ant state can reach any other state in a single step, we have $a_{ij}>0$ for all *I, j.*

4. **The observation symbol probability distribution, B = {$b_{jj}(k)$}.**
   Here ,

$$b_j(k) = P\left[\mathbf{o}_t = \mathbf{v}_k \,\middle|\, q_t = j\right], \qquad 1 \le k \le M, \tag{6.8}$$

   Defines the symbol distribution in state j, j=1,2…. N.

5. **The initial state distribution** $\pi = \{\pi_i\}$
   Here,

$$\pi_i = P[q_1 = i], \qquad 1 \le i \le N. \tag{6.9}$$

From above parameters, to completely specify an HMM requires, specifying two model parameters N and M, then specifying observation symbols and then specifying the three sets of probability measures A, B and Pie(symbol).

Let

$$\lambda = (A, B, \pi) \tag{6.10}$$

This parameter set defines a probability measure for O i.e. P(O|lambda).

10.6 HMM GENERATOR OF OBSERVATION:

Provided the appropriate value of N, M, A, B and pie (symbol), the HMM can be used to generate the observation sequences.

$$\mathbf{O} = (\mathbf{o}_1\, \mathbf{o}_2 \ldots \mathbf{o}_T)$$

I.e.

T = Number of observation

Using the following procedure above observation sequence can be generated:

1. Select an initial state $q_i = i$ according to some initial state distribution pie(symbol)
2. Set $t = 1$.
3. Select $o_t = v_k$, according to some probability distribution in state *i.* i.e. $b_j(k)$
4. Transit to a new state $q_{i+1} = j$, according to some state transition probability distribution from the state*/* i.e. $a_{ij.}$
5. Set $t=t+1$;

Return to step 3 if $t < T$, otherwise terminate the procedure.

From the above procedures, generated states and the observations can be shown as follows in the table:

time, t	1	2	3	4	5	6	...	T
state	q_1	q_2	q_3	q_4	q_5	q_6	...	q_T
observation	o_1	o_2	o_3	o_4	o_5	o_6	...	o_T

Above procedure can be used to generate the observations model to simulate how a given observation sequence was generated by an appropriate HMM

Now let's exercise some numerical problems on HMM:

Consider a three state HM model of coin tossing experiment (three different coins) with following probabilities

	State 1	State 2	State 3
$P(H)$	0.5	0.75	0.25
$P(T)$	0.5	0.25	0.75

Problem 1:
Observation Sequence:

$$\mathbf{O} = (H\,H\,H\,H\,T\,H\,T\,T\,T\,T).$$

Here, what is the most likely state sequence? What is the probability of the above observation sequence and this most likely state sequence?

The most likely state sequence is the one with the highest probability. So p(H) is highest in state 2 and p(T) is highest in state 3,

So according to the given observation :

O= (H H H H T H T T T T)

= (2 2 2 2 3 2 3 3 3 3)

Hence most likely state sequence is q= (2 2 2 2 3 2 3 3 3 3)

Problem 2:

Probability of above observation sequence and the state sequence is

$$P(\mathbf{O}, \mathbf{q}|\lambda) = (0.75)^{10}\left(\frac{1}{3}\right)^{10}.$$

What is the probability that the observation sequence came entirely from state 1?

The probability that the above observation sequence from the state one itself i.e. with state sequence as q^ = (1 1 1 1 1 1 1 1 1 1)

Is

$$P(\mathbf{O}, \hat{\mathbf{q}}|\lambda) = (0.50)^{10}\left(\frac{1}{3}\right)^{10}.$$

The ratio of $P(\mathbf{O}, \mathbf{q}|\lambda)$ to $P(\mathbf{O}, \hat{\mathbf{q}}|\lambda)$ is:

$$R = \frac{P(\mathbf{O}, \mathbf{q}|\lambda)}{P(\mathbf{O}, \hat{\mathbf{q}}|\lambda)} = \left(\frac{3}{2}\right)^{10} = 57.67$$

It means q is more likely than q^

Problem 3:

Consider the following observation sequence:

$$\tilde{\mathbf{O}} = (\overset{2}{H}\overset{3}{T}\overset{3}{T}\overset{2}{H}\overset{3}{T}\overset{2}{H}\overset{2}{H}\overset{3}{T}\overset{3}{T}\overset{2}{H}).$$

How would your answers to parts a and b change?

For the above observation sequence, which has equal number of H and T, the part a and b would remain same, as the most likely state occurs same number of time in both cases.

Problem 4:

With following statetransition probabilities:

$$\begin{array}{lll} a_{11} = 0.9 & , \quad a_{21} = 0.45 & , \quad a_{31} = 0.45 \\ a_{12} = 0.05 & , \quad a_{22} = 0.1 & , \quad a_{32} = 0.45 \\ a_{13} = 0.05 & , \quad a_{23} = 0.45 & , \quad a_{33} = 0.1 \end{array}$$

A new model lambda^t, how would your answers parts 1-3 change? What does this suggest about the type of sequence generated by the models?

The new probability of O and q becomes:

$$P(\mathbf{O}, \mathbf{q}|\lambda') = (0.75)^{10} \left(\frac{1}{3}\right) (0.1)^6 (0.45)^3.$$

The new probability of O and $\hat{\mathbf{q}}$ becomes

$$P(\mathbf{O}, \hat{\mathbf{q}}|\lambda') = (0.50)^{10} \left(\frac{1}{3}\right) (0.9)^9.$$

The ratio is

$$R = \left(\frac{3}{2}\right)^{10} \left(\frac{1}{9}\right)^6 \left(\frac{1}{2}\right)^3 = 1.36 \times 10^{-5}.$$

The probability of O^ and q is not the same as the probability of O and q.

We have

$$P(\tilde{\mathbf{O}}, \mathbf{q}|\lambda') = \frac{1}{3} (0.1)^6 (0.45)^3 (0.25)^4 (0.75)^6$$

$$P(\tilde{\mathbf{O}}, \hat{\mathbf{q}}|\lambda') = (0.50)^{10} \left(\frac{1}{3}\right) (0.9)^9$$

We have

$$R = \left(\frac{1}{9}\right)^6 \left(\frac{1}{2}\right)^3 \left(\frac{1}{2}\right)^4 \left(\frac{3}{2}\right)^6 = 1.67 \times 10^{-7}.$$

Clearly, because $a_{11} = 0.9$, $\hat{\mathbf{q}}$ is more likely.

10.7 THE THREE BASIC PROBLEMS FOR HMMS

Given the form of HMM model, makes us to solve three problems for the model, which will be useful in real life applications
Those three problems are as follows:

Problem 1:
Provided,, the observation sequence, O ={ $o_1, o_{2,,} \ldots o_T$} and a model lambda(symbol) = (A, B, pie(symbol)).
How can we efficiently calculate the probability of the observation sequence in that model, i.e. P(O|lambda)?
Problem 2:
Provided the observation sequence O = { $o_1, o_{2,,} \ldots o_T$} and the model parameter lambda, how to select the best optimal state sequence q= ()
Problem 3:
How do we adjust the parameter lambda = (A, B , pie) to maximize P(O|lambda)?

Problem 1 is the evaluation problem., where we have to compute the probability
Or we can view this problem as to find out how a given model matches the given observation sequence. For example, if we try to find out the best model among several others, then the solution to problem one allows finding out the best match of the observation sequence.
Problem 2, here we uncover the hidden part i.e. the correct state sequence of the model. Solution to this would be to learn the structure of model and to find the optimal state sequence for speech recognition.
Problem 3 , here we optimize the model parameters to describe the model. The observation sequence used to adjust the model parameters is called training sequence because it is used to train the HMM.

Now let's see the mathematical solutions for each problem for HMMs:

Solution to Problem 1: Probability Evaluation

Here we calculate the probability of Observation sequence, O ={ $o_1, o_{2,,} \ldots o_T$} given the model lambda, P(O|lambda)
This can be done using enumerating every possible state sequences of length T.
Consider one such state sequence q = (q1 q2 q3,,, qT)
The probability of the observation sequence O is as follows:

$$P(\mathbf{O}|\mathbf{q}, \lambda) = \prod_{t=1}^{T} P(\mathbf{o}_t|q_t, \lambda) \qquad (6.13a)$$

Where we have assumed statistical independence of observations.
Thus we get,

$$P(\mathbf{O}|\mathbf{q},\lambda) = b_{q_1}(\mathbf{o}_1)\cdot b_{q_2}(\mathbf{o}_2)\cdots b_{q_T}(\mathbf{o}_T). \quad (6.13b)$$

The probability of such state sequence q can be written as

$$P(\mathbf{q}|\lambda) = \pi_{q_1} a_{q_1q_2} a_{q_2q_3} \cdots a_{q_{T-1}q_T}. \quad (6.14)$$

The joint probability of O and q occur simultaneously, is the product of the above two terms i.e.

$$P(\mathbf{O},\mathbf{q}|\lambda) = P(\mathbf{O}|\mathbf{q},\lambda)P(\mathbf{q}|\lambda). \quad (6.15)$$

Now by summing this joint probability for all possible state sequence q is giving

$$P(\mathbf{O}|\lambda) = \sum_{\text{all } \mathbf{q}} P(\mathbf{O}|\mathbf{q},\lambda)P(\mathbf{q}|\lambda) \quad (6.16)$$

$$= \sum_{q_1,q_2,\ldots,q_T} \pi_{q_1} b_{q_1}(\mathbf{o}_1) a_{q_1q_2} b_{q_2}(\mathbf{o}_2) \cdots a_{q_{T-1}q_T} b_{q_T}(\mathbf{o}_T). \quad (6.17)$$

The above equation tells us that, initially at time t=1 , we are in state q_1 with probability pie_1. And which generates the symbol o_1. With probability $b_{q1}(o_1)$
After this , the time t changes to t+1 = 2 and state transit to q_2 with probability a_{q1q2}., it also generates o_2 with probability $b_{q2}(o_3)$.
This process continues until last state transition occurs from q_{T-1} to q_T, which generates $b_{qT}(o_T)$.
For the calculation of P(O|lambda), according to above equation, involves $2T * N^T$. number of calculation, since for t= 1, 2 .. T, there are N possible states, hence N^T. possible state sequences.
These calculations can be very infeasible from even small number of N and T.
Say N= 5 and T=100 then $2*100*5^{100} = 10^{72}$ computations!
Hence a more efficient procedure is required to solve this.

Forward Procedure:

Consider a forward variable $alpha_t = P(o_1 o_2 \ldots. O_t, q_t |=i|lambda)$

$$\alpha_t(i) = P(\mathbf{o}_1\mathbf{o}_2\ldots\mathbf{o}_t, q_t = i|\lambda) \quad (6.18)$$

1. Initialization

$$\alpha_1(i) = \pi_i b_i(\mathbf{o}_1), \qquad 1 \le i \le N. \quad (6.19)$$

2. Induction

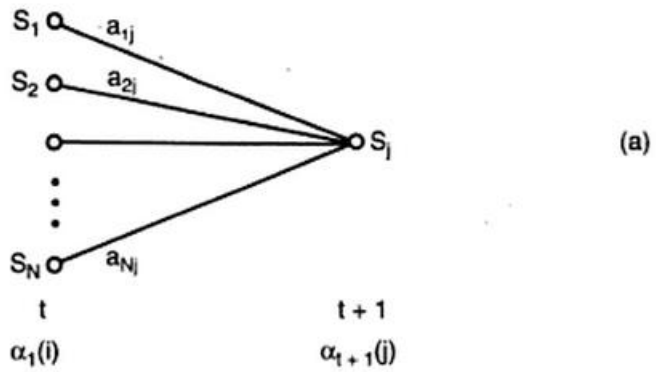
$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)a_{ij}\right] b_j(\mathbf{o}_{t+1}), \qquad \begin{matrix} 1 \le t \le T-1 \\ 1 \le j \le N \end{matrix} \quad (6.20)$$

3. Termination

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i). \quad (6.21)$$

Step 1:

Initialize the forward probabilities as joint probability of state *I* and initial observation o_i.
Step 2;
This is heart and very imp step, and it can be explained as shown in below figure:



This shows that state j can be reached at time t + 1 from N possible states, i.
Alpha(i) is the probability of the joint event oi, o2… oj are observed, and the state at time t is I, the product $alpha_i(i)a_{ij}$ is then the probability of the joint event that o1 , o2,… ot are observed, the state j is reached at time t+1 via state I at time t.
Summing the product all the N possible states, *I* , at time t results in probability of j at time t+1.
Step 3:
This gives desire calculation of p(O|lambda) as the sum of forward variable alphaIi).
This is the case since, by definition,
OOOO
This forward procedure requires N^2T calculations, rather than $2TN^T$ as required in direct calculation.
Say for N=5 and T =100, we need about 3000 computations for the forward method.
Forward probability calculation based upon the lattice structure is shown below figure

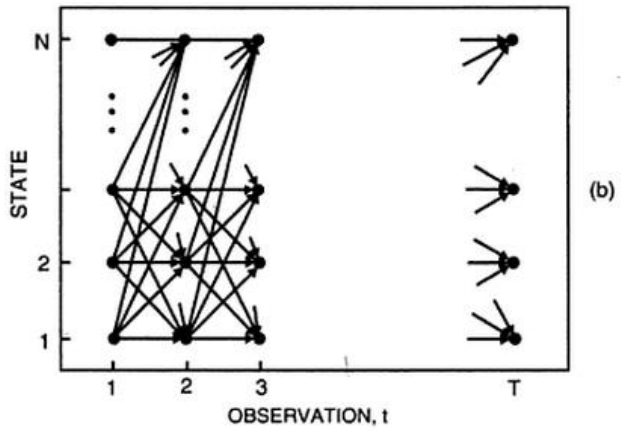


Figure 6.5 (a) Illustration of the sequence of operations required for the computation of the forward variable $\alpha_{t+1}(j)$. (b) Implementation of the computation of $\alpha_t(i)$ in terms of a lattice of observations t, and states i.

Figure 10.5

10.7.1 The backward procedure:

Consider a backward variable beta(i) defined as:

$$\beta_t(i) = P(\mathbf{o}_{t+1}\mathbf{o}_{t+2} \cdots \mathbf{o}_T \,|\, q_t = i, \lambda) \qquad (6.23)$$

We solve beta(i) inductively as follows:

1. Initialization

$$\beta_T(i) = 1, \quad 1 \le i \le N. \qquad (6.24)$$

2. Induction

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j),$$

$$t = T-1, T-2, \ldots, 1, \quad 1 \le i \le N. \qquad (6.25)$$

Step 1 arbitrarily defines $beta_T(i)$ to be 1 for all i.
Step 2 can be explained through following figure:

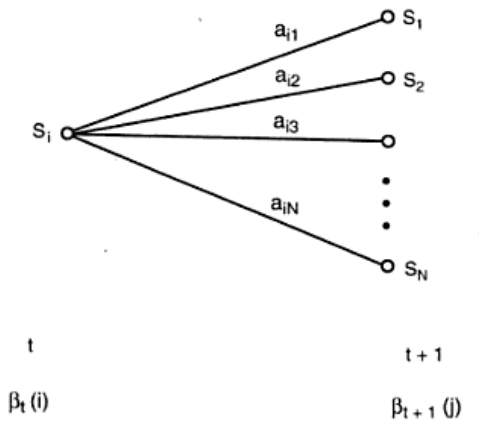


Figure 6.6 Sequence of operations required for the computation of the backward variable $\beta_t(i)$.

You have to consider all the possible states j at time t+1, accounting for the transition from I to j, as well as the observation o_{i+1} in the state j.
Computation of beta(i), requires on the order of N^2T calculations
This can be shown by following figure:
OOOOO

Solution to Problem 2 – "optimal State sequence"

There is no such exact solution as problem 1 to the problem 2.
This problem is to finding the best state sequence associated with given observation sequence.
The optimal criteria are to select the state q that is individually most likely at each time t.
To implement this solution, we define a posteriori probability variable,

$$\gamma_t(i) = P(q_t = i|\mathbf{O}, \lambda) \qquad (6.26)$$

Which is the probability of being in state I at time t, given the observation sequence O and model lambda?

We can express this in several forms as

$$\begin{aligned}\gamma_t(i) &= P(q_t = i \mid \mathbf{O}, \lambda) \\ &= \frac{P(\mathbf{O}, q_t = i \mid \lambda)}{P(\mathbf{O} \mid \lambda)} \\ &= \frac{P(\mathbf{O}, q_t = i \mid \lambda)}{\sum_{i=1}^{N} P(\mathbf{O}, q_t = i \mid \lambda)}.\end{aligned} \qquad (6.27)$$

Since P(O, q_t=i|lambda) is equal to $alpha_i$(i)$beta_T$(i), we can write it as

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \qquad (6.28)$$

$Alpha_i$(i) accounts for partial observation sequence o1, o2, … ot, and
$Beta_t$(i) accounts for the remainder of the observation sequence ot+1, ot+2, oT.

10.7.2The Viterbi algorithm:

We have to find the best state sequence q= (q 1, q2, … qT)
For the given observation sequence O= (o1, o2, … oT), we need to define the quantity

$$\delta_t(i) = \max_{q_1 q_2 \cdots q_{t-1}} P\left[q_1 q_2 \cdots q_{t-1}, q_t = i, \mathbf{o}_1 \mathbf{o}_2 \ldots \mathbf{o}_t \mid \lambda\right] \qquad (6.30)$$

Symbol is the best score or the highest probability along a single path at time t.
Using induction we can have,

$$\delta_{t+1}(j) = \left[\max_i \delta_t(i) a_{ij}\right] \cdot b_j(\mathbf{o}_{t+1}). \qquad (6.31)$$

For retrieving the state sequence, we have to keep track of the argument that maximized the above equation for each I and j.
The entire procedure can be now depicted as follows:

1. Initialization

$$\delta_1(i) = \pi_i b_i(\mathbf{o}_1), \quad 1 \le i \le N \tag{6.32a}$$

$$\psi_1(i) = 0. \tag{6.32b}$$

2. Recursion

$$\delta_t(j) = \max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij}\right] b_j(\mathbf{o}_t), \quad \begin{matrix} 2 \le t \le T \\ 1 \le j \le N \end{matrix} \tag{6.33a}$$

$$\psi_t(j) = \arg \max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij}\right], \quad \begin{matrix} 2 \le t \le T \\ 1 \le j \le N. \end{matrix} \tag{6.33b}$$

3. Termination

$$P^* = \max_{1 \le i \le N} \left[\delta_T(i)\right] \tag{6.34a}$$

$$q_T^* = \arg \max_{1 \le i \le N} \left[\delta_T(i)\right]. \tag{6.34b}$$

4. Path (state sequence) backtracking

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \qquad t = T-1, T-2, \ldots, 1. \tag{6.35}$$

This Viterbi also is almost same as implementing the forward calculation. But the difference is in maximization over previous states, while in forward procedure it was performed by summing.
Also the lattice structure efficiently implements the computations of this procedure.

10.7.3 Alternative Viterbi algorithm:

We can have a simple change in above algorithm, by taking logarithms of the model parameters, the Viterbi algorithm can be implemented without using including any multiplications.
Thus the procedure would look like as follows:

0. Preprocessing

$$\tilde{\pi}_i = \log(\pi_i), \qquad 1 \le i \le N$$
$$\tilde{b}_i(\mathbf{o}_t) = \log[b_i(\mathbf{o}_t)], \quad 1 \le i \le N,\ 1 \le t \le T$$
$$\tilde{a}_{ij} = \log(a_{ij}), \qquad 1 \le i, j \le N$$

1. Initialization

$$\tilde{\delta}_1(i) = \log(\delta_1(i)) = \tilde{\pi}_i + \tilde{b}_i(\mathbf{o}_1), \quad 1 \le i \le N$$
$$\psi_1(i) = 0, \qquad 1 \le i \le N$$

2. Recursion

$$\tilde{\delta}_t(j) = \log(\delta_t(j)) = \max_{1 \le i \le N} \left[\tilde{\delta}_{t-1}(i) + \tilde{a}_{ij}\right] + \tilde{b}_j(\mathbf{o}_t)$$
$$\psi_t(j) = \arg \max_{1 \le i \le N} [\tilde{\delta}_{t-1}(i) + \tilde{a}_{ij}], \qquad 2 \le t \le T,\ 1 \le j \le N$$

3. Termination

$$\tilde{P}^* = \max_{1 \le i \le N} [\tilde{\delta}_T(i)]$$
$$q_T^* = \arg \max_{1 \le i \le N} [\tilde{\delta}_T(i)]$$

4. Backtracking

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \qquad t = T-1, T-2, \ldots, 1$$

The calculation required for this implementations is on the order of $N^2 * T$ additions plus some calculations for prepossessing.

Exercise:

Given the model of the coin toss experiment, with three different coins and with following probabilities:

	State 1	State 2	State 3
$P(H)$	0.5	0.75	0.25
$P(T)$	0.5	0.25	0.75

All the state transition probabilities are equal to 1/3 and initial probabilities equal to 1/3 for the observed sequence:

$$\mathbf{O} = (H\ H\ H\ H\ T\ H\ T\ T\ T\ T)$$

Find the most likely path with the Viterbi algorithm.

Solution:

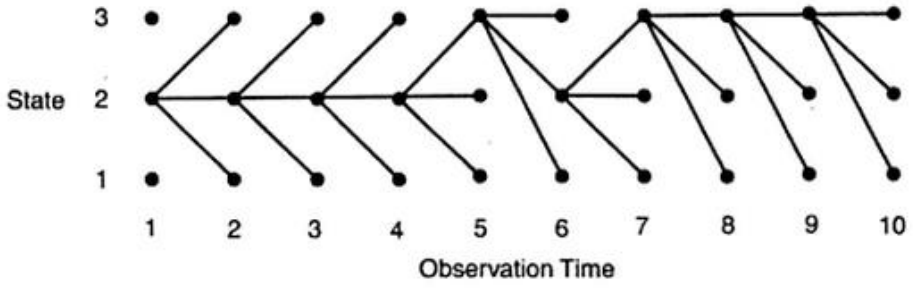
Since all a_{ij} coefficients are equal to 1/3, we can omit these terms as well as the initial probabilities, providing,

$$\delta_1(1) = 0.5, \quad \delta_1(2) = 0.75, \quad \delta_1(3) = 0.25.$$

The recursion for $\delta_t(j)$ gives $(2 \le t \le 10)$

$\delta_2(1) = (0.75)(0.5)$, $\delta_2(2) = (0.75)^2$, $\delta_2(3) = (0.75)(0.25)$
$\delta_3(1) = (0.75)^2(0.5)$, $\delta_3(2) = (0.75)^3$, $\delta_3(3) = (0.75)^2(0.25)$
$\delta_4(1) = (0.75)^3(0.5)$, $\delta_4(2) = (0.75)^4$, $\delta_4(3) = (0.75)^3(0.25)$
$\delta_5(1) = (0.75)^4(0.5)$, $\delta_5(2) = (0.75)^4(0.25)$, $\delta_5(3) = (0.75)^5$
$\delta_6(1) = (0.75)^5(0.5)$, $\delta_6(2) = (0.75)^6$, $\delta_6(3) = (0.75)^5(0.25)$
$\delta_7(1) = (0.75)^6(0.5)$, $\delta_7(2) = (0.75)^6(0.25)$, $\delta_7(3) = (0.75)^7$
$\delta_8(1) = (0.75)^7(0.5)$, $\delta_8(2) = (0.75)^7(0.25)$, $\delta_8(3) = (0.75)^8$
$\delta_9(1) = (0.75)^8(0.5)$, $\delta_9(2) = (0.75)^8(0.25)$, $\delta_9(3) = (0.75)^9$
$\delta_{10}(1) = (0.75)^9(0.5)$, $\delta_{10}(2) = (0.75)^9(0.25)$ $\delta_{10}(3) = (0.75)^{10}$

This leads to a diagram (trellis) of the form:



Hence the most likely state sequence is {2,2,2,2,3,2,3,3,3,3}.

Solution to Problem 3 – Parameter Estimation

The third most difficult problem of HMM is to define a method for obtaining or adjusting the model parameter lambda to satisfy all the optimization criterion.
There is not any analytical way to

$$\xi_t(i,j) = P\left(q_t = i, q_{t+1} = j \,|\, \mathbf{O}, \lambda\right). \qquad (6.36)$$

$$\begin{aligned}\xi_t(i,j) &= \frac{P\left(q_t = i, q_{t+1} = j, \mathbf{O}|\lambda\right)}{P\left(\mathbf{O}|\lambda\right)} \\ &= \frac{\alpha_t(i)a_{ij}b_j(\mathbf{o}_{t+1})\beta_{t+1}(j)}{P\left(\mathbf{O}|\lambda\right)} \\ &= \frac{\alpha_t(i)a_{ij}b_j(\mathbf{o}_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(\mathbf{o}_{t+1})\beta_{t+1}(j)} \qquad (6.37)\end{aligned}$$

$$\gamma_t(i) = \sum_{j=1}^{N}\xi_t(i,j). \qquad (6.38)$$

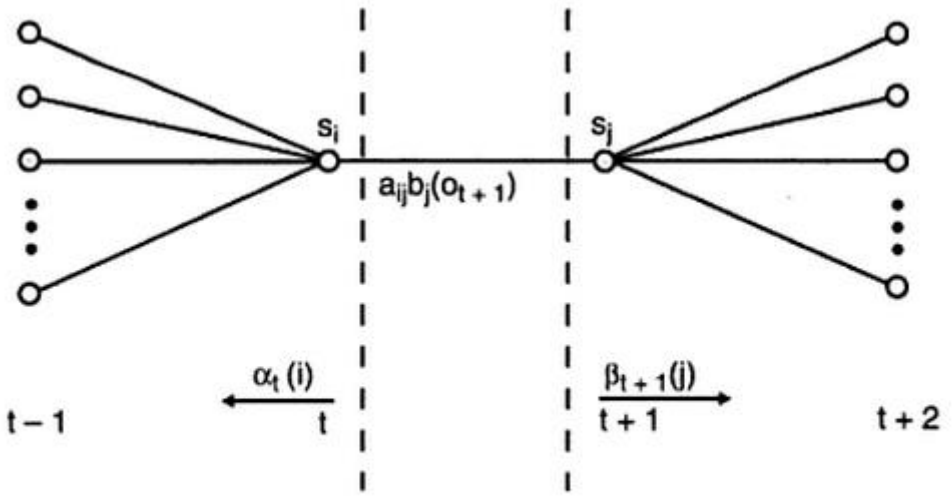


Figure 6.7 Illustration of the sequence of operations required for the computation of the joint event that the system is in state i at time t and state j at time $t+1$.

$$\sum_{t=1}^{T-1}\gamma_t(i) = \text{expected number of transitions from state } i \text{ in } \mathbf{O} \tag{6.39a}$$

$$\sum_{t=1}^{T-1}\xi_t(i,j) = \text{expected number of transitions from state } i \text{ to state } j \text{ in } \mathbf{O}. \tag{6.39b}$$

$$\bar{\pi}_i = \text{expected frequency (number of times) in state } i \text{ at time } (t=1) + \gamma_1(i) \tag{6.40a}$$

$$\bar{a}_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i}$$

$$= \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}. \tag{6.40b}$$

$$\bar{b}_j(k) = \frac{\text{expected number of times in state } j \text{ and observing symbol } \mathbf{v}_k}{\text{expected number of times in state } j}$$

$$= \frac{\sum_{\substack{t=1 \\ \text{s.t.} o_t = \mathbf{v}_k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}. \qquad (6.40c)$$

References and further Reading

[1] L.R. Rabiner , B. H. Juang , "Fundamentals of Speech Recognition", Prentice-Hall, Englewood Cliffs, N.J.

IDOL

CHAPTER 11

THEORY AND IMPLEMENTATION OF HIDDEN MARKOV MODELS PART -B

11.0 Objectives:

- Types of HMMs
- Implementation issues for HMMs
- HMM system for isolated word Recognition

11.1 TYPES OF HMMS

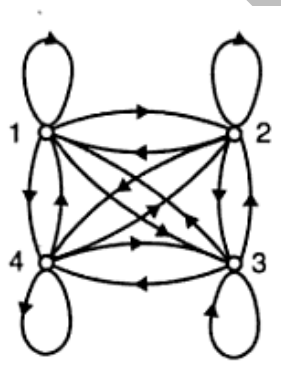
Using the structure of transition matrix A, we can classify the types of HMMs.

There are three types of HMM:

1. Erodic Model or Fully Connected HMM
2. Left Right Model or Bakis Model
3.Cross Coupled Model

11.1.1 Erodic Model or Fully Connected HMM:

In this type of model, every state of the model can be reached to every other state (mostly in one single step).
In other words, an erodic model has property that every state can be reached from every other state in some distinct or finite steps.
Consider a model with N = 4 states as shown below, as each state can be reached from every other state, every a_{ij} coefficient is positive.



(a)

For the above type of model, state transition matrix would be as follows:

IDOL

$$A=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}.$$

11.2.2 Left Right Model or Bakis Model

Another type of HMM has been found to model observed properties of signal, better than erodic model.
One such model is left right model. It is so called because of the state sequence associate with the model. Here, the state sequence increasesor remain same with the increase in the time dimension, that is, the states transit from left to right. From this property, we can clearly say that, this type of model can easily represent whose property changes with change in time in a successive manner from an example Speech.

The fundamental property of this model is, the state transition coefficients have

$$a_{ij} = 0, \qquad j < i \tag{6.45}$$

Which represent that, transition to a state whose indices are lower than current state, is not allowed. The next state to transit should have grater or equal indices as current state.
The initial state probability will have following property

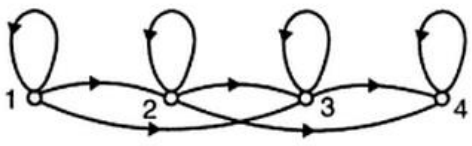
$$\pi_i = \begin{cases} 0, & i \neq 1 \\ 1, & i = 1 \end{cases} \tag{6.46}$$

This is because the state sequence must start with state 1 and end with State N.

There arefew constraints on the state transition coefficients to make sure that large change in state transition does not take place;, the constraints can be formulated as follows/:

$$a_{ij} = 0, \qquad j > i + \Delta i \tag{6.47}$$

Consider the model shown below, with N = 4. Is one example of left right model, where the states are transit to states of higher or equal indices i.e. moved from left to right.



(b)

If you can notice, in the above model, Triangle(symbol) I is 2 that is no jumps with more than 2 states are allowed.
The state transition matrix A, for the above model would look like as follows:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix}.$$

From the above model, we can also depict that the last state in left right model, the state transition coefficients are specified as

$$a_{NN} = 1 \qquad (6.48a)$$

$$a_{Ni} = 0, \quad i < N. \qquad (6.48b)$$

11.1.3 Cross - coupled model

This is another variation of left right model, where two parallel left right HMMs are represented.

It does follow all the constraints of left right model related to a_{ij}. However it has certain flexibility than left right model.

Following figure shows a 6 state parallel path left right model:

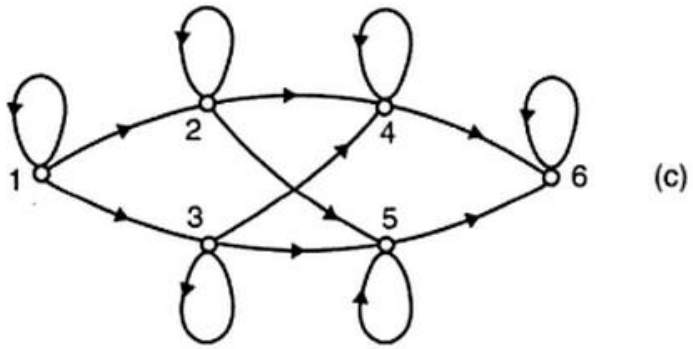


Figure 6.8 Illustration of three distinct types of HMMs. (a) A 4-state ergodic model. (b) A 4-state left-right model. (c) A 6-state parallel path left-right model.

IDOL

Also note that, imposition of the constraints of left right model or those ofjumped model, have no effect on the re-estimation procedure. This is the case, any HMM parameter set to zero initially will remain zero throughout the same.

11.2 IMPLEMENTATION ISSUES FOR HMMS

In this section ,we deal with several practical implementation issues, including scaling, multiple observation and missing data and choice of model size and types.

11.2.1 Scaling

To understand why scaling is required for implementing the reestimation procedure of HMMs , consider the definition of $\alpha_t(i)$. It can be seen that $\alpha_t(i)$ consists of the sum of a large number of terms,each of the form

$$\left(\prod_{s=1}^{t-1} a_{q_s q_{s+1}} \prod_{s=1}^{t} b_{q_s}(\mathbf{o}_s)\right)$$

With q_t = I and b is a discrete probability as defined. Since each a and b term is less than 1, it can be seen that as t starts to get big (10 or more), each term of $\alpha_t(i)$ starts to head exponentially to zero. For sufficiently large t (100 or more) the dynamic range of the $\alpha_t(i)$ computation will exceed the precision range. Hence the only resonable way to perform the computation is to incorporate a scaling procedure.

11.2.2 Multiple Observations Sequences

The major problem with left – right models is that one cannot use a single observation sequence to train the model (i.e. for reestimation of model parameters).This is because the transient nature of the states within the model allows only a small number of observations for any state.Hence , to have sufficient data to make reliable estimates of all model parameters, one has to use multiple observation sequences.

We denote the set of K observation sequences as

$$\mathbf{O} = [\mathbf{O}^{(1)}, \mathbf{O}^{(2)}, \ldots, \mathbf{O}^{(K)}]$$

Where $\mathbf{O}^{(k)} = (\mathbf{o}_1^{(k)} \mathbf{o}_2^{(k)} \ldots \mathbf{o}_{T_k}^{(k)})$ is the kth observation sequence. Assume each observation sequence is independent of every other observation sequence , goal is to adjust the parameters of the model λ to maximize

$$P(\mathbf{O}|\lambda) = \prod_{k=1}^{K} P(\mathbf{O}^{(k)}|\lambda)$$

$$= \prod_{k=1}^{K} P_k.$$

Reestimation formulas are based on frequencies of occurrences of various events,Thus the modified reestimation formulas for a_{ij} and $b_j(L)$ and π_i is not reestimated since $\pi_1 = 1$, $\pi_i = 0$, $i \neq 1$.

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) a_{ij} b_j(\mathbf{o}_{t+1}^{(k)}) \beta_{t+1}^k(j)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) \beta_t^k(i)}$$

and

$$\overline{b_j}(\ell) = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{\substack{t=1 \\ \text{s.t. } O_t = v_\ell}}^{T_k-1} \alpha_t^k(i) \beta_t^k(i)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) \beta_t^k(i)}$$

In this manner, the same scale factors will appear in each term of the sum over t as appears in the Pk term, and hence will cancel exactly.

11.2.3 Initial Estimates of HMM Parameters

11.2.3.1 HMM SYSTEM FOR ISOLATED WORD RECOGNITION

Let us consider HMM to build an isolated word recognizer.
Assume we have a vocabulary of V words to be recognized and for each word is to be modeled by distinct HMM.
For each word there is training set of K utterances of that word, which represent various observation sequences.
For isolated word recognizer following steps is performed:
1. Build $lambda_v$ for each word v in the vocabulary; here we have to estimate the AB and pie, of the training observation sequence of the vth word.

2. For each unknown word to recognize following procedure shown in figure is followed:

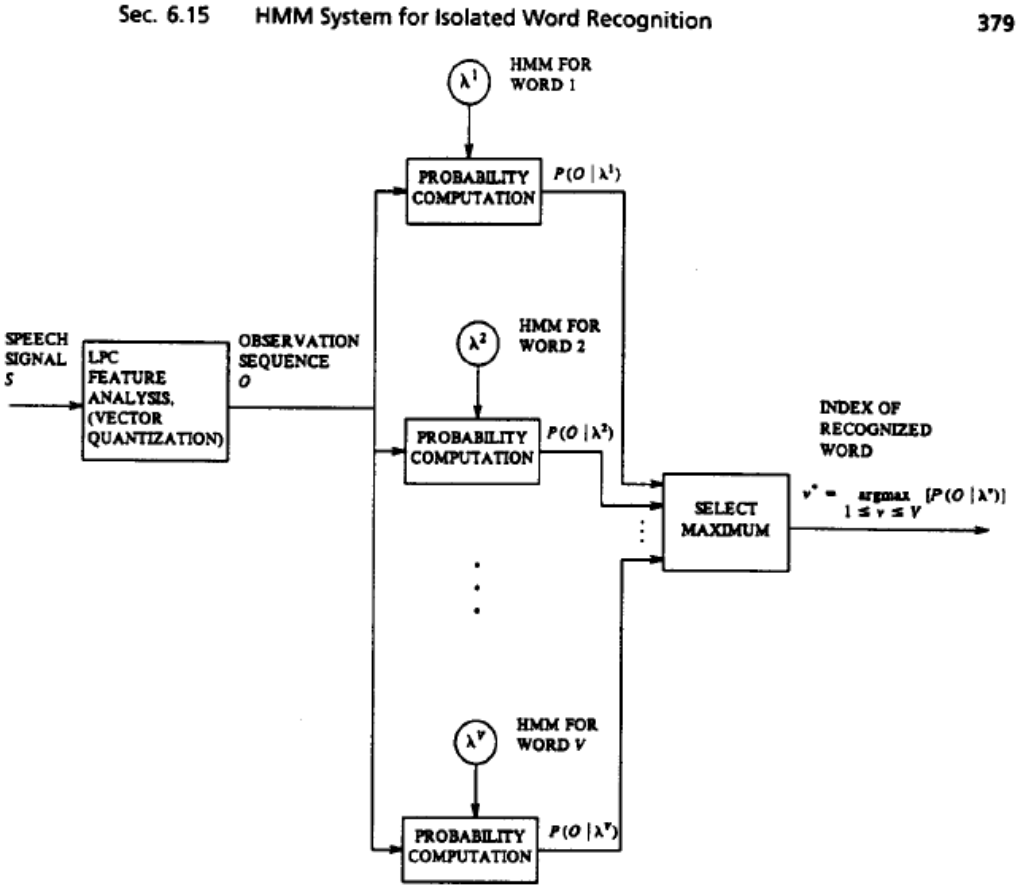


Figure 6.13 Block diagram of an isolated word HMM recognizer (after Rabiner [38]).

First, Measurement of the Observation sequence O via feature analysis of the speech is calculated.
Then, Calculation of model likelihood for all possible models i.e. P(O|lambda) is calculated.
Probability computation is mostly performed with Viterbi algorithm., which requires $V * N^2 * T$ computations.
For V=100, N=5 and T=40, around 10^3 computations are required.
Finally, Selection of the word whose likelihood is highest, i.e.

$$v^* = \arg \max_{1 \le v \le V} [P(\mathbf{O}|\lambda_v)]. \qquad (6.122)$$

11.2.3.2 Choices of Model Parameter:

For isolated word recognizer, now we have to select the parameters of the model.
It is clear that left –right model is very suitable for word recognizer, as we can associate model states with respect to time in sequential manner.
For selecting number of states there are two possible ways,

One is, let number of states approx. equal to the number of sounds within the word, hence 2 to 10 states would be appropriate.
Another way is, let the number of states approx. equal to the average number of observations in a spoken versions of the word.
We also restrict the word model to have same number of states, for each word.
Following figure shows the effect of varying the number of states in word model.
It shows the pilot of average word error rate vs. N .
From below graph we can see, the error is insensitive for N , also for N=6 have the less error rate as compared to other Ns.

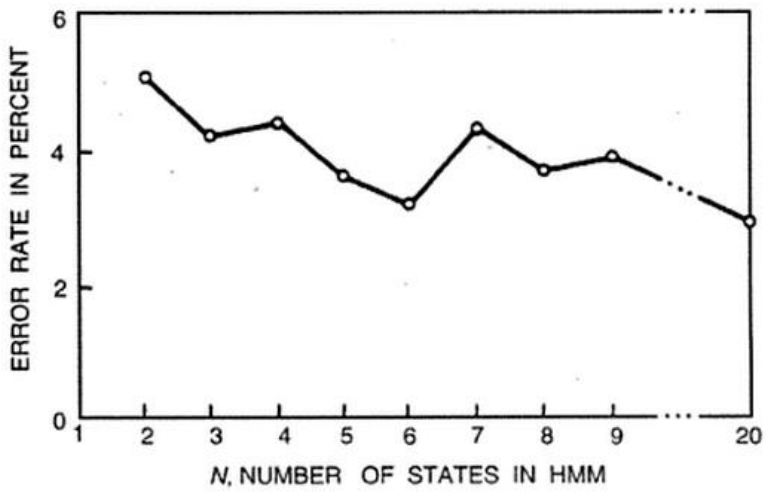


Figure 6.14 Average word error rate (for a digits vocabulary) versus the number of states N in the HMM (after Rabiner et al. [18]).

Now , next is to select the observation vector.
For continues model, autocorrelation coefficient or weighted cepstral coefficients are used as observation vector.
Here, we may use, M=64~256 mixture per state.
It is easier and convenient to use diagonal covariance matrices.
For discrete model, a codebook is generated of discrete symbols.
Here, the codebook is of as many as M=512~1024 code words.
Following figure shows how the mixture densities for modeling observations vectors are needed.
It also shows a comparison of marginal distribution $b_j(o)$ against histogram of the actual observations within a state.
The covariance matrices are constrained to be diagonal for each individual mixture.
In figure, the first model state of word “zero”, is shown.

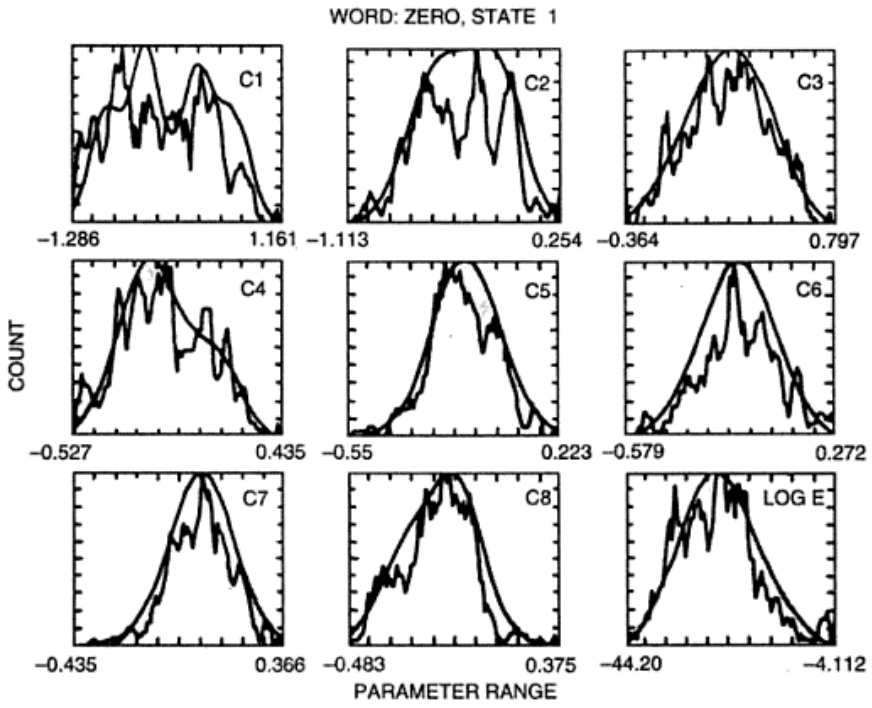


Figure 6.15 Comparison of estimated density (jagged contour) and model density (smooth contour) for each of the nine components of the observation vector (eight cepstral components, one log energy component) for state 1 of the digit zero (after Rabiner et al. [38]).

Another fact about the HMM is that to limit the parameters estimates to prevent it from becoming very small.

Consider an example of the constraint $b_j(k)$ to be greater than or equal to some minimum value l.

As shown in the below figure, it should be clear that there is always a finite probability of its occurrences, when scoring an unknown observation ser. It shows a curve of average error rate vs. the parameter

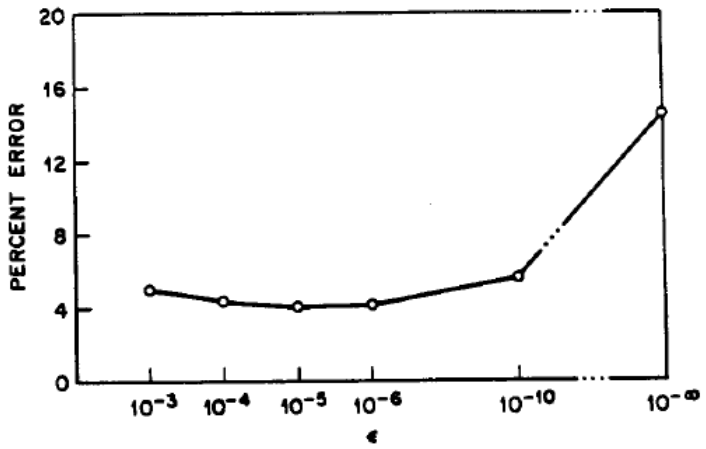


Figure 6.16 Average word error rate as a function of the minimum discrete density value ϵ (after Rabiner et al. [18]).

From this we can see that from a range 10^{-3} to 10^{-10}, the average error rate is almost constant. But after that the error rate increases very sharply. Hence it is imp to constraint the value of coefficients to be in a specific range.

11.2.3.3 Segmental K-means segmentation into states:

K means iterative procedure for clustering data is used for the initial estimates of the parameters.

As shown in the figure,

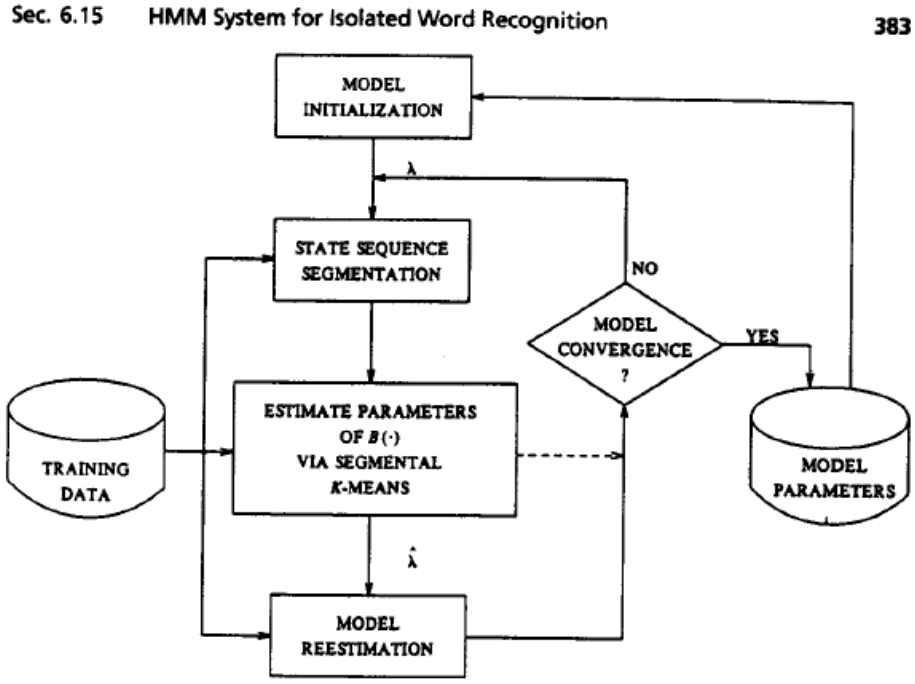


Figure 6.17 The segmental *k*-means training procedure used to estimate parameter values for the optimal continuous mixture density fit to a finite number of observation sequences (after Rabiner et al. [38]).

Here we have a training set of observations and an initial estimates of parameter.

After model initialization, the training observation sequence is segmented into states. This is achieved by using Viterbi algorithm and then backtracking along the optimal path.

The result of segmenting each training sequence is, for every N states, a maximum likelihood estimates of the set of the observation that occur within exact state j according to current model.

Here we consider, the model of discrete symbol, and each observation sequence is coed using the codebook.

The update estimates of $b_j(k)$ parameter is

$\hat{b}_j(k)$ = number of vectors with codebook index k in state j divided by the number of vectors in state j.

When we use, continuous observation densities, a segmental K means procedure is used to cluster the observation vectors thing each state j into a set of M clusters, where each cluster represents one of the M mixtures of the $b_j(o_t)$ density.
From clustering, an updates set of model parameters is derived as follows:

$\hat{c}_{jm}$ = number of vectors classified in cluster m of state j divided by the numb of vectors in state j

$\hat{\mu}_{jm}$ = sample mean of the vectors classified in cluster m of state j

$\hat{U}_{jm}$ = sample covariance matrix of the vectors classified in cluster m of state j

Incorporation of states Duration into the HMM

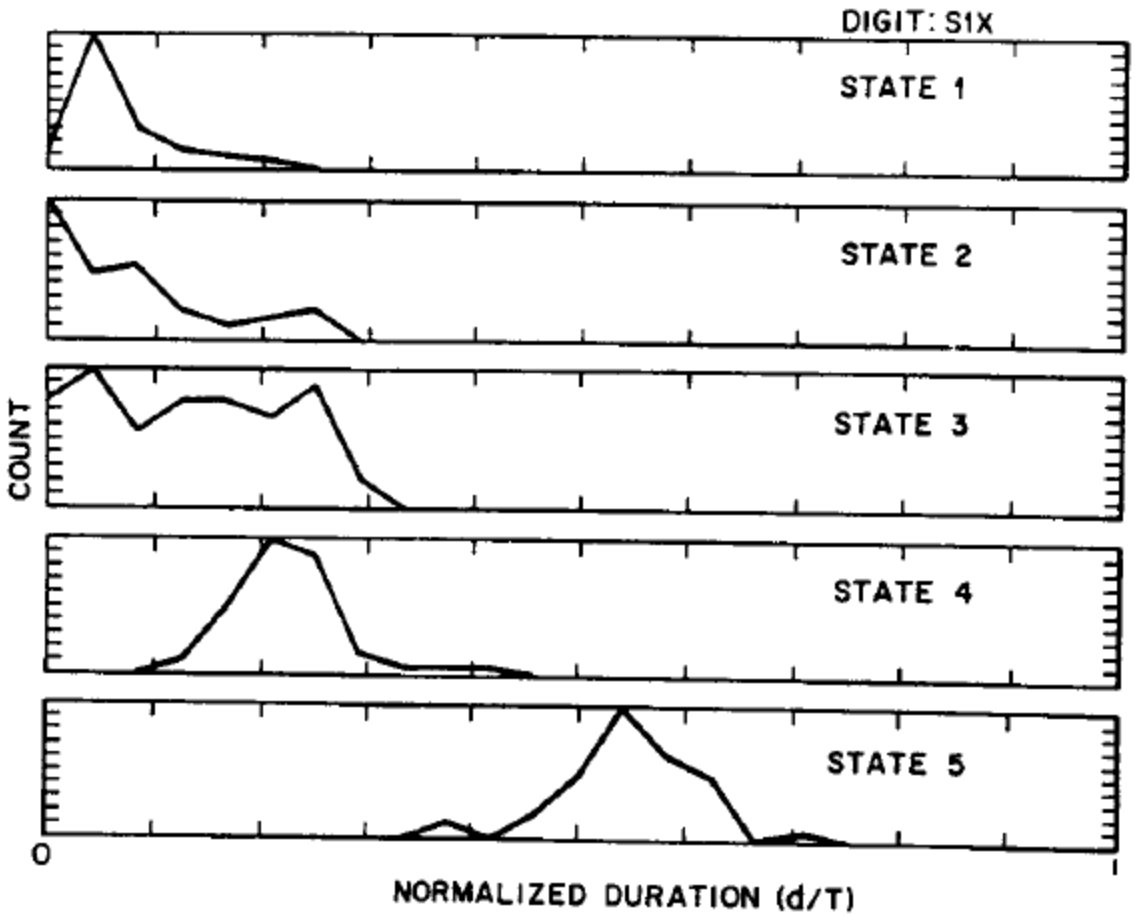


Figure 6.19 Histograms of the normalized duration density for the five states of the digit "six" (after Rabiner et al. [38]).

$$\log \hat{P}(\mathbf{q}, \mathbf{O}|\lambda) = \log P(\mathbf{q}, \mathbf{O}|\lambda) + \alpha_d \sum_{j=1}^{N} \log [p_j(d_j)] \qquad (6.123)$$

Reference and further reading:

[1] L. Rabiner and B. Juang, “Fundamentals of Speech Recognition”, Pearson Education.

IDOL

CHAPTER 12

SPEECH RECOGNITION BASED ON CONNECTED WORDS MODELS PART- A

12.0 Objectives

- General notations for the connected Word-Recognition problem
- The two level dynamic programming algorithm
- The level building algorithm
- Connected digit recognition implementation

12.1 INTRODUCTION

Up till now, we were discussing about isolated word or phrase recognition. In that we made certain assumptions.

One of them is that, speech which is to be recognized was a single word a single phrase. Also it has to be recognized complete entity with no explicit knowledge of phonetic content of the word. Hence, for vocabulary of V words, the recognition procedure consists of matching the spectral vector sequences of unknown spoken words and selecting the best matching pattern of the spoken word.

Another assumption was, each spoken word has a defined beginning and ending, which can be find out using End Point detection algorithm. For application s like “Command and Control”, where application requires to a single word as command, this type of recognizer works fine and appropriate.

However application which requires recognizing continuous sequence of words from the vocabulary, at that scenario, isolated recognizer does not fit in.

Following figure shows a three digit sequence in which the upper energy graph shows the string spoken as isolated digits, i.e. after each word there is a small pause.

IDOL

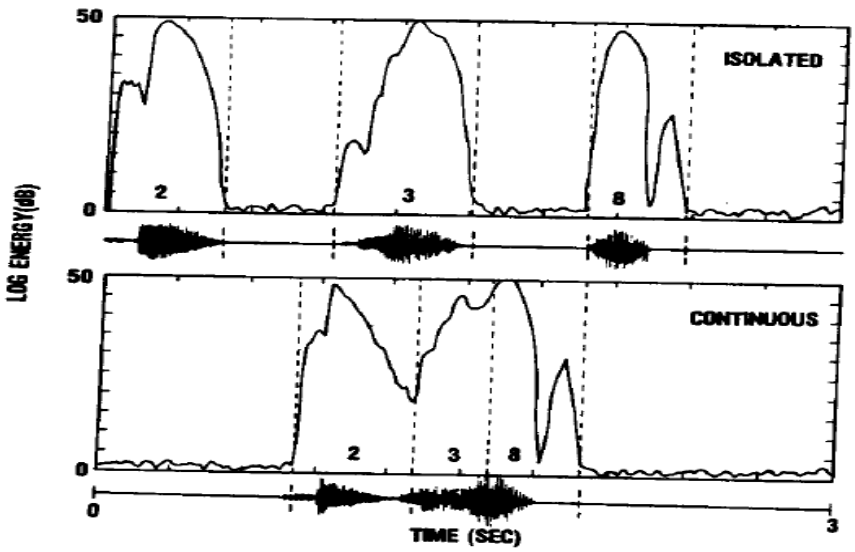


Figure 7.1 Illustration of an isolated string of digits (upper panel) and a fluently spoken version of the same digit string (lower panel).

Figure: 12.1. illustration of an isolated string of digits and fluently spoken version of the same digit string.

And the lower panel energy graph shows, sequence of same digits spoken without having any pause, i.e. in continues manner.

Speaking isolated digits looks unnatural and takes 2.5 sec, while the continues manner spoken digits takes only 1 sec. Hence it is needed for the techniques to recognize fluent speech.

From the view of speech recognition algorithm, there are two classes of fluent speech strings:

One type is with set of strings derived small to moderate size vocabularies, including digits, strings, letter sequences, combinations of alpha numerals and strings for accessing limited databases, based on small to moderate vocabularies. The basic speech recognition unit here is a word or phrase as in isolated word recognition.

Another type is set of continues speech strings derived from moderate to large size vocabularies, where basic speech recognition unit cannot be a single word because of its complexity constraints. In this case, sub word speech recognition is required.

IDOL

Now let us formulate the problem of the connected word recognition, for that we will refer the below figure:

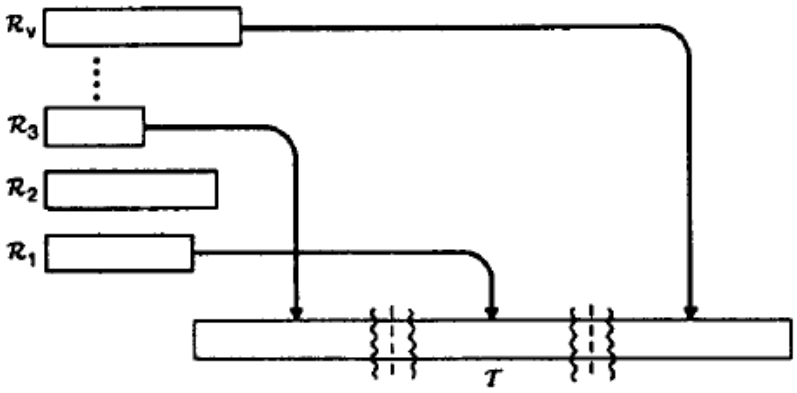


Figure 7.2 Illustration of the connected word-recognition problem.

Figure 12.2 : Illustration of the connected word-recognition problem

Consider, we have spectral vectors from a fluently spoken string of words, T = {t(1), t(2),…,t(M)}, and we also have the spectral patterns for each V reference patterns, (R_1 to R_v) corresponding to V unique words in the vocabulary.
This problem can be states as,

Given a fluently spoken sequence of word, how can we determine the optimum match in terms of a concatenation of word reference pattern?

We need to first solve following problem before resolving above problem.

1. The number of words, L, in the string is unknown to us, (we can just have the range of the numbers of words.
2. The utterance boundaries of spoken words within the sequence are not known., hence we just have the beginning of first word and ending of last word, no other info is available of middle words.
3. The word boundaries are usually fuzzy or non-unique.. and this makes difficult to find out the exact boundaries of words, because of sound articulation..

IDOL

For example, boundaries between the digit 3 and digit 8 in the figure 1, is fuzzy as the ending of word three /i/ and the starting of word eight /e/ are articulates. Fuzziness of the boundaries is shown as squiggly lines in figure 2

4. For a set of V word reference patterns, and for a given value of L, there are V^L possible combinations of composite matching patterns, for any small values of V and L, he exponential number of composite matching patterns implies that the connected word problem cannot be solved by exhaustive means. A non-exhaustive algorithm is required for connected word recognition.

12.2 GENERAL NOTATIONS FOR THE CONNECTED WORD-RECOGNITION PROBLEM

We denote spectral sequence of vectors of the test patterns as

$$\mathcal{T} = \{\mathbf{t}(1), \mathbf{t}(2), \ldots, \mathbf{t}(M)\} = \{\mathbf{t}(m)\}_{m=1}^{M} \qquad (7.1)$$

t(m) is an appropriate spectral vector (LPC or filter bank)

Similarly, we denote, set of word reference patterns as R_i, 1<i<V word vocabulary, where each pattern is of the from,

$$\mathcal{R}_i = \{\mathbf{r}_i(1), \mathbf{r}_i(2), \ldots, \mathbf{r}_i(N_i)\} \qquad (7.2)$$

Where N_i, is the duration of the i^{th} word reference pattern.

The connected word recognition problem, can be states as finding the optimum sequence of word reference patterns R^*, the best matches T. By assuming there are L word patterns, the best sequence pattern R^*, is a concatenation of L reference patterns i.e.

$$\mathcal{R}^* = \{\mathcal{R}_{q^*(1)} \oplus \mathcal{R}_{q^*(2)} \oplus \mathcal{R}_{q^*(3)} \oplus \cdots \oplus \mathcal{R}_{q^*(L)}\} \qquad (7.3)$$

In which each index, $q^*(l)$ is in the range [1,V].

Consider an arbitrary "super reference" pattern R^s to determine the sequence of word indices $q^*(l)$, of the form

$$\mathcal{R}^s = \mathcal{R}_{q(1)} \oplus \mathcal{R}_{q(2)} \oplus \mathcal{R}_{q(3)} \oplus \cdots \oplus \mathcal{R}_{q(L)} = \{\mathbf{r}^s(n)\}_{n=1}^{N^s} \qquad (7.4)$$

In which N^s is the total duration of the concatenated reference pattern R^s.

The time aligned distance between R^s and T is obtain via dynamic wrapping, shown in below figure,

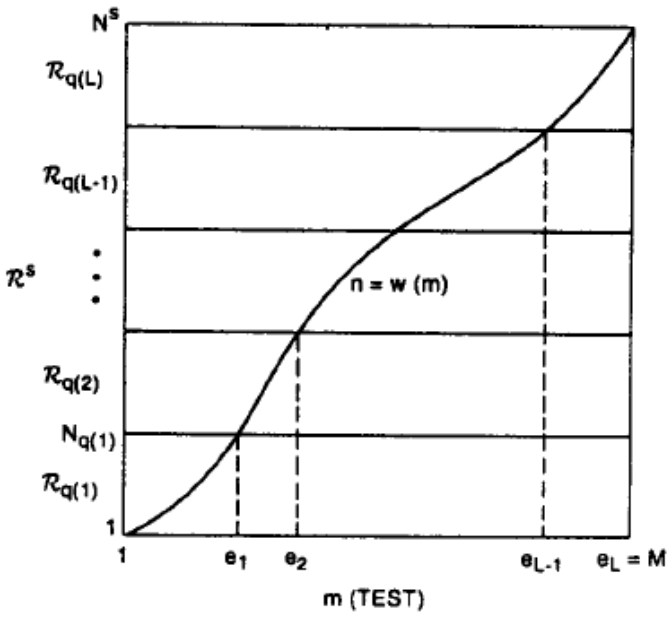


Figure 7.3 Determination of the optimum alignment of super-reference pattern $\mathcal{R}^s$ to $\mathcal{T}$, along with corresponding word boundary frames.

Figure:12.3 Determination of the optimum alignment of super-reference pattern

An the distance is obtained using following formula:

$$D(\mathcal{R}^s, \mathcal{T}) = \min_{w(m)} \sum_{m=1}^{M} d(\mathbf{t}(m), \mathbf{r}^s(w(m))) \qquad (7.5)$$

Here D() is a local spectral distance measure, w(.) is a warping function for the time index.

The word boundary frames in the input string can be found on the basis of the word boundary frames in super reference pattern, by using appropriate back tracking, as illustrated in above figure.

As shown in the figure, the first frame in the reference word $r_{q(1)}(N_{q(1)})$ maps to frame e_1 in the test pattern. And the last frame in the second reference word $r_{q(2)}(N_{q(2)})$ maps to frame e_2 in the test pattern and so on.

$$D^* = \min_{\mathcal{R}^s} D(\mathcal{R}^s, \mathcal{T})$$

$$= \min_{L_{min} \le L \le L_{max}} \min_{\substack{q(1),q(2),\ldots,q(L) \\ 1 \le q(i) \le V}} \min_{w(m)} \sum_{m=1}^{M} d(\mathbf{t}(m), \mathbf{r}^s(w(m))) \qquad (7.6)$$

12.3 THE TWO LEVEL DYNAMIC PROGRAMMING ALGORITHM (TWO LEVEL DP)

General idea of two levels DP algorithm is to break up the continuous equation of 7.6 into two stages i.e. two levels.

At first level, match the each individual word reference pattern R_v against arbitrary position of test string.

Consider the following figure:

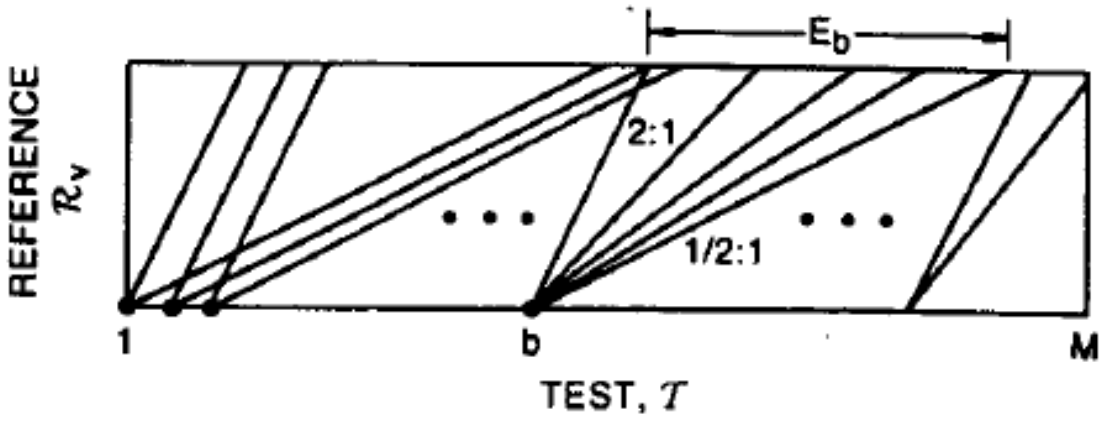


Figure 7.4 Computation ranges for matching $\mathcal{R}_v$ against portions of $\mathcal{T}$.

Figure: 12.4 .Computation ranges for matching R, against portions of T

For the range of beginning test frames of the match b, 1< b<M, for range of ending test pattern , e , 1<e<M.

We have to compute:

$$\hat{D}(v,b,e) = \min_{w(m)} \sum_{m=b}^{e} d(\mathbf{t}(m), \mathbf{r}_v(w(m))) \qquad (7.9)$$

This equation gives the minimum distance for every possible vocabulary pattern, R_v, between each pair of beginning and ending frames (b, e.)

We can eliminate v by finding best match between b and e, giving

$$\tilde{D}(b,e) = \min_{1\le v\le V} [\hat{D}(v,b,e)] = \text{best score} \qquad (7.10)$$

$$\tilde{N}(b,e) = \arg \min_{1\le v\le V} [\hat{D}(v,b,e)] = \text{best reference index} \qquad (7.11)$$

Due to this, data storage is reduced by factor of v with no loss of optimality.

Even this equation can be reduced by using a range limited DTW algorithm as shown in below figure:

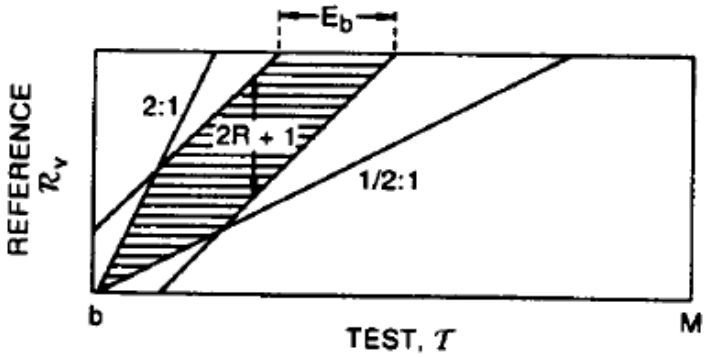


Figure 7.5 Use of range limiting to reduce the size of individual time warps.

Figure 12.5 Use of range limiting to reduce the size of individual time warps.

Here, by selecting appropriate range variable R, the ending region E_b, can be made smaller for a normal 2:1 expansion.

After the array of best score i.e. D(b,e), is obtained from the first level. The next level is to bring together all the individual reference pattern score to minimize the overall accumulated distance over the entire test string. This is achieved using dynamic programming.

Consider ending frame e, as shown in the below figure:

Figure 7.6 Series of paths ending at frame *e*.

Figure 12.6 Series of paths ending at frame e.

The shortest distance between a concatenated sequence of l reference patterns as $D_l(e)$, can be computed as

$$\bar{D}_\ell(e) = \min_{l \le b < e} [\tilde{D}(b,e) + \bar{D}_{\ell-1}(b-1)] \qquad (7.12)$$

Hence, the best path ending at frame e using l reference pattern is one with minimum distance over all possible beginning frames b, of the concatenation of the best path ending at frame b-1 suing exactly l-1 patterns, plus the distance of eq 7.10 of the best path from frame b to frame e.

We can formulate the DP algorithm for finding the overall best path.

To understand this algorithm better, let us take one example o0f decoding a given matrix of distances, D(b,e) into best strings.

Exercise: 12.1:

Assume we are given 15 cross 15 matrix of distances, D(b,e), representing the best accumulated distances between frames b and e, as shown below:

ending frame, e

beginning frame, b	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	–	–	9	10	13	17	22	25	29	33	37	41	45	50	60
2	–	–	–	7	10	14	17	21	25	29	32	36	40	45	53
3	–	–	–	–	9	13	16	19	23	26	28	32	35	40	47
4	–	–	–	–	–	6	8	9	11	12	14	19	23	28	33
5	–	–	–	–	–	–	4	6	8	10	12	15	19	23	30
6	–	–	–	–	–	–	–	9	12	16	19	23	27	30	33
7	–	–	–	–	–	–	–	–	15	18	22	27	32	37	45
8	–	–	–	–	–	–	–	–	–	12	16	21	25	29	33
9	–	–	–	–	–	–	–	–	–	–	6	8	10	13	16
10	–	–	–	–	–	–	–	–	–	–	–	5	7	9	12
11	–	–	–	–	–	–	–	–	–	–	–	–	4	6	8
12	–	–	–	–	–	–	–	–	–	–	–	–	–	2	4
13	–	–	–	–	–	–	–	–	–	–	–	–	–	–	2
14	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
15	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

1. **Find the best path for a 1, 2, and 3 word match.**
2. **What are the total distance scores for the 3 string lengths? Which string length is most likely?**

Solution to 1:

We will be using dynamic programming solution to determine the best one, two and third word match

From the recursion above we get

$$\bar{D}_1(e) = \tilde{D}(1, e), \qquad 2 \le e \le 15$$

$$\bar{D}_2(e) = \min_{1 \le b < e} [\tilde{D}(b, e) + \bar{D}_1(b-1)], \qquad 3 \le e \le 15$$

$$\bar{D}_3(e) = \min_{1 \le b < e} [\tilde{D}(b, e) + \bar{D}_2(b-1)], \qquad 4 \le e \le 15$$

From the matrix of distance we get,

e	$\bar{D}_1(e)$	$\bar{D}_2(e)$	$\bar{D}_3(e)$
3	9	-	-
4	10	-	-
5	13	-	-
6	17	$15 = (6 + 9)$	-
7	22	$14 = (4 + 10)$	-
8	25	$16 = (6 + 10)$	-
9	29	$18 = (8 + 10)$	$30 = (\bar{D}_2(6) + 15)$
10	33	$20 = (10 + 10)$	$26 = (\bar{D}_2(7) + 12)$
11	37	$22 = (12 + 10)$	$22 = (\bar{D}_2(8) + 6)$
12	41	$25 = (15 + 10)$	$23 = (\bar{D}_2(9) + 5)$
13	45	$29 = (19 + 10)$	$24 = (\bar{D}_2(10) + 4)$
14	50	$33 = (23 + 10)$	$24 = (\bar{D}_2(11) + 2)$
15	60	$40 = (30 + 10)$	$26 = (\bar{D}_2(12) + 4)$

For a one word match, we have D1 = D1 (15) =60, with path (1,15).

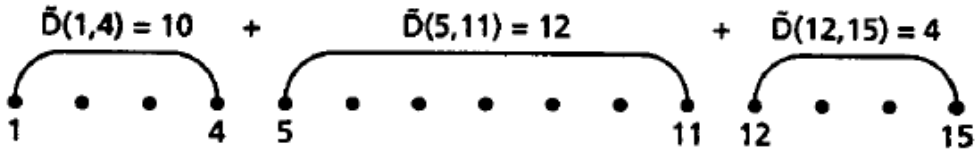
For a two word match, we have D2 = D2(15) =40, with path (1,4),(5,15).. It means 1 word span between frame 1 to 4 and word 2 spans between frames 5 to 15.

For a three word match, we have D3(15) = 26, with path (1,4),(5,11) and (12,15), it means word 1 span frame 1 to 4, word 2 spans frame 5 to 11 and word 3 spans 12 to 15.

Solution 2:

From the above distances, of word one, two and three matches, which are 60, 40 and 26. Clearly the length 3 has the smallest distance and is most likely.

The overall path of the 3 word string is



COMPUTATION OF THE TWO LEVEL DP ALGORITHM

The total computation of two level dp algorithm is to determine D(v, b, e). this is computations is essentially the computation of V.M time warps corresponding to a single time warp for each reference pattern V and for each training frame M.
The size of each time warp is approx. N(2R+1) for an R frame range limited DTW search with reference pattern of avg length N.

Hence total computation of the two level DP algorithm is:

$$C_{2L} = V \cdot M \cdot \bar{N}(2R + 1) \text{ (grid points)} \quad (7.13)$$

The required storage of the range reduced algorithm is :

$$S_{2L} = 2M(2R + 1) \quad (7.14)$$

For a typical set of values i.e. M=300 frames, N=40 frames, V = 10 words and R = 5 frames, it requires computation for 1,320,000 grid points and storage of 6600 locations.

12.4 LEVEL BUILDING (LB) ALGORITHM

Following figure a shows the computation associated with time aligning R^s to T. It provides computation of accumulated distance at each grid point within a constrained region of vertical strips.

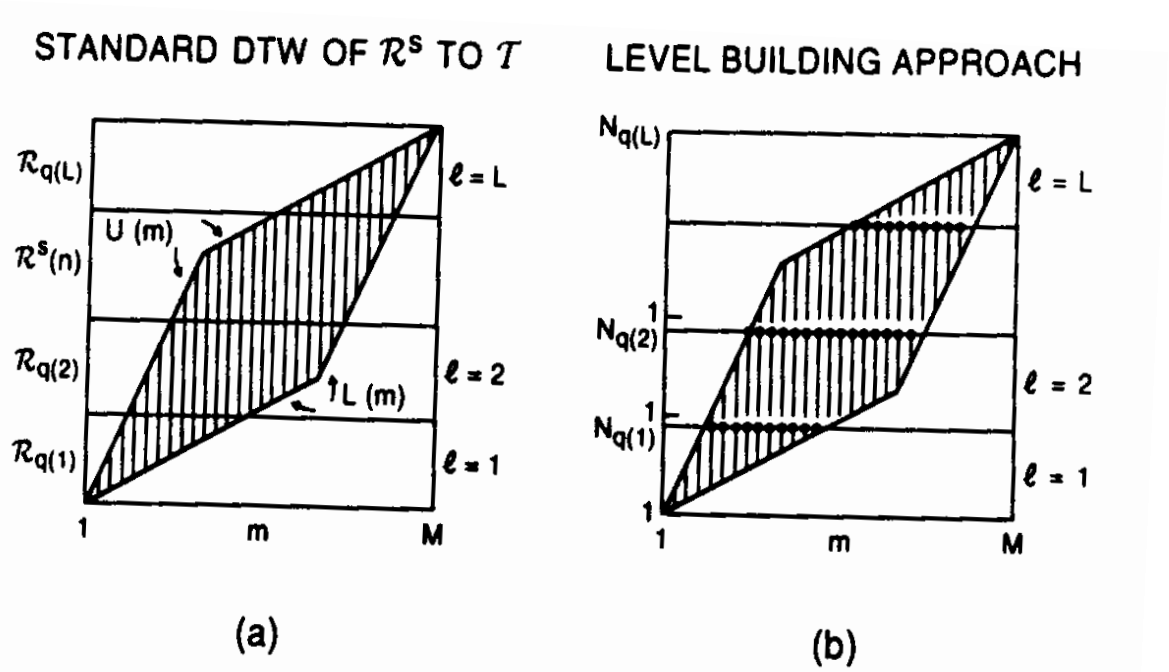


Figure 7.7 Illustration of standard DTW alignment of super-reference and test patterns (a), and level building alignment (b) (after Myers and Rabiner [4]).

Figure : 12.7 illutartion of standard DTW alignment of super-reference and test pattern

Usually the computation is performed in a frame synchronous manner that is the computation of given test frame , m is preformed and after that m+1, m+2 etc. until the last test frame are reached.

An alternate way of Aligning R^s and T is shown in figure b, here the computation for a given test frame m, is truncated at a fixed horizontal level frame. Iterating the search along vertical stripes corresponds tom region of intersection with second word . Similarly it is performed for all level until the complete alignment grid is covered.

In two level DP algorithm:

First level is calculation is of local warping distances for word candidates. For each test frame whose vertical stripe intersect with horizontal level corresponds to the end of the first word

Second level is searching for optimal word path

The difference between computation procedure a and b is minor.

The key difference between level building and two level DP algorithm is that partial word decisions are made during the DP search and this reduces the search range in later stages of Level building algorithm. While in two level DP, all the word scores until the end of the entire utterance and then decide the string in a separate and independent second level DP calculation.

For level building, we do exactly V time warps so that total computation is in order of V * L.

Since L is less than number of test frames, M, it involves less computations as compared to two level DP algorithm, which requires V * M time warp.

12.4.1 MATHEMATICS OF THE LEVEL BUILDING ALGORITHM

We define $D_v^l(m)$ as minimum accumulated distance at level l, for reference pattern R_v and to the frame m of test pattern.

Consider implementation of level l as shown in figure below:

IDOL

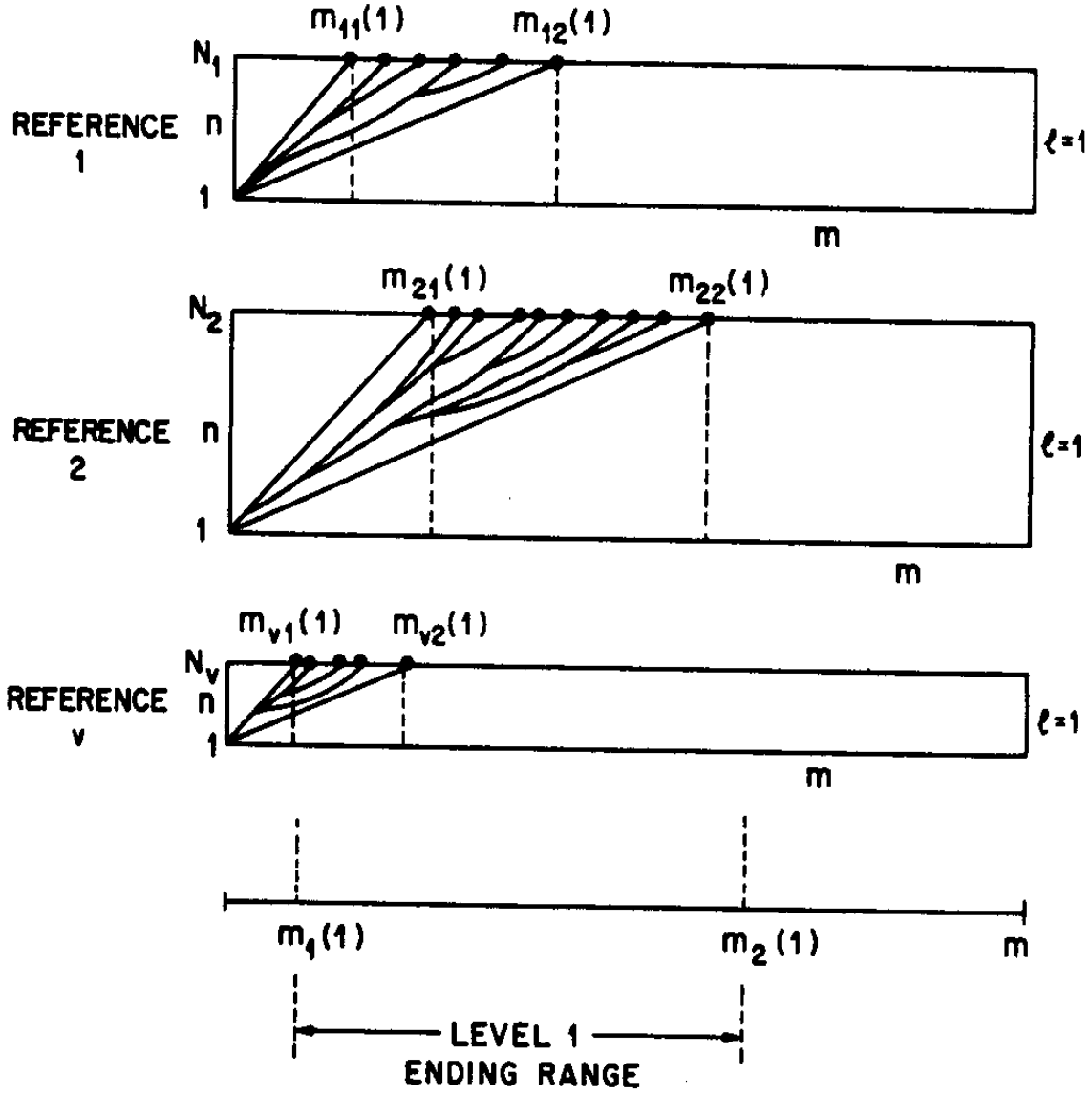


Figure 7.8 Implementation of level 1 of the level building algorithm (after Myers and Rabiner [4]).

Figure: 12.8 Implementation of level 1 of the level building algorithm

The first reference pattern R_1 is aligned with T beginning frame 1 of T using a DTW procedure.

The warping paths intersect the last frame in R_1 at range of test frames, namely $m_{l1}(1) < m < m_{l2}(1)$

F0r each frame we stores the accumulated distance $D_1^1(m)$.

Similarly, for second reference pattern R_1 with N_2 frames, begin at 1 of T and obtain warping paths to the range namely $m_{21}(1) < m < m_{22}(1)$.
This is iterated for V reference patterns at level 1.

The output obtained is array of accumulated distance and their ranges i.e.,

$$\begin{aligned}
&\bar{D}_1^1(m), \quad m_{11}(1) \le m \le m_{12}(1) \\
&\bar{D}_1^2(m), \quad m_{21}(1) \le m \le m_{22}(1) \\
&\vdots \\
&\bar{D}_1^V(m), \quad m_{V1}(1) \le m \le m_{V2}(1).
\end{aligned} \tag{7.15}$$

After that we can define the ending range of level 1, $m_1(1)<m<m_2(1)$, as a composite range of D_l^v is define as

$$m_1(1) = \min_{1 \le v \le V} [m_{v1}(1)] \tag{7.16}$$

$$m_2(1) = \max_{1 \le v \le V} [m_{v2}(1)] \tag{7.17}$$

For each frame, m, in the range $m_1(\ell) \le m \le m_2(\ell)$, ,
We need to store

$$\bar{D}_\ell^B(m) = \min_{1 \le v \le V} [\bar{D}_\ell^v(m)] \quad \text{– best distance at level } \ell \text{ to frame } m \tag{7.18}$$

$$\bar{N}_\ell^B(m) = \arg \min_{1 \le v \le V} [\bar{D}_\ell^v(m)] \quad \text{– reference pattern index which gave distance at level } \ell \text{ to frame } m \tag{7.19}$$

$$\bar{F}_\ell^B(m) = \bar{F}_\ell^{\bar{N}_\ell^B(m)}(m) \quad \text{– backpointer to best ending frame at previous level that achieves } \bar{D}_\ell^B(m) \tag{7.20}$$

This reduces the amount of storage required without losing the data.

Only after the first level computation is performed and completed, the second level computation starts.

Consider below figure, for the second level computation.

IDOL

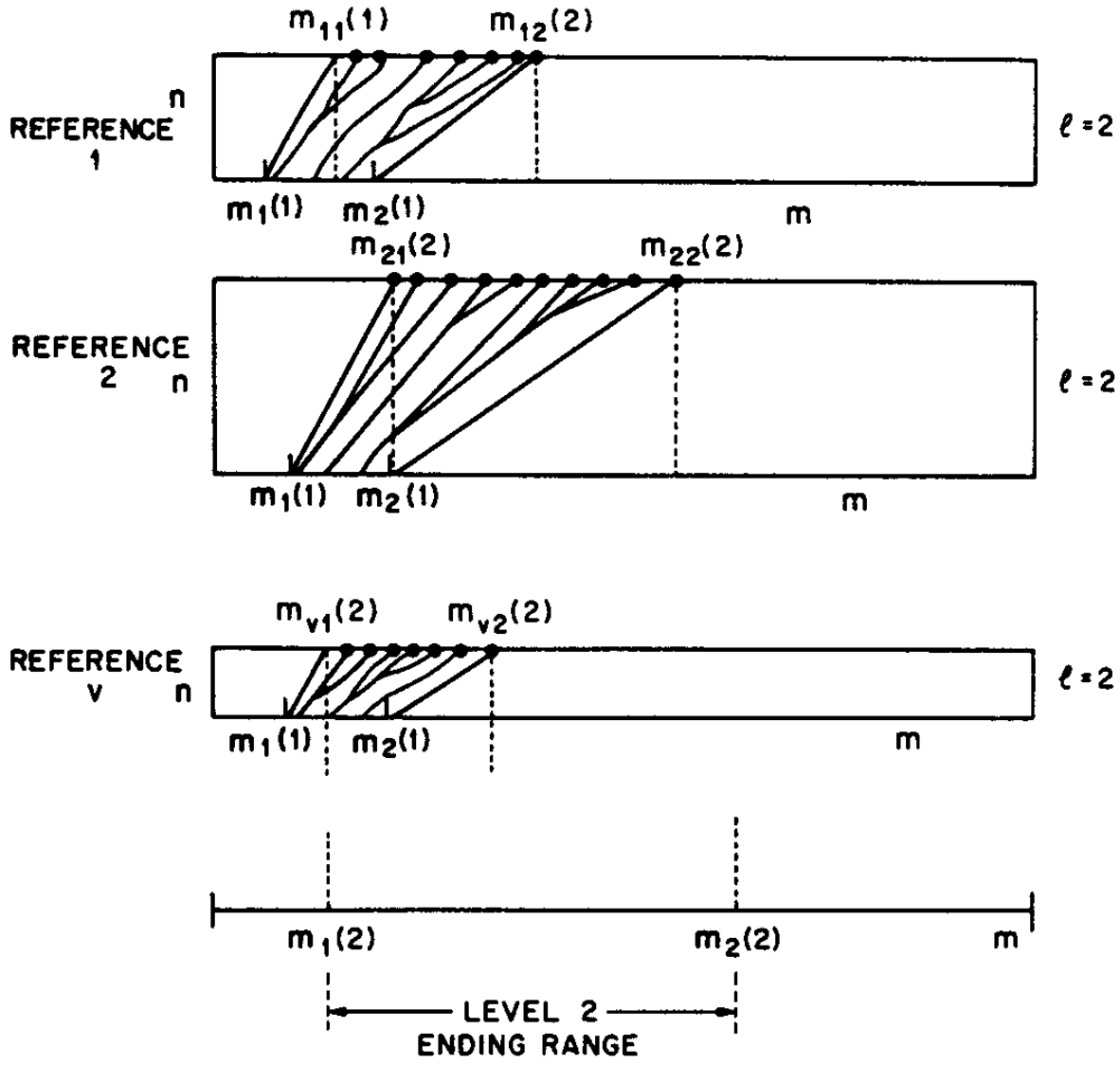


Figure 7.9 Implementation of level 2 of the level building algorithm (after Myers and Rabiner [4]).

For each reference pattern, R_v, the range of starting frame is m1(1)<m<m2(1) as the every ending point of first level, is the starting point of second level.

Except the broadened beginning region, each DTW is same at level 2 as of level 1.

Hence for R_1 the range of ending frame at level 2 is $m_{11}(2)<m<m_{12}(2)$ and so on..

So we derive the ending range at the end of level 2 as

$$m_1(2) = \min_{1 \le v \le V} [m_{v1}(2)] \quad (7.21)$$

$$m_2(2) = \max_{1 \le v \le V} [m_{v2}(2)] \quad (7.22)$$

Similarly we also define the best Distance $\bar{D}_2^B(m)$, the reference pattern $\bar{N}_2^B(m)$, and the back pointer $\bar{F}_2^B(m)$.

We can continue this level building procedure for all the level, $L_{max.}$

Once all the level building has been done, we get the final solution as ,

$$D^* = \min_{1 \le \ell \le L_{max}} [\bar{D}_\ell^B(m)]. \quad (7.23)$$

A drawback of level building algorithm is level synchronous and not time synchronous, so we can go back to the test frame at many levels. Due to which real time implementation of this is difficult.

For a better understanding of level building algorithm, let take one example, as shown in below figure.

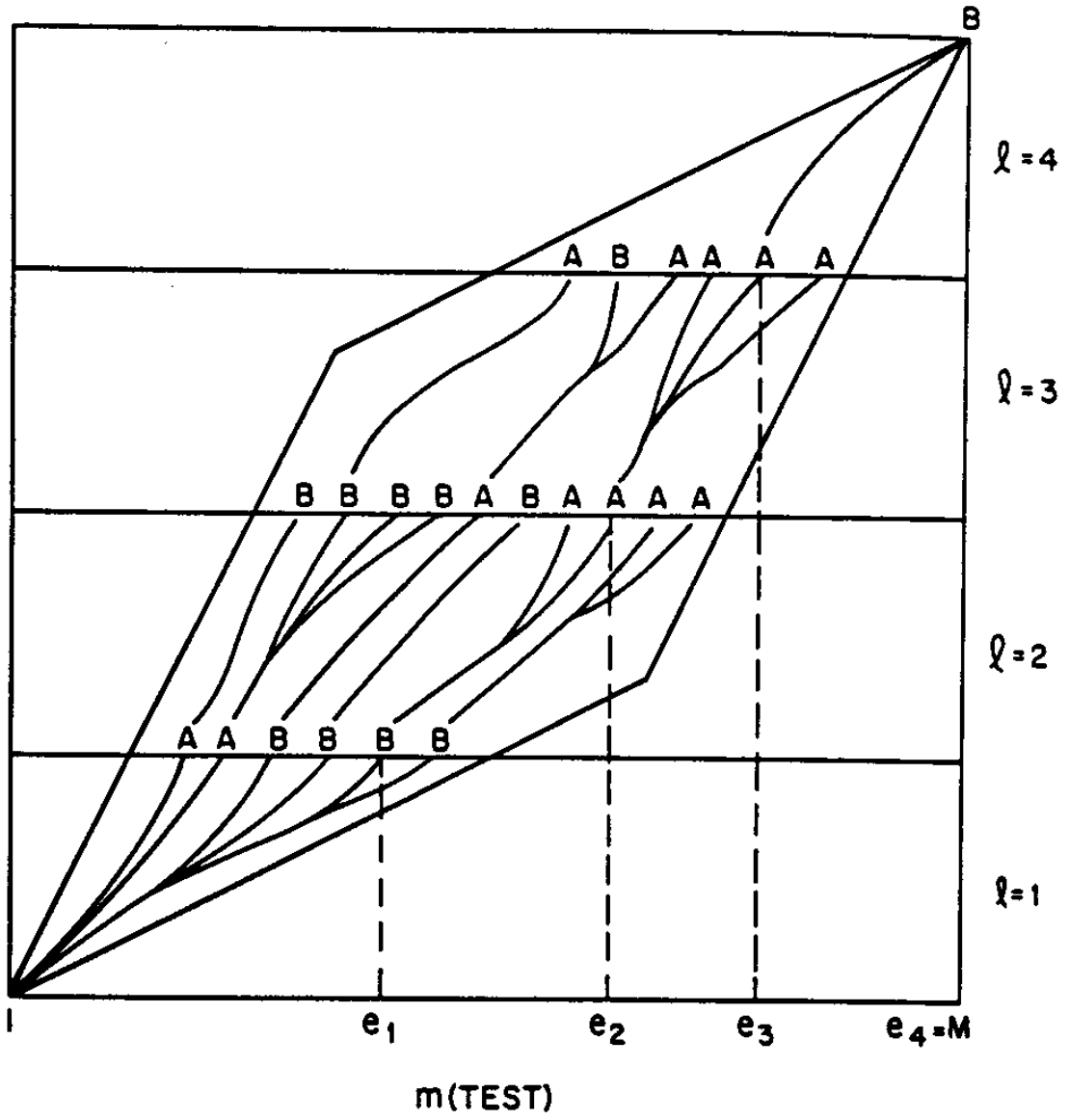


Figure 7.10 Simple example illustrating level building on two reference patterns of equal length (after Myers and Rabiner [4]).

Assume, vocabulary has two words, A and B. and two reference pattern R_A and R_B. of equal length.

Also assume, I =4.

As both words are of equal length, there ending regions are identical.

At each level ending region, we select reference pattern that give the smallest distance to that frame.

At level 1 there are 6 ending frames. (AABBBB)

At level 2 there are 10 ending frames. (BBBBABAAAA)

At level 3 there are 6 ending frames. (ABAAAA)

And finally at 4^{th} level there is only one ending frame M, the end of test utterance.

Here, the sequence of reference patters, providing the best score is,

$$\mathcal{R}^* = \mathcal{R}_B \oplus \mathcal{R}_A \oplus \mathcal{R}_A \oplus \mathcal{R}_B$$

12.4.2 MULTIPLE LEVEL CONSIDERATIONS:

Due to the multiple levels, the computations go on increasing. There are many simple techniques to reduce the unnecessary computations.

Consider standard warping range of level building algorithm in below figure:

IDOL

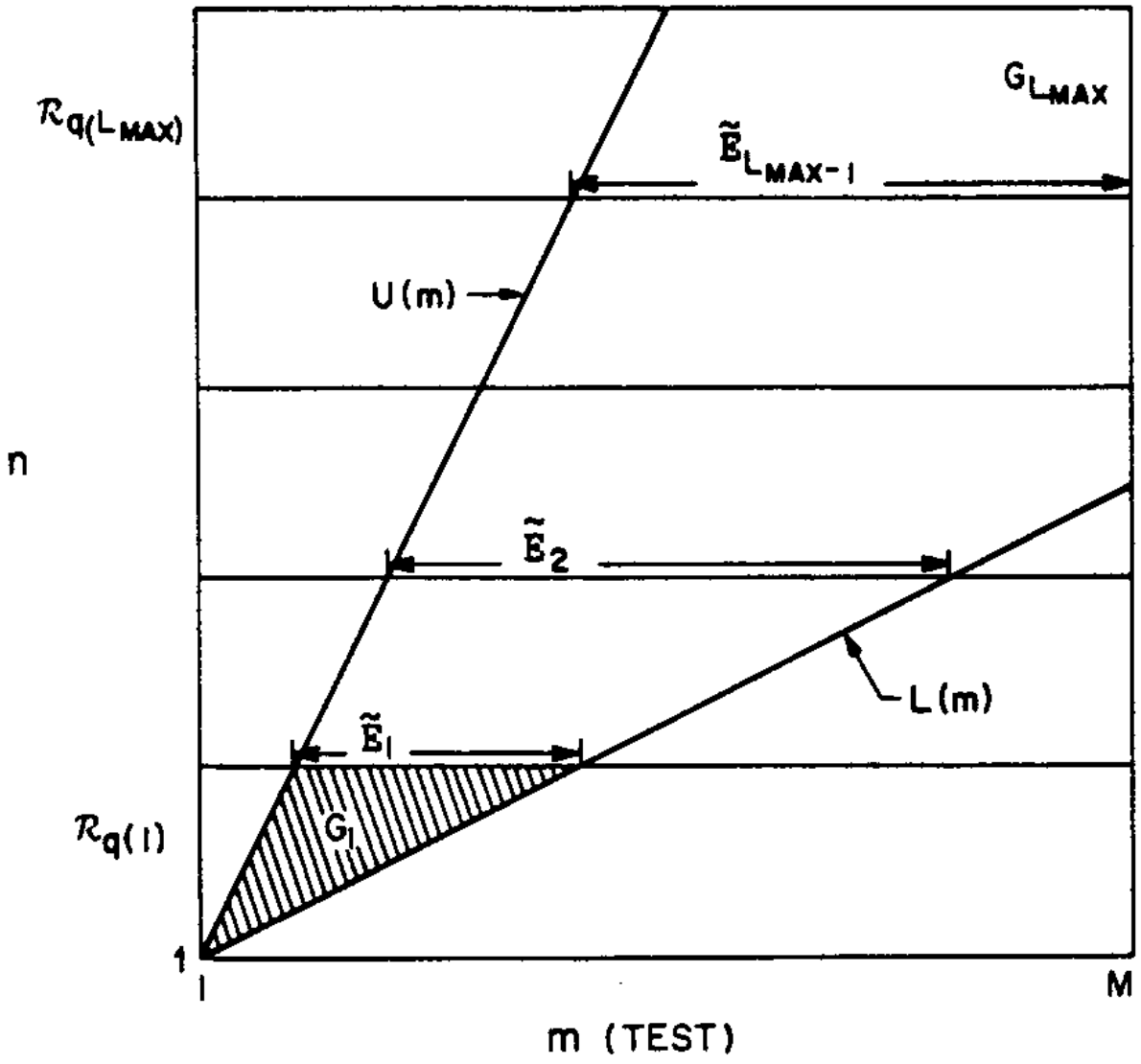


Figure 7.11 Computation region of level building algorithm for a fixed-length reference pattern (after Myers and Rabiner [4]).

The lower and upper constraint lines can be shown as

$$L(m) = (m + 1)/2 \tag{7.24}$$

$$U(m) = 2(m - 1) + 1 \tag{7.25}$$

For a fixed length reference pattern, we can compute the ending region as shown in the above figure.

As you can see, there are few points which are computed unnecessarily, and they won be needed for the best path calculations.

Hence we can add a constraint that, if a grid point is cab able of reaching the end of reference patterns, then only it is computed else not. This is shown in the figure below:

IDOL

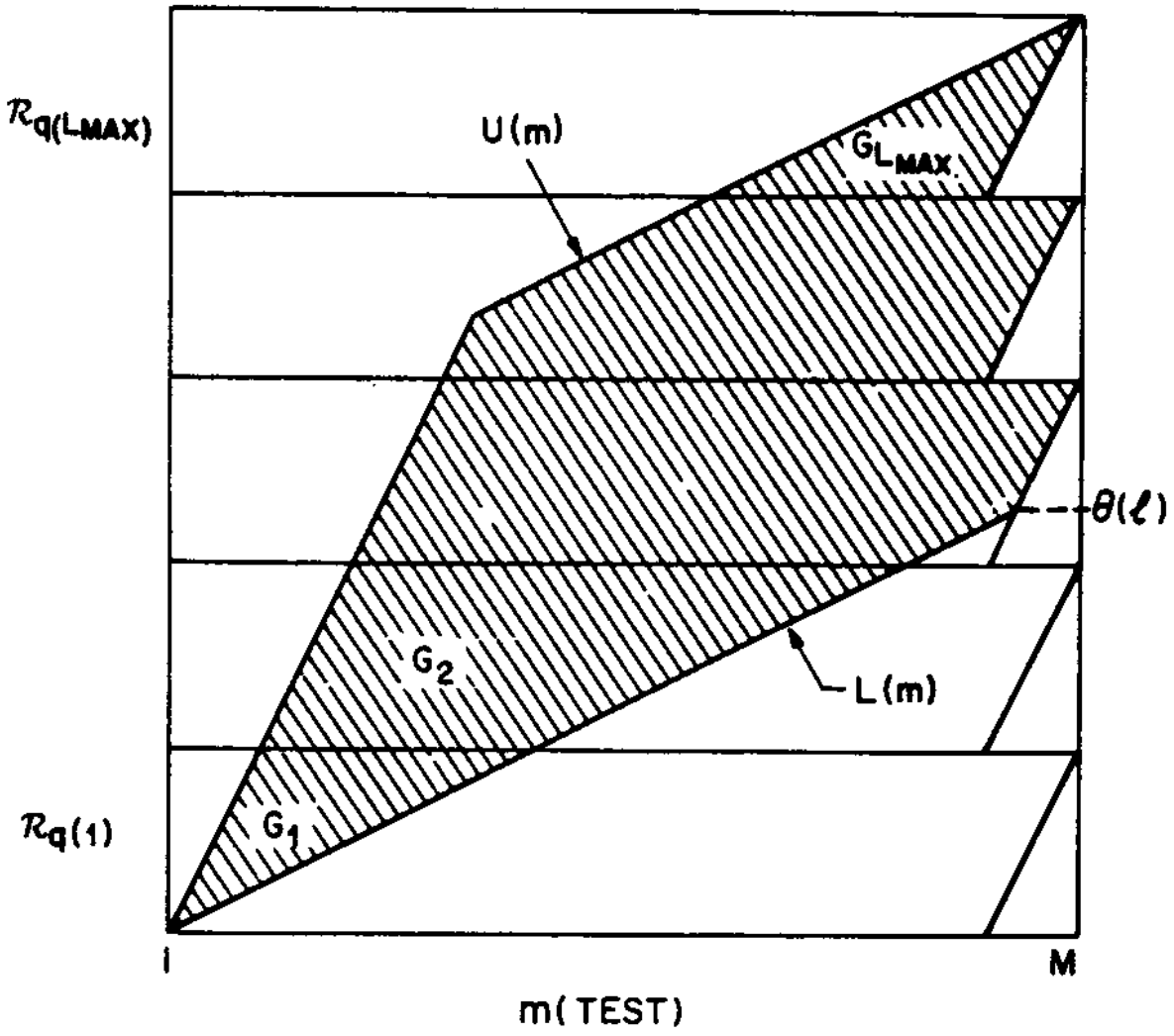


Figure 7.12 Reduced computation region using upper- and lower-level constraints (after Myers and Rabiner [4]).

In the figure, lines are drawn at each level of slope 2.

There is also a line drawn at level Lmax, of slope1/2.

Now, the lower and upper constraint can be defined as:

$$L(m) = \max\left[\frac{m+1}{2}, 2(m-M) + \theta(\ell)\right] \quad (7.26)$$

$$U(m) = \min\left[2(m-1), \frac{1}{2}(m-M) + \theta(L_{\max})\right] \quad (7.27)$$

$\theta(\ell)$. Is the accumulated number of frames of the reference patterns used up to level l where: $(m+1)/2 = 2(m-M) + \theta(\ell)$

Due to which, the resulting region, is reduced to some extent, as compared to the region showed in the 7.11 figure.

The length of each reference pattern is different in general, its actual computation regions for LB search is shown below:

It shows the computation regions for lowest shortest and longest reference patterns

IDOL

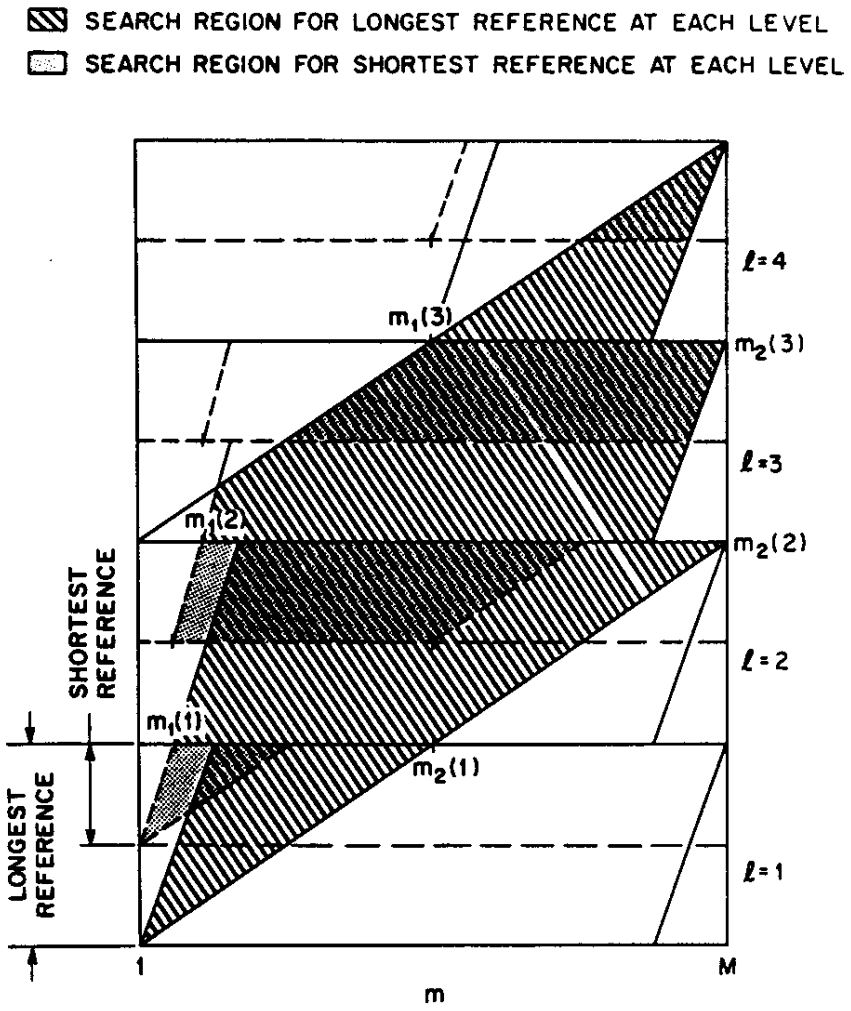


Figure 7.13 Overall computation pattern of level building algorithm for variable length reference patterns (after Myers and Rabiner [4]).

12.4.3 COMPUTATION OF THE LEVEL BUILDING ALGORITHM

The computation of level building algorithm is a series of V time wraps at each level where V is the size of vocabulary.

If we assume maximum number of levels in L_{max} then we need VL_{max} time warps

Overbound on size of each time warp is NM/3 grid points where N is average length of each reference pattern and M is number of frames in test pattern

Hence total computation of level building algorithm is

$C_{LB} = V.L_{max}.\overline{N}.M/3$ grid points

With storage

$S_{LB}=3M.L_{max}$

Since we need to storage for $\bar{D}^B, \bar{N}^B and\ \bar{F}^B$ at each value of m and for each level l

The basic computation of level building algorithm is factor of 4.7 less than that of the two level DP method; the storage of both method is same

12.4.4 Implementation aspect of level building

There are several ways from which can reduce the computation load of the algorithm

Beginning range reduction

For the complete level building algorithm at level $l-1$ we retain the best score $\bar{D}^B_{l-1}$ for each frame m in the ending region $m_1(l-1) \le m \le m_2(l-1)$. The best global path is not at boundary but somewhere in middle range of m we eliminate some of ending range at level $l-1$, now search at level l requires less computation

To reduce the ending range at level $l-1$ we need to normalize the best accumulated distance score by number of frames.

Find locally minimum normalized score as

$$\phi_{\ell-1} = \min_{m_1(\ell-1)\le m\le m_2(\ell-1)} \left[\frac{\bar{D}^B_{\ell-1}(m)}{m}\right].$$

We now define a reduce range level threshold as $M_T\ \phi_{l-1}$ where M_T is a defined parameter an search the range $m_1(l-1) \le m \le m_2(l-1)$. To find indices $S_l^1 and\ S_l^2$ such that

$$S_\ell^1 = \arg \max_{m_1(\ell-1)\le m\le m_2(\ell-1)} \left[\frac{\bar{D}^B_{\ell-1}(m)}{m} > M_T\cdot\phi_{\ell-1} \quad \forall\, m \le S_\ell^1\right]$$
$$S_\ell^2 = \arg \max_{m_1(\ell-1)\le m\le m_2(\ell-1)} \left[\frac{\bar{D}^B_{\ell-1}(m)}{m} > M_T\cdot\phi_{\ell-1} \quad \forall\, m \ge S_\ell^2\right].$$

Too small value of M_T will allow the best path to be prematurely eliminated hence proper choice of M_T is essential

2. Global range reduction - ϵ

The idea is reduce search range along the reference axis, by tracking the global minimum and allowing only a range around global minimum

Thus for each test frame m, at each level, for each reference pattern R_v, we determine local minimum c(m) as

$$c(m) = \arg \min_{c(m-1)-\epsilon \le n \le c(m-1)+\epsilon} [D_l^v(m-1, n)]$$

In which $D_l^v(m-1, n)$ is defined to be the best distance at level l using reference R_v at test frame m-1 to reference frame n and with C(1) defined to be 1

3. Test pattern ending range - δ_{END}

The idea is to allow a range of test pattern ending frames rather than restricting the ending frame to m=M.

If we extend the end of the test pattern by δ_{END} frames the global level building solution modified to be

$$D^* = \min_{1 \le l \le L_{max}} \min_{M-\delta_{END} \le m \le M} [\bar{D}_l^B(m)].$$

4. Reference pattern uncertainty regions - $\delta_{R_1}, \delta_{R_2}$

For coarticulation of words across word boundaries the level building algorithm allows a range of beginning and ending frames of reference pattern.

At any level the path can begin over the range 1≤ n≤ δ_{R_1} and end at any frame in the range

N_v -δ_{R_2}≤n≤N_v

For appropriate values of $\delta_{R_1}, \delta_{R_2}$possible to have a path which skips $(\delta_{R_1} + \delta_{R_2})$ frames at highly coarticulated word boundries

IDOL

It can be shown that with proper choice of the implementation parameter, M_T , ϵ , δ_{END}, δ_{R_1} and $\delta_{R_2}$the overall computation can be reduced from the standard level building approach

12.4.5 Integration of grammar network

In implicit from of grammar each word in string can be followed by any other word in string

For some connected word recognition task there is an explicit of grammar governing which word can logically follow other words to form a valid sentence in language

We can represent the grammar by a finite state network (FSN) of the form

G =A(Q,V,δ,q_0,Z)

Where

Q = set of states

V = set of vocabulary words

δ = set of transition

$q_0 \epsilon Q$ = initial state

$Z \subseteq Q$ = set of terminal states

And the set of transition obeys the rule

$\delta(q, v) = s$

Meaning that word v drives from q to s

To integrate the FSN Grammar network into the following level building algorithm we must do following

1. Identify levels with states rather than word postion so that word candidate at the level l^{th} need not be temporally contiguous to those at level $(l + 1)$ level
2. Partition the vocabulary so that only reference patterns for words leaving the l^{th} state are mathched at the l^{th} level
3. Retain state backtracking pointers for recovering the best matching string.

IDOL

References and further Reading

[1] L.R. Rabiner , B. H. Juang , “Fundamentals of Speech Recognition”, Prentice-Hall, Englewood Cliffs, N.J.

IDOL

CHAPTER 13

SPEECH RECOGNITION BASED ON CONNECTED WORDS MODELS PART-B

13.0 Objectives

- The one pass algorithm
- Multiple candidate strings
- Grammar networks for connected digit recognition
- Segmental K-Means training procedure
- Connected digit recognition implementation

13.1 THE ONE PASS (ONE STATE ALGORITHM)

Another approach for the connected word recognition is called One Pass procedure or one state algorithm or frame synchronous level building.

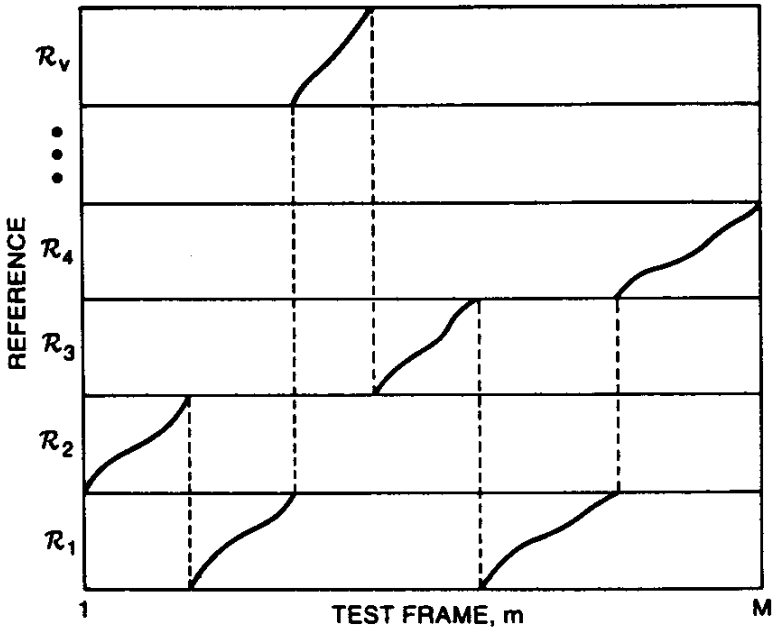


Figure 7.20 The one-pass connected word recognition algorithm (after Bridle et al. [6]).

Figure 13.1 : one –pass connected word recognition algorithm

In above figure, as you can see, horizontal axis has test patterns and vertical axes have reference patterns.

Accumulated distance is calculated as,

$d_A(m, n, v)$ as:

$$d_A(m, n, v) = d(m, n, v) + \min_{n-2 \le j \le n} (d_A(m-1, j, v)). \qquad (7.37)$$

m is test frame index

v is the reference pattern index

n is reference frame index of R_v.

d(m, n, v) is the distance between the test frame t(m) and the reference $R_v(n)$.

After this the recursion is performed of the above given equation, as

$$d_A(m, 1, v) = d(m, 1, v) + \min \left[\min_{1 \le r \le V} [d_A(m-1, N_r, r)], d_A(m-1, 1, v) \right]. \qquad (7.38)$$

The combinatory for internal and boundary frames are displayed in below figure:

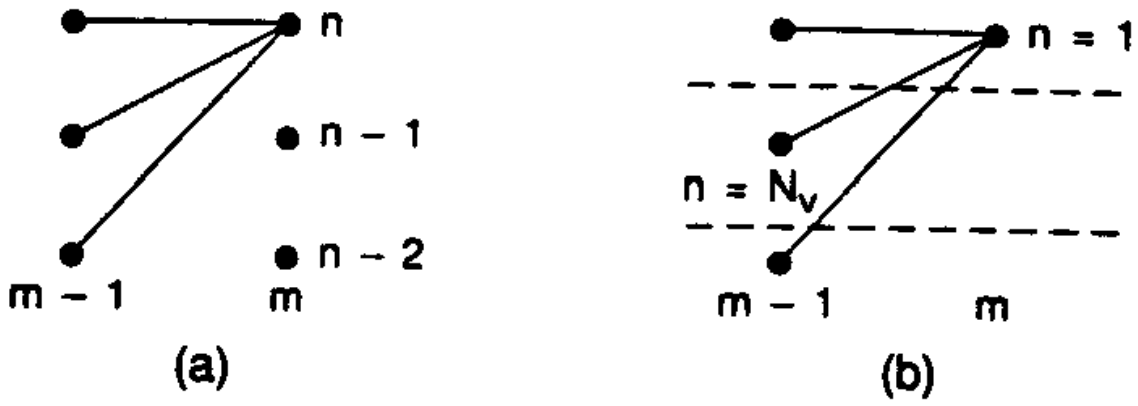


Figure 7.21 Combinatorics for the one-pass algorithm.

Figure 13.2 Combinatorics for the one-pass algorithm

Figure a show the internal frames selects the best internal path within reference pattern. Boundary frames chooses either a straight within a reference pattern.

The final solution for best path is

$$D^* = \min_{1 \le v \le V} [d_A(M, N_v, v)]. \qquad (7.39)$$

This algorithm calculates the best path to every reference pattern for every test frame. It backtrack the best score to provide the best word sequence.

A problem with the one pass algorithm is there is no way to control the length of resulting string The algorithm just finds the best path, it does not considers the constraints of the string length.

For adding the length constraint, we extend the accumulated distance to include the level information.

The recursion is computed with the level information as,

$$d_A^{\ell}(m, n, v) = d(m, n, v) + \min_{n-2 \le j \le n} [d_A^{\ell}(m-1, j, v)] \qquad (7.40)$$

At each boundary, now the computation will look like,

$$d_A^{\ell}(m, 1, v) = d(m, 1, v) + \min \left[\min_{1 \le r \le V} d_A^{\ell-1}(m-1, N_v, r), d_A^{\ell}(m-1, 1, v) \right] \qquad (7.41)$$

With

$$D^* = \min_{1 \le \ell \le L_{\max}} \min_{1 \le v \le V} [d_A^{\ell}(M, N_v, v)]. \qquad (7.42)$$

A good point about one pass algorithm is that computation for the frame m can be done in synchronous way, due to which it is suitable for processor in real time implementation, where processors can compute all the necessary calculations in a single frame interval.

Also, the computation of d(m,n,v) is independent of level, and hence is computed only once and stored for use in every level computation.

13.2 MULTIPLE CANDIDATE STRINGS

Until now, we were searching for the most suitable staring for the spoken word, we can also determine, the second best match, third best match string for the spoken word.

This multiple candidates for the recognition of the string are useful in the grammar network.

For finding out the multiple candidates for the string, we can keep track of best distance as well as the second best distance at each level of the frame. This would require additional storage for the second match string distance.

Following figure helps to illustrate the two level LB search for keeping track of two best matching strings.

As shown in the figure, there are two distinct paths (solid lines) for the best path string. The dashed line shows the path of the second best string.

At the end, the best path of second best string and first best string branches off.

IDOL

Now there are total four best paths,

11 (best from level 2 and best from level 1)

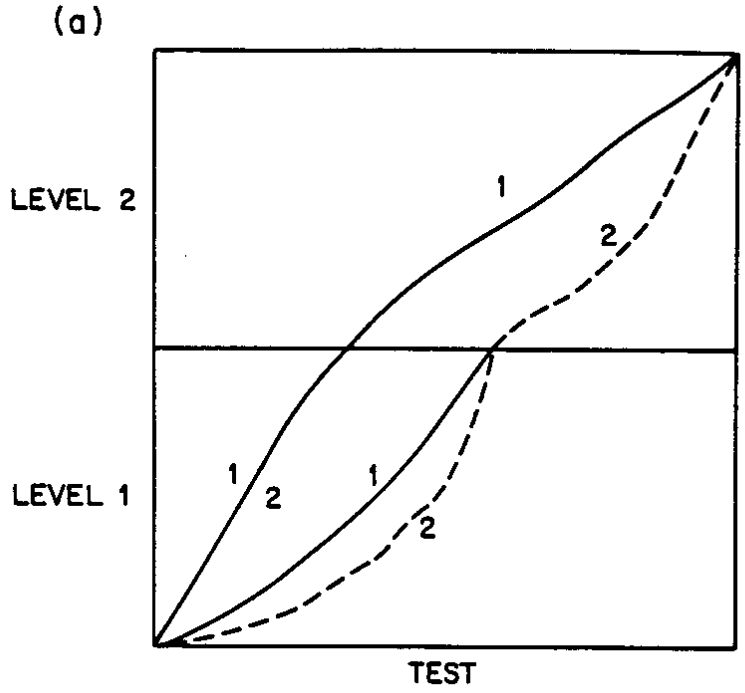
12 (best from level 2 and second best from level 1)

21 (second best from level 2 and best from level 1)

22 (second best from level 2 and second best from level 1)

The overall best path is of course 11 path

And the second best path is either 12 or 21.



This above procedure can be extended to find the best three candidate of the spoken word.

It would require finding the best three candidates at each level. After L levels there would be 3^L scores would be obtained. Ti9s is showed in the below figure.

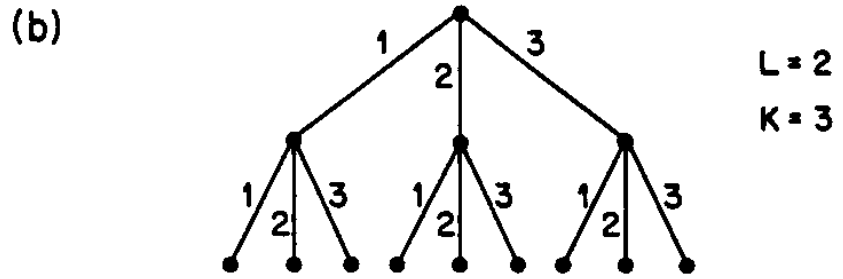


Figure 7.22 Description of procedure for determining multiple candidate scores (after Myers and Rabiner [4]).

Figure 13.3 : Description of procedure for determining multiple candidate scores Here for two levels match there would be nine strings, as $3^2 = 9$.

In the below figure a gain, for L=4 level, with two best candidates. There are four possible choices for the second best string candidate.

Those four possibilities of path are 1112, 1121, 1211 and 2111 paths.

IDOL

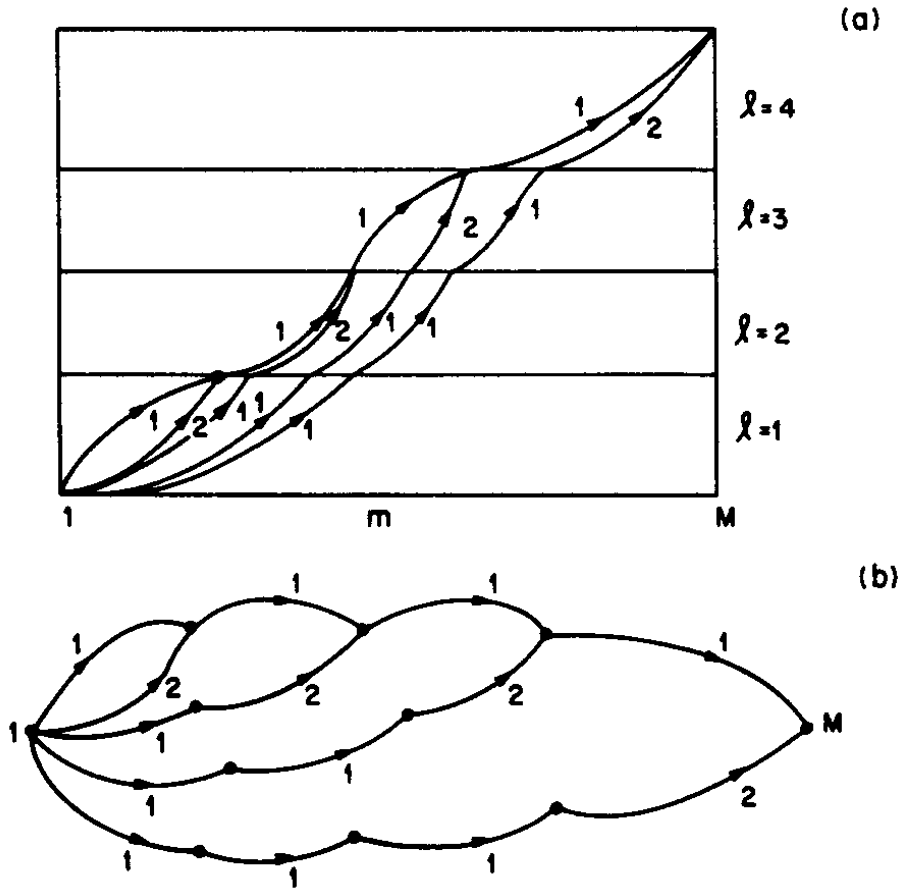


Figure 7.23 Candidate strings for a four-level search (after Myers and Rabiner [4]).

Figure : 13.4 . Candidate strings for a four-level search

This procedures uses level building algorithm for finding out the second best string, and it does not guarantees that the obtained second best string is truly the second best string, because in keeping track of multiple strings to any ending frame, the procedure requires to the candidate string to coke from different reference pattern.

Following figure shows the flaw of level building algorithm. Here the second best string BA (reference pattern B followed by A).

And since AA is the best string candidate, A is not allowed as last candidate of the last frame for level 2.

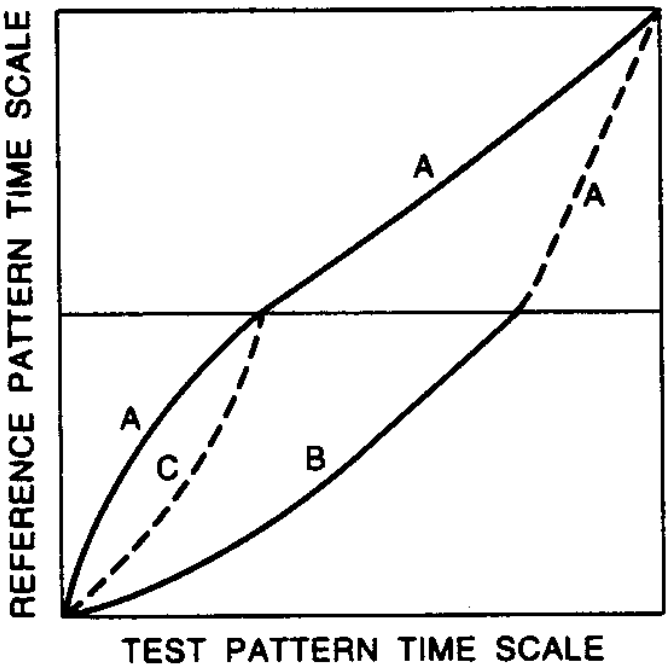


Figure 7.24 Illustration of level building flaw for determining the second-best candidate string (after Myers and Rabiner [10]).

Figure 13.5: illustration of level building flaw for determining the second-best candidate string

13.3SUMMARY OF THE COONECTED RECOGNITION ALGORITHMS

Till now, we have understood three approaches for connected word algorithm.

1. Two level DP algorithm
2. LB method
3. One pass algorithm

All these algorithms provide the same best matching score for the best matching string. Although they differ in computational efficiency, storage requirement and ease of realization.

All these three algorithm are almost amenable for integration of FSN grammar network for word sequence.

Consider a FSN grammar network denoted as g in below figure. Input to the node are various word arc corresponds to words in vocabulary from previous grammar node i-1, i+1. At g, now the maximum likelihood path over P(g) is computed. The node g, then propagate the best path to the next grammar node, j-1 and j+1.

The grammar node computation is iterated over all nodes in the grammar in an organized pattern

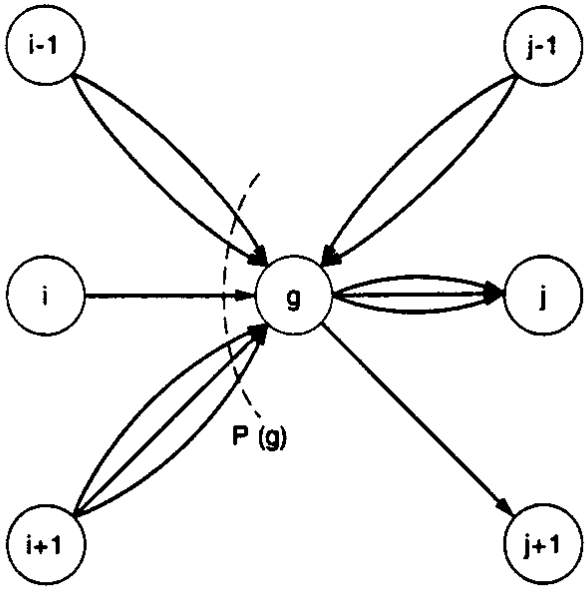


Figure 7.26 A typical grammar node of an FSN grammar network (after Lee and Rabiner [9]).

Figure: 13.6 A typical grammar node of an FSN grammar network

This FSN is integrated into the connected word algorithm, which is shown in below figure. For every spoken word, a spectral vector is computed.

Then the maximum likelihood distances are computed for every reference pattern state.

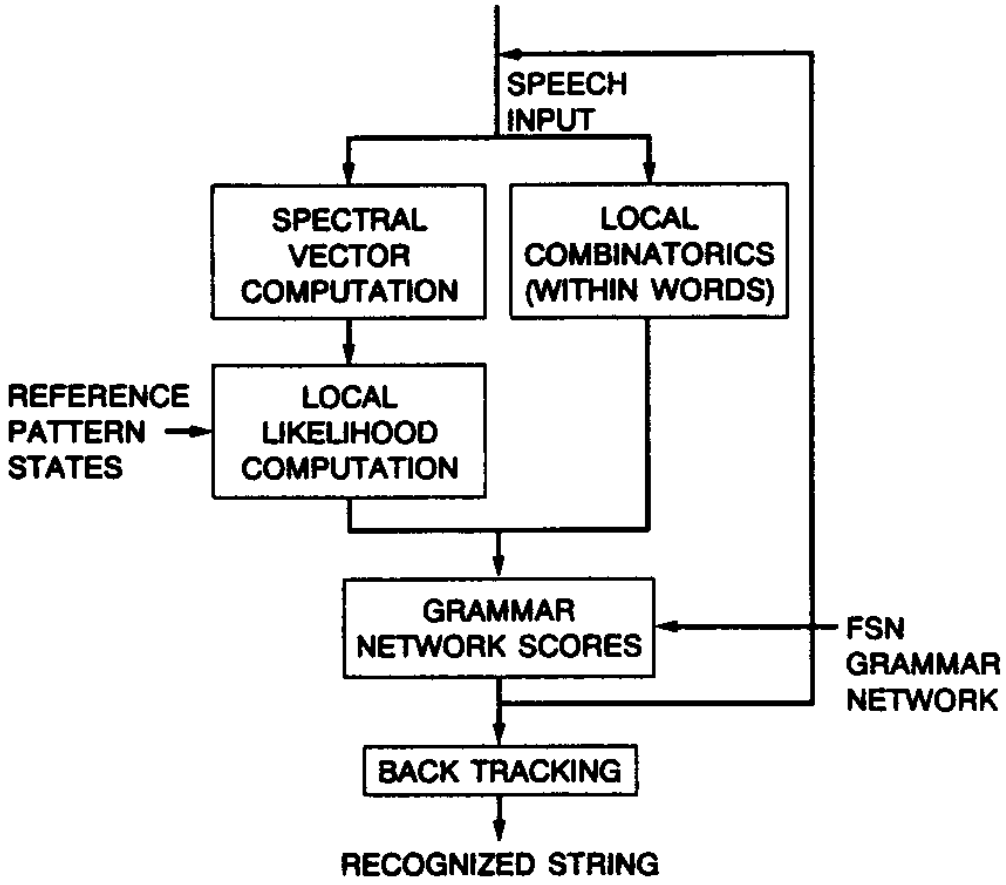


Figure 7.27 Block diagram of connected word recognition computation.

Figure 13.7 : Block diagram of connected word recognition computation

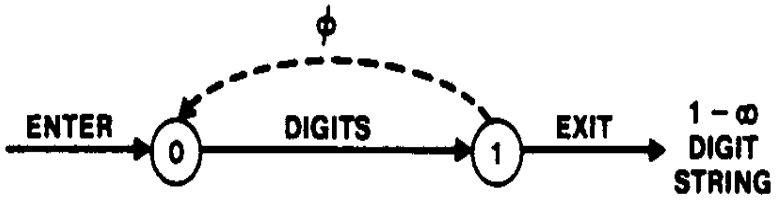
With parallel to that, a local comabinatorics are computed within the reference patterns. After that grammar network scores are computed . This entire process continues for all the test frames, and a backtracking method is used for selecting the best matching string.

13.4 GRAMMER NETWORKS FOR COONECTED DIGIT RECOGNITION

Connected digit recognition is a very important application of connected word recognition. Connected digit recognition can be useful in credit card entry, all digit dialing of telephone number, personal identification entry etc. and so on.

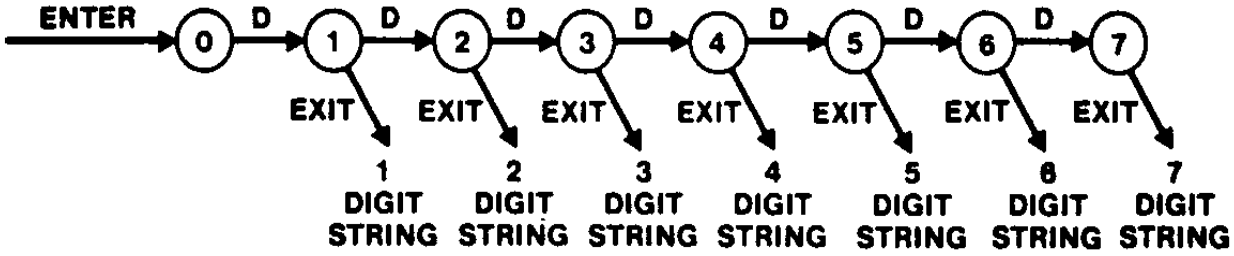
Let us understand the connected word recognition. Following figure shows three grammar network for digit recognition..

In grammar network 1 , is simple one pass algorithm without level information.



It finds the best match string, but it cannot put a constraint on the length of string. This will be unusable for applications for known length tasks such as selecting best 10 digit string for a mobile number.

The network of following figure, it actually breaks down the digit strings into 7 digits, and after that the usual level building approach is followed., with L=7.



Here the local combinatorics search requires 7 times of grammar network 1 and rest local distance calculations are same.

The network 3 of following figure, is a combination of network 1 and 2, where it breaks the string into three distinct level which are $3n_1+1$, $3n_2+2$ and $3n_3+3$ digits where n_1, n_2, and n_3 are arbitrary integers.

Here, the input is n digit string, the most likely errors are single digit insertion or deletion, and will be the correct string. Computation required are 3 times of network 1, however same local distance calculations,

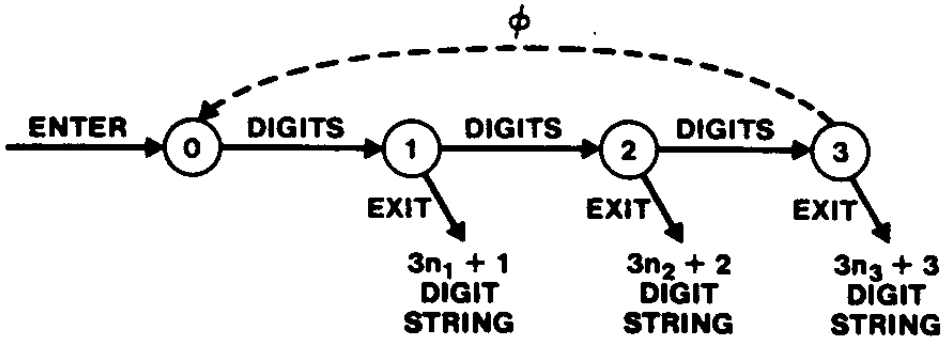


Figure 7.28 Three possible grammar networks for connected digit recognition (after Lee and Rabiner [9]).

Figure 13.8 : Three possible grammar networks for connected digit recognition

13.5 SEGMENTAL K-MEANS TRAINING PROCEDURE

As we have seen the connected digit recognition procedure, one major problem with those procedures is of segmenting the string into separate strings.
We can manually segment the string into individually spoken word of each string. But this process is error prone as the exact boundaries of connected word is not consistent. Hence now we need an automatic procedure for segmenting the string of connected word into individual words and the perform the training from the segmented strings.

We will understand one such algorithm of segmenting a connected word string into individual digits, called segmental K means training procedure.

This algorithm is a variation of well-known k means iteration.

Basically, it has a training set of labeled connected digit strings and initial isolated (individual) set of digit models.

The procedure of Segmental k means training algorithm is as follows:

1. Given the set of word patterns and the training files, we can use any connected word recognition procedure for segmenting each training string into isolated strings. And then those are stored in files accordingly to identify the digits.

2. Each file of isolated word, is segmented into states say, all the files for the word string 1s. Within each state the parameters of mixture density are determined using Vector Quantization clustering procedure. Resulting this would be an updated set of word models.

3. A test for convergence is made, based on testing files or likelihood scores of the training set files. If that convergence shows improvement the procedure is iterated again, I.e. step 1 and 2 are repeated with updated model. Else the procedure is terminated and the final updated models are the final set of models considered as the isolated strings of the provided connected word.

This procedure can be explained through the following diagram,.

IDOL

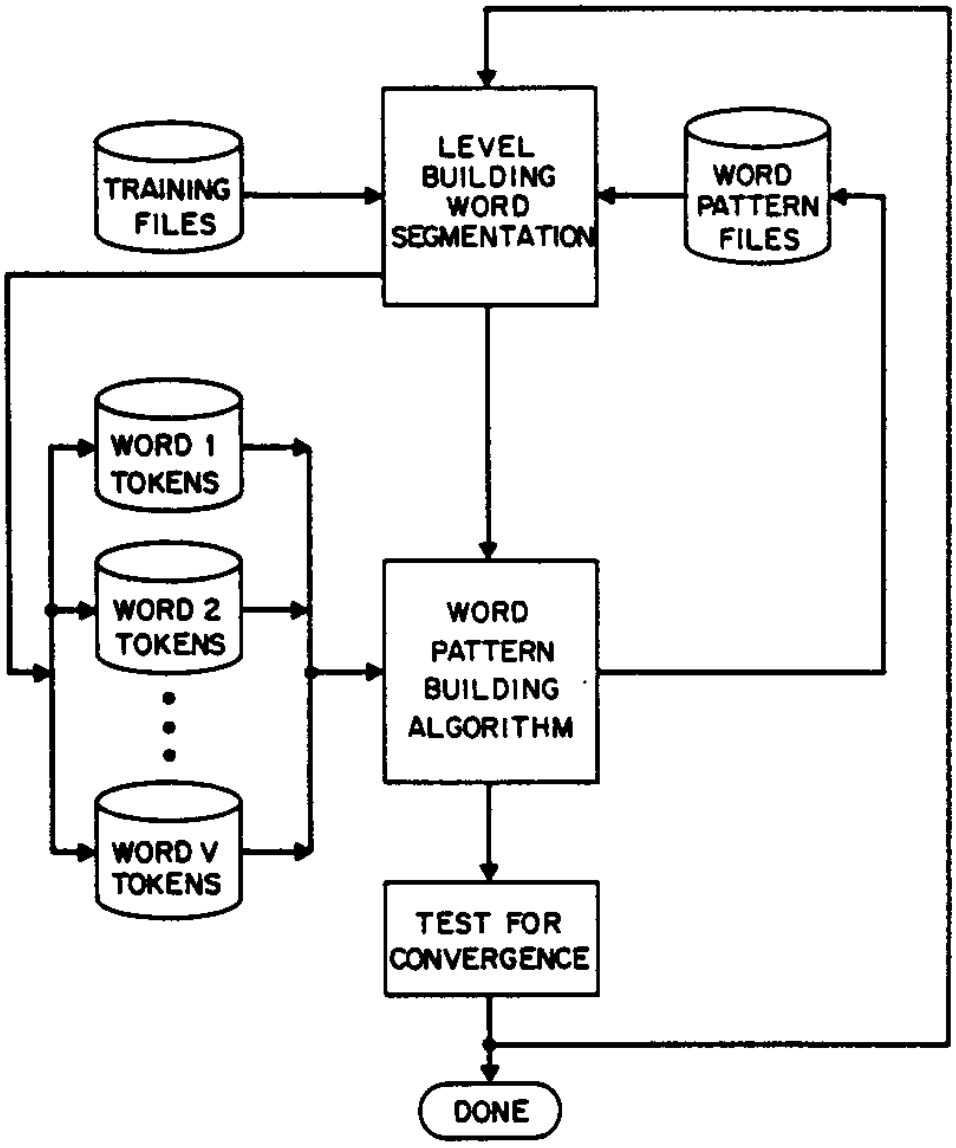


Figure 7.29 The segmental *k*-means training algorithm for connected word strings (after Rabiner et al. [13]).

Figure 13.9 : The segmental k-means training algorithm for connected word strings

13.6 CONNECTED DIGIT RECOGNITION IMPLEMENTATION

Black diagram of a system for connected digit recognition is shown in below figure:

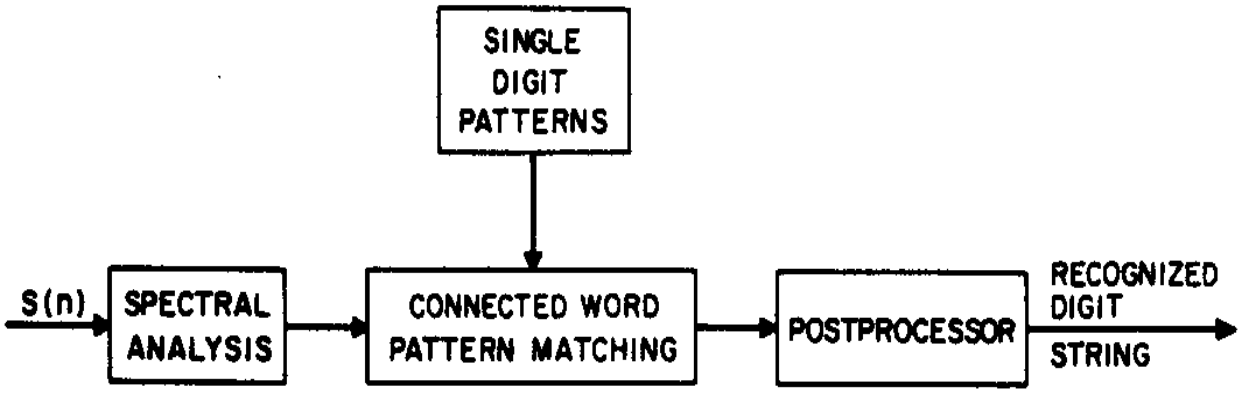


Figure 7.30 Block diagram of connected digit recognition method (after Rabiner et al. [13]).

Figure 13.10 . Block diagram of connected digit recognition method.

There are three steps in above process:

1. Spectral Analysis: Here, s(n), speech signal is converted into spectral representation using filter bank vector, LPC based vector
2. Connected Word Pattern Matching: Next, the spectral vector of test pattern i.e. connected digit string is compared with the single digit pattern using any one of the connected word algorithm disused above. And the output will be set of candidate strings of different length, displayed according to the best match score.
3. Post-processing: here, output of previous step, i.e. candidate strings are further processed to remove unlikely candidates from the candidate sets. This process chooses the most likely digit string.

13.6.1 HMM BASED SYSTEM FOR CONNECTED DIGIT RECOGNITION

Highest string accuracy can be achieved using based on LPC Cepstral analysis and HMMs.

LPC Analysis uses following characteristic:
Sampling rate – 6.67 kHz
Analysis window size – 300 samples
Analysis window shift – 100 samples
LPC Order - 8
Cepstrum Order – 12
Delta –delta cepstrum order – 12
Cepstral window – raised, sinelike window

In Hidden Markov Model, for each digit recognition uses left right models. HMM has following properties;

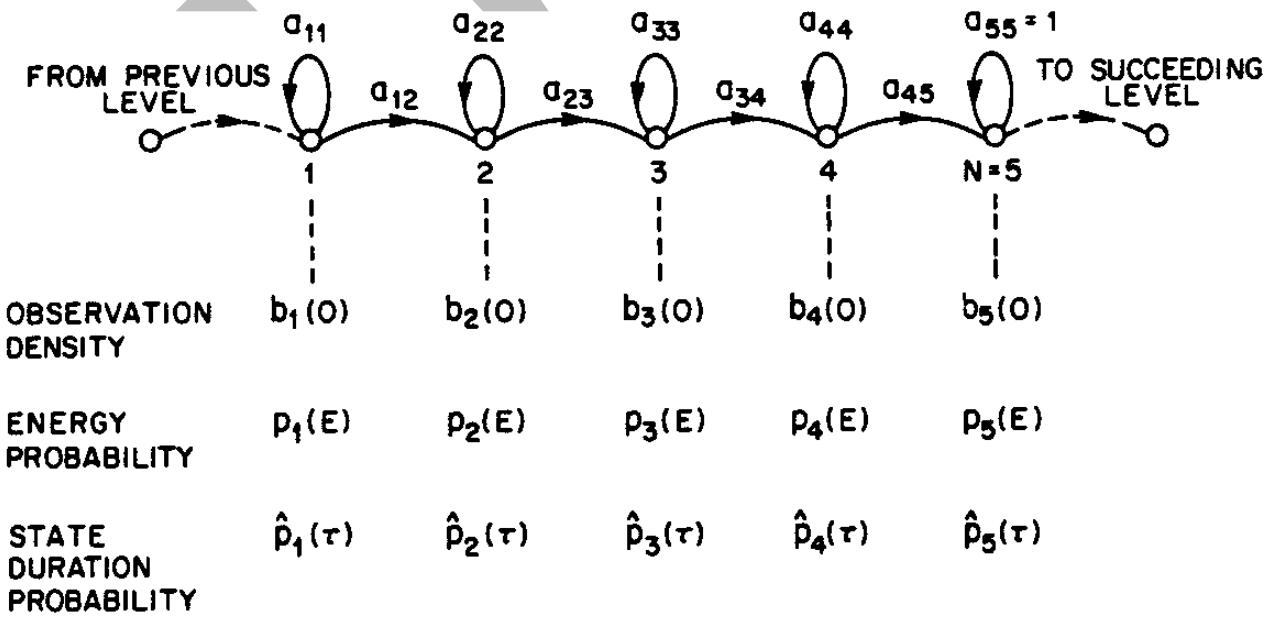


Figure 7.31 Connected digit HMM (after Rabiner et al. [13]).

Figure 13.11 Connected digit HMM

States, N = 5 to 10 for different digits
Number of mixture components per state are minimum 3 to maximum 64
A log energy probability density are used within each state
For post processing, a single Gaussian digit duration density is used based on the mean duration and variance.

13.6.2 PERFORMANCE EVALUATION ON CONNECTED DIGIT SRINGS:

Evaluation of performance of connected digit recognition procedures, let us considers some set of databases.

DB50:
It consists of 50 talkers of which 25 are female and 25 are male.
Every speaker provided 600 to 1150 digit strings, which gets to total of almost 47336 strings.
They are recorded of local, dialed up and over telephone
Strings are variable in length, minimum 1 and maximum 7 lengths
No pauses were used for speaking the string digits

T1 Set:
It consists of, 112 speakers of which 55 are male and 57 are female, which are from 22 regional accents.
Every speaker spoke 77 connected digit strings of length 1-5 or 7 digits.
Input is in wideband but with low pass filter of telephone band.
Digits are selected randomly from the strings.
No pauses within most of the strings, but few of the strings have internal pauses.

DB50 dataset is used in case of speaker trained and multi speaker mode, while T1 set is used for speaker independent tests.

IDOL

Following figure shows the plot of average speaking rate of two data sets as a function of the number of digits, in the string.

As you can see, the rate of T1 set is bit less than the rate of DB50.

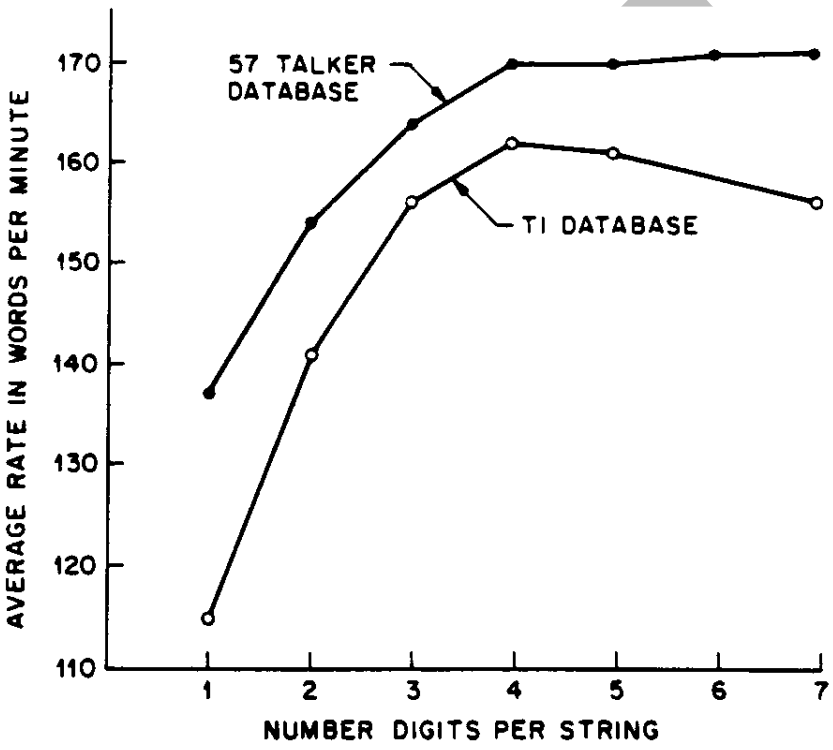


Figure 7.32 Average speaking rate of talkers in the two connected digit databases as a function of the number digits per string (after Rabiner et al. [13]).

Figure: 13.12 Average speaking rate of talkers in the two connected digit databases as a function of the number digits per string

Following conditions are used in performance evaluation of HMM when used for training:

1. Speaker- Trained Mode: For each speaker, half string of spoken word is used for training and other half is used for testing. Then a single HMM is used for a single digit.
2. Multi speaker Mode: Half training set and half testing set. One fourth of the training set is used for training 6 models per digits.
3. Speaker Independent Mode: Specified training and testing sets are used. Silence model is not considered for this mode. No pauses between the digits.

The performance of connected digit recognizer is shown in the below figure:

TABLE 7.1. Average String Error Rates (%) for Connected Digit Recognition Tests

Mode	Training Data		Testing Data	
	UL	KL	UL	KL
Speaker Trained	0.4	0.16	0.8	0.35
Multi-Speaker	1.7	1.0	2.85	1.65
Speaker Independent	0.3	0.05	1.4	0.8

Above stats shows the results for, performing testing on the training data itself and testing data provided. Furthermore, average string error rate for unknown length (UL) and known length (KL).

For known length the error rate is around, and for known length it is 0.4 to 1.7 lengths.

References and further Reading

[1] L.R. Rabiner , B. H. Juang , “Fundamentals of Speech Recognition”, Prentice-Hall, Englewood Cliffs, N.J.

CHAPTER 14

LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION

14.0 Objective

In this chapter you will learn

- What is subword speech unit
- How to use standard statistical modelling technique to model subword units
- How subword units can be trained automatically from continuous speech
- What are the problems of creating and implementing word lexicons

14.1 Introduction

- The standard approach to large vocabulary continuous speech recognition is to assume a simple probabilistic model of speech production by specified word sequence W, produces an acoustic observation Y with probability P(W,Y).
- The goal is then decode the word string based on acoustic observation sequence so that the decoded string has the maximum posteriori probability (MAP) i.e.

$$\hat{W} \ni P(\hat{W}|Y) = \max_{W} P(W|Y). \quad \text{---- 14.1}$$

Using Bayes' rule the above equation can be written as

$$P(W|Y) = \frac{P(Y|W)P(W)}{P(Y)}. \quad \text{---- 14.2}$$

- Since P(Y) is independent of W the MAP decoding rule of eq 14.1 is

$$\hat{W} = \arg\max_{W} P(Y|W)P(W).$$

- The first term P(Y|W) of above equation is called the acoustic model as it estimates the probability of a sequence of acoustic observation conditioned on word string
- The second term P(W) is called language model as it describes the probability associated with postulated sequence of words.
- Such language models can incorporate both syntactic and semantic constraint of the language and recognition task.

- The languages models are integrated into acoustic model and are represented in a finite state network as to be integrated into acoustic model in straightforward manner.

14.2 SUBWORD SPEECH UNITS

For speech model recognition we have used the whole-word model as the basic of speech unit for isolated and continuous speech recognition system.

The advantages of using whole-word model are:

- Their acoustic representation is well defined and acoustic variability occurs at the beginning and end of the word.
- It removes the need for lexicon, hence it makes the recognition structure simple.

The disadvantage of using whole-word model is:

- Number of word utterance in training set need to be large
- Because of the large vocabulary, the phonetic content of the individual word inevitably overlap.

Hence we need efficient speech representation for such large vocabulary system that is why we use subword speech units

Following are the choices of subword units which can used to model speech

1. Phonelike Units (PLUs)
    - Here we define the units based on linguistic similarity of sound but model the unit based on acoustic similarity.
    - The acoustic properties of this units are different than the acoustic properties of basic phonemes
    - In English there are 50 PLUs
    - Advantage of PLU is ease of creating word lexicons

2. Syllable-like units
    - Here, to initially define the units we use vowel nucleus plus optional initial or final consonants or consonants clusters, and model the unit based on acoustic similarity
    - In English there are 10,000 syllables

3. Demisyllable-like units

IDOL

- Consist either the initial consonant cluster and some part of the vowel nucleus or the remaining part of the vowel nucleus and final consonant cluster
- In English there are around 2000 demisyllable-like units

4. Acoustic Units
    - Defined using specified objective criterion on the basis of clustering of speech segments from segmentation of fluent, unlabelled speech.
    - Codebook of speech units is created whose interpretation, in terms of classical linguistic units, is at best vague and worst totally non-existent.
    - To model a range of speech vocabularies ,set of 256-512 acoustic units is appropriate.

- We will take an example of English word “segmentation”. The representation of this word by above subword unit set is

  PLUs: /s/ /ɛ/ /g/ /m/ /ə/ /n/ /t/ /e^{y}/ /sh/ /ə/ /n/ (11 units)

  Syllables: /seg/ /men/ /ta/ /tion/ (4 syllables)

  Demisyllables: /sɛ/ /ɛg/ /mə/ /ən/ /te^{y}/ /e^{y}sh/ /shə//ən/ (8 demisyllables)

  Acoustic units: 17 111 37 3 241 121 99 171 37 (9 acoustic units)

- From above example we can see the number of subword units for “segmentation” word is small as 4 from syllables and large as 11 from PLU
- The issue in subword unit is the choice of subword unit set are the context sensitivity and the ease of training the unit from fluent speech
- There is no perfect set of subword unit.
- PLU set is extremely context sensitive but due to small number of PLU they are easy to train
- Syllables are least context sensitive but there are so many of them that they are difficult to train

IDOL

14.3 SUBWORD UNIT MODEL BASED ON HMMs

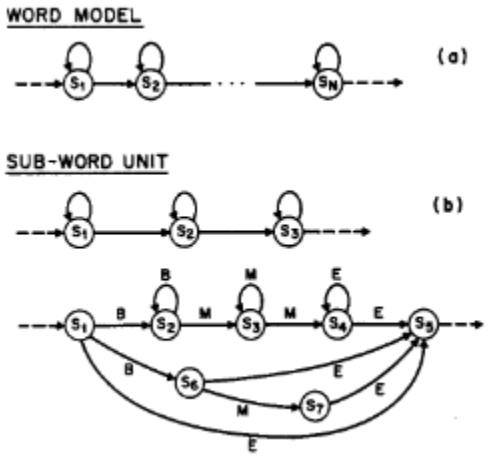


Fig:14.1 HMM representations of word (a) and a subword unit (b)

For Whole-word model basically uses a left-to-right HMM with N states as shown in above figure (a), N can be fixed value or number of sound in word or average number of frames in word.

For subword units the number of states in the HMM is fixed value as shown in above figure (b), where three state model is used. The shortest token s of each subword units must last at least three frames

Following are the three approaches has been used to represent the spectral density associated with each subword as shown in following figure

IDOL

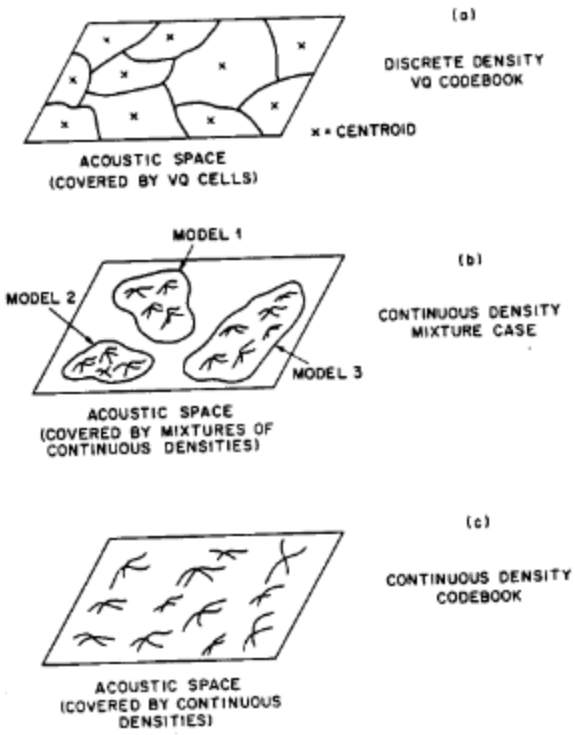


Fig:14.2 Representations of the acoustic space of speech by (a) partitioned VQ cells , (b) sets of continuous mixture Gaussian densities , and (c) a continuous-density codebook

1. VQ-based codebook

- as shown in part (a) of above figure
- The probability density of the observed spectral sequence is simple discrete density defined over the codebook vectors.
- It implicitly isolate the part of acoustic space within spectral vectors occur and assigning the appropriate codebook vector a fixed probability for spectral vector within each isolated region regardless of its proximity to corresponding codebook vector.

2. Mixture density

- As shown in part (b) of above figure
- Represents the continuous probability density by mixture density that explicitly defines the part of acoustic space in which spectral vectors occur.
- Each mixture component has spectral mean and variance that is highly dependent on the spectral characteristic of subword unit.
- Hence the models for different subword units usually do not have the substantial overlap in acoustic space.

3. Continuous density codebook

- As shown in part (c) of above figure
- The entire acoustic space is covered by a set of independent Gaussian densities with the resulting set of means and covariance stored in codebook.
- It differs from discrete density case in way the probability of an observation vector is computed instead of assigning a fixed probability according to the closeness of the observation vector to the codebook vector.
- Each state of each subword unit the density is assumed to be a mixture of the finite codebook density. Since the codebook set of Gaussian densities is common for all states of all subword models we can precompute the likelihood associated with an input spectral vector for each of codebook vector and determine the state likelihood using only a dot product with the state mixture gains
- This mixed density method has been called tied mixture approach as well as the semi-continues modelling.

14.4 TRAINING OF SUBWORD UNITS

- There is inherent tying between the subword unit across the words and sentence, every subword unit occurs many times in any size training set. So the training of the model for subword unit is extremely difficult
- Hence the estimation algorithm like forward backward procedure or k-means algorithm can start with uniform segmentation and rapidly converge to the best model estimates in few iteration.
- Assume ,We have a labelled training set of speech utterance where each sentence consist of speech waveform and its transcription in to words
  - A word lexicon which provides transcription of every word in the training set And a silence precede of follow any word within sentence
  - Based on above assumption typical sentence in training set is

    $S_w : w_1, w_2, w_3, \ldots, W_l$

    In which each W_i is a word in lexicon

  - The sentence “show all alerts” is three word sentence

    W_1 = show

    W_2= all

    W_3=alerts

  - The sentence S can be written in terms of subword unit as

$$S_U: \quad U_1(W_1)U_2(W_1)\ldots U_{L(W_1)}(W_1) \oplus U_1(W_2)U_2(W_2)\ldots U_{L(W_2)}(W_2) \oplus U_1(W_3)U_2(W_3)\ldots U_{L(W_3)}(W_3) \oplus \cdots \oplus U_1(W_I)U_2(W_I)\ldots U_{L(W_I)}(W_I),$$

- Where $U_i(.)$ represents the i^{th} unit and $L(w_1)$ length of word W_1 etc.
- Then we replaced each subword unit by its HMM
- This process is shown in following figure

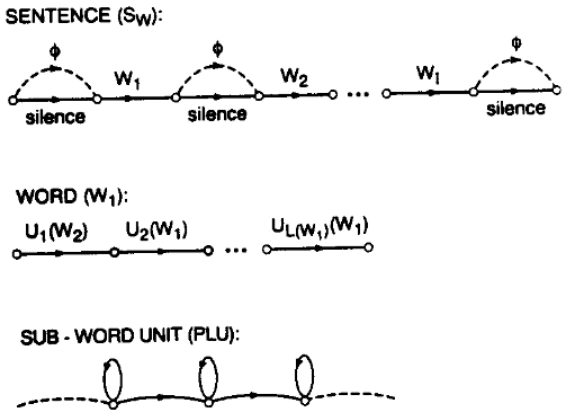


Fig: 14.3 Representation of a sentence, word and subword unit in terms of FSNs

- Here sentence is represented as finite-state network (FSN) where the arcs are either word or silence. Each word represented as FSN of subword unit and each subword unit is represented as a three state HMM
- The following figure shows the process of creating the composite FSN for sentence based on single word pronunciation lexicon

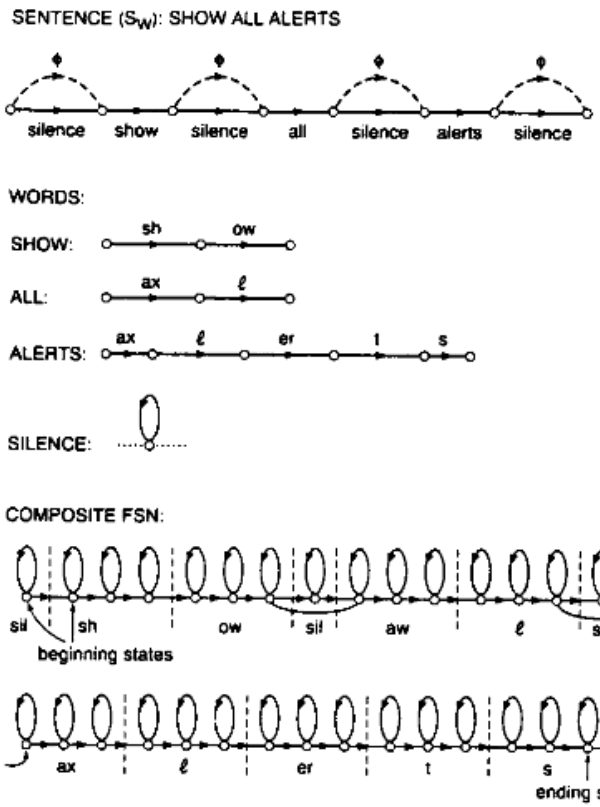


Fig: 14.4 creation of composite FSN for sentence "show all alerts"

- Once the composite sentence FSN is created for each sentence in training set, the training problem becomes one of estimating the subword unit model parameter which maximize the likelihood of the models for all given training data.
- Thus maximum likelihood problem can be solved using forward-backward procedure or k-means algorithms
- We use k-means training procedure to estimate the set of model parameter is as follows
  1. Initialization

     Linearly segment each training utterance into units and HMM states assuming no silence between words ,a single lexical pronunciation of each word and a single model for each subword unit
  2. Clustering :

     All feature vectors from all segments corresponding to a given state (i) of a given subword unit are partitioned into M cluster using the k-means algorithm.

3. Estimation :
   The mean vector μ_{ik} the covariance vector matrices U_{ik} and the mixture weights c_{ik} are estimated for each cluster k in state i
4. Segmentation :
   The updated set of subword unit models is used to resegment each training utterance into units and states. Here multiple lexical entries can be used for any word in vocabulary
5. Iteration :
   steps 2-4 are iterated until convergence

- In this we saw, one can use a training set of speech and optimally determine the parameters of a set of subword unit HMMs.
- The resulting parameter estimates are extremely robust to training material as well as to details of word pronunciation is obtained from word lexicon
- Common word lexicon is used for both training and recognition hence errors in associating proper subword units to words are consistent throughout the process

14.5 LANGUAGE MODELS FOR LARGE VOCABULARY SPEECH RECOGNITION

- A large vocabulary speech recognition system is dependent on the language embedded in the input speech.
- Therefore incorporation of knowledge of language in the form of language model is essential.
- The main goal of language model is to provide estimate of the probability of a word sequence W for recognition task.
- Assume that W is specified sequence of words i.e.

$$W = w_1 w_2 \ldots w_Q \qquad \text{.....(1)}$$

Then probability of W can be computed as

$$P(W)=P(w_1 w_2 \ldots w_Q)=P(w_1)P(w_2|w_1)P(w_3|w_1 w_2)\ldots P(w_Q|w_1 w_2 \ldots w_{Q-1}) \quad ...(2)$$

- But unfortunately it is impossible to estimate the conditional word probabilities $P(w_j|w_1 \ldots w_{j-1})$ for all word and all sequence length in given language.
- Hence there are language model define for this

1. N-gram word Model
   - In this model we appropriate the term $P(w_j|w_1 \ldots w_{j-1})$ as

$$P(w_j|w_1 w_2 \ldots w_{j-1}) \approx P(w_j|w_{j-N+1} \ldots w_{j-1})$$

- This is based on preceding N-1 words.
- N-gram model is also difficult to estimate reliably for all but N=2 or 3
- Hence in practice often uses the next model i.e. word pair model

2. Word pair model
   - Specifies the which word pair are valid in the language through use of binary indicator function

$$P(w_j|w_k) = \begin{cases} 1 & if\ w_k\ w_j\ is\ valid \\ 0 & othewise \end{cases}$$

3. No-grammar model
   - Assumes $P(w_j|w_k) = 1$ for all j and k
   - So that every word is capable of being followed by every other word in the language

Other than this model includes formal grammar, N-grams of word classes etc. These types of grammars provide more realistic models for natural language input to machines than the artificial N-gram or words or the word pair grammars. These models are difficult to integrate with acoustic decoding.

14.6 STATISTICAL LANGUAGE MODELING

- The language model P(W) has to be estimated from the collection of text.
- Here we will discuss about the how to construct such statistical model from collection of training corpus.
- The word sequence probability P(W) is approximated by

$$P_N(W) = \prod_{i=1}^{Q} P(w_i|w_{i-1}, w_{i-2}, \ldots, w_{i-N+1}),$$

....(1)

- The above equation is called N-gram language model.
- The conditional probabilities $P(w_i|w_{i-1},\ldots,w_{i-N+1})$ can be estimated by simple relative frequency approach

$$\hat{P}(w_i|w_{i-1}, \ldots, w_{i-N+1}) = \frac{F(w_i, w_{i-1}, \ldots, w_{i-N+1})}{F(w_{i-1}, \ldots, w_{i-N+1})},$$

.....(2)

- In which F is the number of occurrences of string in collection of training text

- In order to estimate above equation(2) to be reliable, F has to be substantial in given corpus.
- The implication of this are, that the size of training corpus may be prohibitively large and F = 0 for many possible word strings due to the limited size of corpus
- One way to circumvent this problem is to smooth the N gram frequencies
- Consider N = 3 the trigram model
- The smoothing is done by interpolating the trigram, bigram and unigram relative frequencies

$$\hat{P}(w_3|w_1,w_2) = p_1\frac{F(w_1,w_2,w_3)}{F(w_1,w_2)} + p_2\frac{F(w_1,w_2)}{F(w_1)} + p_3\frac{F(w_1)}{\sum F(w_i)},$$

In which non negative weights satisfy $p_1+p_2+p_3 = 1$ and $\sum F(w_i)$ is the size of corpus.

14.7 Perplexity of the language model

How well the language model perform in the context of the speech is answered based on the concept of source of information in information theory

To provide such measure of performance we discuss several concepts including entropy, estimate entropy and perplexity

Consider an information source that puts out sequence of words $w_1,w_2,\ldots w_Q$ each of which is chosen from $\bar{V}$ with size $|\bar{V}|$ according to law

The entropy of source can be defined as

$$H = -\lim_{Q\to\infty}\left(\frac{1}{Q}\right)\left\{\sum P(w_1,w_2,\ldots,w_Q)\log P(w_1,w_2,\ldots,w_Q)\right\}, \quad \ldots..(1)$$

In which p() is the probability of argument string

If the word in string are sequence are generated by the source in an independent manner

$$P(w_1,w_2,\ldots,w_Q)=P(w_1)P(w_2)\ldots p(w_Q) \quad \ldots..(2)$$ then

$$H = -\sum_{w\in\bar{V}} P(w)\log P(w), \quad \ldots...(3)$$

Which is the first order entropy of the source

The quantity of H of eq. (1) can be considered as the average information of source when it puts out a word w.

Equivalent source of entropy H is one that has as much information as a source which puts out words equiprobably for vocabulary of size 2^H

One way to estimate H is to use $P(W) = P(w_1,w_2,\ldots,w_Q)$ from the language model. If N gram model is used an estimate of H is thus

$$H_p = -\frac{1}{Q}\sum_{i=1}^{Q} \log\ P(w_i|w_{i-1}, w_{i-2}, \ldots, w_{i-N+1}).$$

In general

$$H_p = -\frac{1}{Q} \log\ \hat{P}(w_1, w_2, \ldots, w_Q),$$

Where $\hat{P}(w_1,w_2,\ldots,w_Q)$ is an estimate of $P(w_1,w_2,\ldots,w_Q)$

The quantity H_p is an estimate entropy as calculated from sufficient long sequence based on a language model.

Associated with H_p is a quantity called perplexity defined as

$$B = 2^{H_p} = \hat{P}(w_1, w_2, \ldots, w_Q)^{-1/Q}.$$

The H_p is the average difficulty or uncertainty in each word based on the language model.
Another way to view perplexity is to consider it as the average number of possible words following any string (N-1) words in large corpus based on N-gram language model.
Perplexity is an important parameter in specifying the degree of sophistication in a recognition task from the source of uncertainty to the quality of the language model

14.8 Overall recognition system based on subword unit

- The following diagram show the overall continuous speech recognition system based on subword speech unit

IDOL

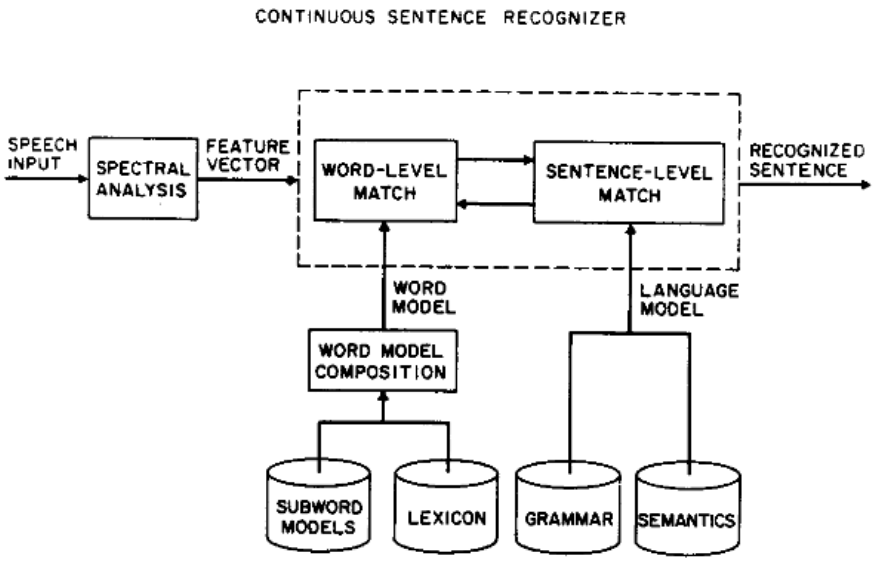


Fig: 14.7 Overall block diagram of subword unit based continuous speech recognizer

1)Spectral analysis

It is first step in processing.

It is to derive the feature vector used to characterize the spectral properties of the speech input.

For most part we will consider a spectral vector with 38 components consisting of 12 cepstral components, 12 delta cepstral components, 12 delta-delta cepstral components, delta log energy and delta-delta log energy.

2) Combined word level/sentence level match

- It is second step in processing
- Using set of subword HMMs and word lexicon, a set of word models is created by concatenating each of the subword unit HMMs as specified in lexicon
- Sentence level match is done via an FSN realization of the word grammar and the semantics as expressed in a composite FSN language model

Most system uses similar to the frame synchronous level-building method to solve for the best recognition sentence

IDOL

Consider a recognizer of above figure for a database management task called Naval Resource Management task-defined within DARPA community, which has a 991 –word vocabulary

Following are the typical sentences used to query the database include

What is mishawaka's percent fuel

Total ships that will arrive in diego-garcia by next month

Do any vessels that are in gulf of tonking have asw mission area of m4

Show the names of any submarines in yellow sea on twenty eight October

List all the alerts

What's jason's m-rating on mob

Give t-lam vessels that weren't deployed in November

Thus vocabulary includes many jargon words such as m4, m-rating ,mob and t-lam

Several long content word such as mishwaka's, diaego-garcia, submarines, November etc.,

And short function words as is, the , by, do, in, of etc.

A wide range of sentences can be constructed from the 991 –word vocabulary to query this database

It is possible to construct the finite state network presentation of the full grammar associated with all sentence

The perplexity of the full grammar network computed is 9.

The least constraining grammar is the no grammar (NG) in which any word in vocabulary is allowed to follow any word in vocabulary

The perplexity of the FSN of NG case is 991

FSN for NG is shown in following figure

IDOL

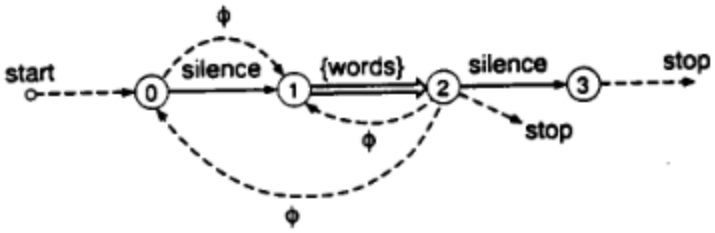


Fig 14.8 FSN for the NG syntax

A second FSN form of task syntax is created a word pair (WP) grammar that specifies explicitly which words can follow each of 991 words in vocabulary

The perplexity of this grammar is about 60

The above figure could be used for WP grammar, A more efficient structure exploits the fact that only a subset of the vocabulary occurs as the first word in sentence(B) and last word in sentence(E)

Hence we can partition the vocabulary into four non overlapping sets of words

$\{BE\}$ = set of words that can either begin or end a sentence, $|BE|$=117

$\{B\bar{E}\}$ = set of words that can begin a sentence but cannot end a sentence, $|B\bar{E}|$=64

$\{\bar{B}E\}$ = set of words that cannot begin a sentence but can end a sentence, $|\bar{B}E|$=488

$\{\overline{B\bar{E}}\}$ = set of words that cannot either begin or end a sentence, $|\overline{B\bar{E}}|$=322

The resulting FSN is shown in following figure

IDOL

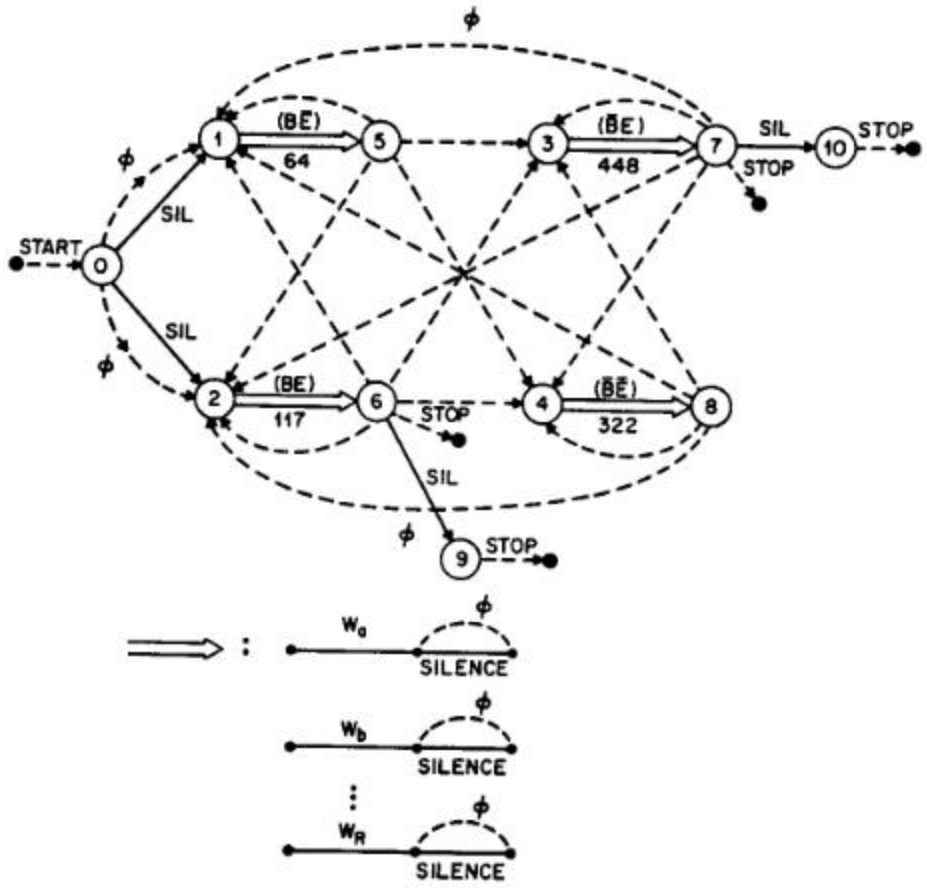


Figure 14.9 FSN for the WP syntax

This network has 995 real arcs and 18 null arcs

To account for silence between words, each word arc bundle is expanded to individual words followed by optional silence as shown at the bottom figure

Hence overall FSN allows recognition of sentence of the form

S: (silence)-{ $B\bar{E}, BE$}-(silence)-({W})…({W})-(silence)-{ $\bar{B}E, BE$ }-(silence)

Finally one could construct a task syntax based on statistical word bigram probabilities we assign a probability p_{ij}, to each word (W_i,W_j) where p_{ij} is the probability that W_i is followed immediately by W_j.

If W_n is the n^{th} word in a string of words then

$P_{ij} = P(W_n=W_j|W_{n-1}=W_i)$

The advantage of the word bigram approach is that perplexity is reduced considerably for the Resource Management task, with essentially no increase in complexity of the implementation.

14.8.1 Control of word Insertion/word deletion rate

Using structure of the type shown in above figure 8.9 ,there is no control on the sentence length.

It is possible to generate the sentences that are arbitrarily long by inserting a large number of short-function words to prevent this from occurring it is a simple matter to incorporate a word insertion penalty into Viterbi decoding such that a fixed negative quantity is added to the likelihood score at the end of each word

By adjusting a the word penalty we can control the rate of word insertion and deletion

A very large word penalty will reduce the word insertion rate and increase the word deletion rate

A very small penalty will have opposite impact. A value for word penalty is usually experimentally determined

14.8.2 System performance on the resource management task

To evaluate the recognizer performance five different sets of test data were used including

Train 109 : a randomly selected 2 sentences from each of the 190 training talkers

Feb 89 : set of 30 sentence from each 10 talkers none of whom was in training set distributed in feb 1989

Oct 89 : second set of 30 sentence from each 10 additional talkers none of whom was in training set distributed in oct 1989

Jun 90 : set of 120 sentence from each 4 new talkers none of whom was in training set distributed in jun 1990

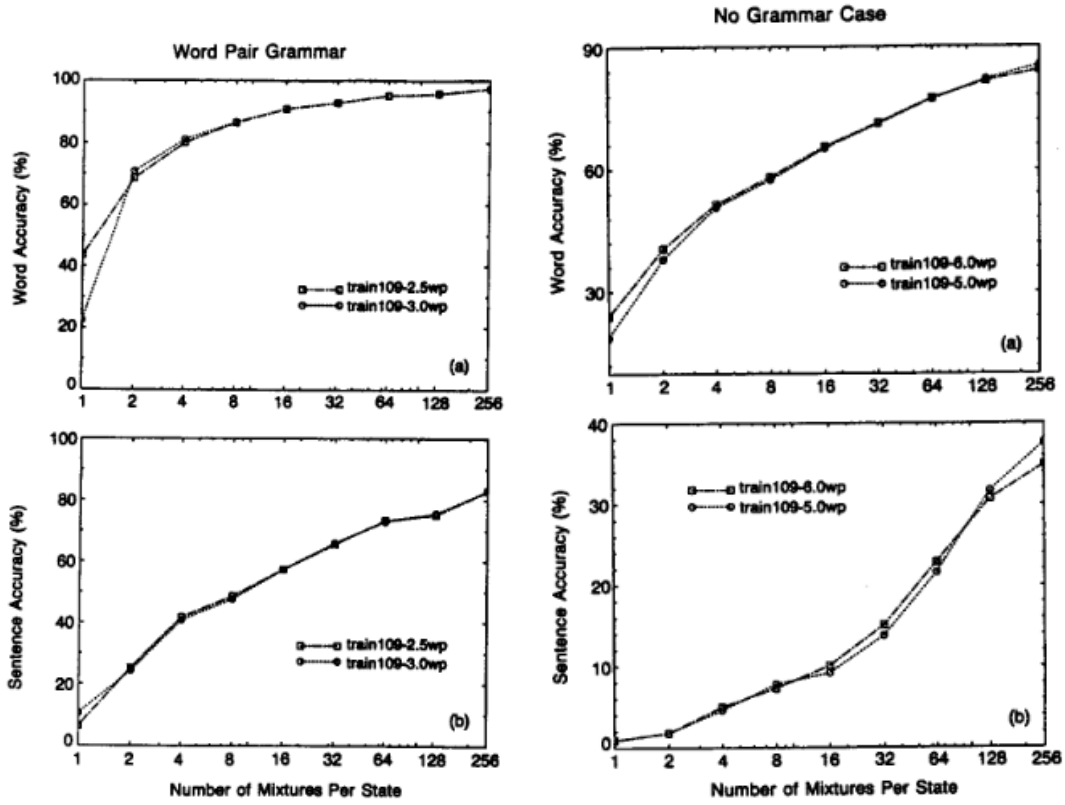
Feb 91 : set of 30 sentence from each 10 new talkers none of whom was in training set distributed in feb 1991

The recognizer performance was evaluated for each of the test sets using both WP an NG syntax and different word penalties

IDOL

The recognition result are presented in the form of word accuracy and sentence accuracy as a function of mixture of per state for each PLU model.

The recognition result on the training subset(train 109) are given in following figure (for WP syntax) and (for NG syntax).



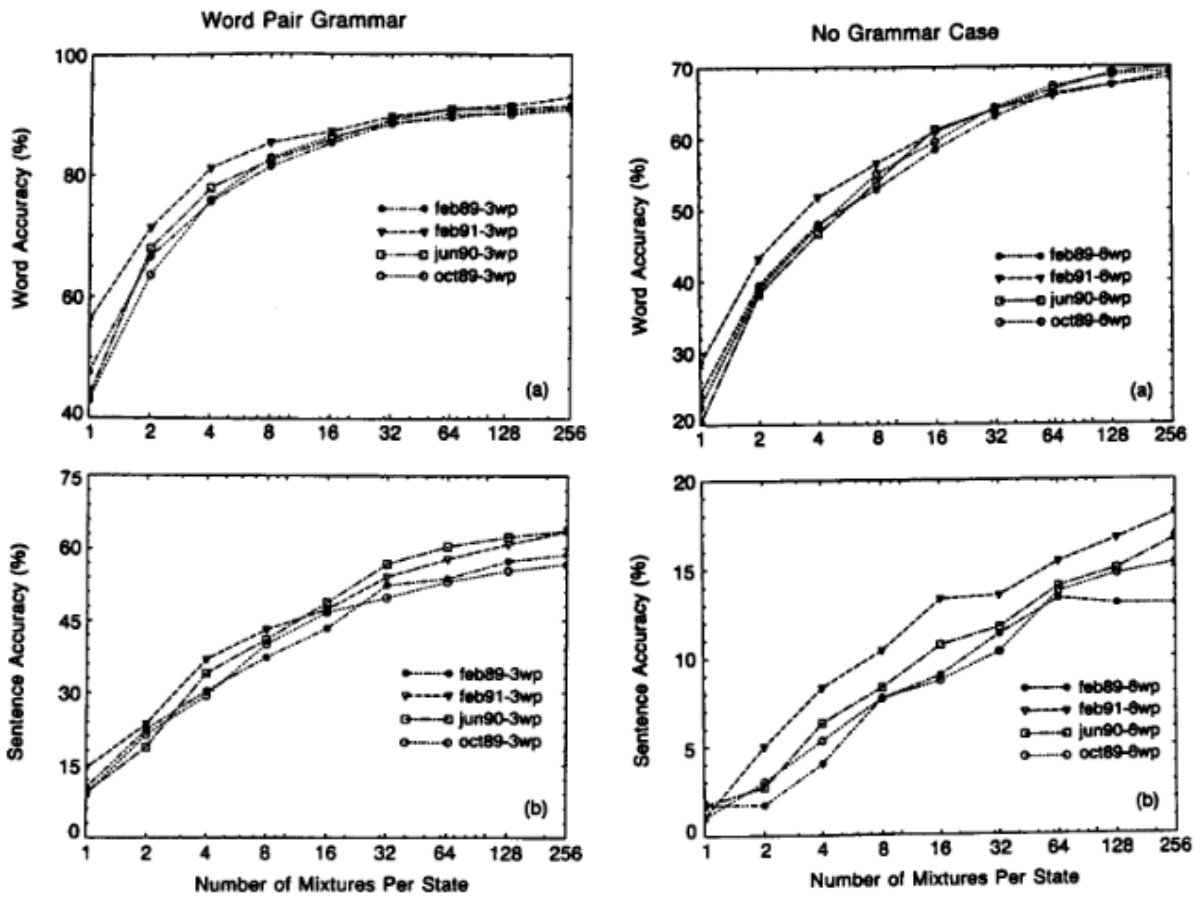
The upper curves show word accuracy versus number of mixture per state for two different values of word penalty

The lower curve show sentence accuracy for same parameters

Number of mixture per state increases going from about 43.6% accuracy for 1 mixture per state to 97.3% for256 mixture per state for the WP syntax using a word penalty of 2.5.

For NG syntax using word penalty of 6.0 the comparable results were 24% word accuracy for 1 mixture per state and 84.2% word accuracy for 256 mixture per state

The recognition result on independent test set are given in figures (for WP syntax) and (for NG syntax).



For WP syntax

Range of word accuracy for 1 mixture per state is 42.9%(for feb 89) to 56.0%(for jun 90) whereas 256 mixture per state the range is 90.9%(for feb 89) to 93.0% (for jun 90)

For NG syntax the range of word accuracy for 1 mixture per state is 20.1% (for feb 91) to 28.5% (for jun 90) and 256 mixture per state it is 68.5%(for oct 89) to 70.0% (for feb 91)

The most significant aspect of the performance is the difference in accuracies between the test set and training subset

Thus there is gap of 4-7% in word accuracy for WP syntax at 256 mixture per state and gap of 14.2-15.7% for NG syntax at 256 mixture per state.

Such gaps are indicative of the ability of training procedure to overtrain on the training set, thereby achieving significantly higher recognition accuracy on this set

14.9 Context dependent subword unit

- There are several advantages to using Context dependent subword unit.
- First, is the model of subword units are easily trained from fluent speech, with no human decision.
- Second, the resulting units are generalizable to new context with no extra efforts.

14.9.1 Creation of context dependent Diphones and triphones

- Consider the basic set of context-independent PLUs
- We use the symbol p to denote an arbitrary PLU.
- We can define a set of context-dependent (CD) diphones as

  p_L - p - $ left context (LC) diphone

  $ - p - p_R right context (RC) diphone,
- in which p_L is the PLU immediately preceding p (the left context sound), P_R is the PLU immediately following p (the right context sound), and $ denotes a don't care or don't know condition

- similarly we can define a set of context dependent triphone as

  p_L - p - p_R Left – right context (LPC) triphone

In theory, number of diphones is 46x45 for basic set of 47 PLUs and excluding silence or about 2070 left context diphone units ,the number of left-right context triphone units is 45x46x45 or 93150 units

14.9.2 using interword training to create CD units

- The lexical entry for each word uses right or left context diphone units for the first and last sound of each word, we can utilize the known sequence of words to replace these diphone units with the triphone unit appropriate to the words spoken.
- Hence the sentence "Show all ships" would be represented using intraword units as

  $-sh-ow sh-ow-$ $-aw-l aw-l-$ $-sh-i sh-i-p i-p-s p-s-$

IDOL

- whereas the sentence would be represented as using both intra word and inter word units

  $-sh-ow sh-ow-aw ow-aw-I aw-I-sh I-sh-i sh-i-p i-p-s p-s-$

- there were only two triphones based on intra word units.
- there are six triphones based on intraword and interword units i.e. a threefold increase in context dependent triphone units
- Even when using interword units, the problems associated with estimating model parameters from a small number of occurrences of the units is the major issue.

14.9.3 Smoothing and interpolation of CD PLU Models

For training set of reasonable size, there is sufficient data to reliably train context-independent unit models but the number of units becomes larger the amount of data available for each unit decreases and the model estimates become less reliable.

There is no ideal solution for this problem, a reasonable comprise is to exploit the reliability of the estimates of higher level unit models to smooth or interpolate the estimates of lower level unit model. In many ways this such smoothing and interpolation can be achieved

Simplest way is interpolate the spectral models with all higher models that are consistent with the model

By this we mean that the model for the CD unit pL - p – pR should be interpolated with the models for the units $ -p –pR($\lambda_{\$-p-pR}$),pl - p - $ ($\lambda_{pL-p-\$}$) and $ - p - $ ($\lambda_{\$-p-\$}$). Such an interpolation of model parameters is meaningful only for discrete densities, within states of the HMM, based on a common codebook. Thus if each model λ is of the form (A, B, π) where B is a discrete density over a common codebook, then we can formulate the interpolation as:

$$\hat{B}_{p_L-p-p_R} = \alpha_{p_L-p-p_R} B_{p_L-p-p_R} + \alpha_{p_L-p-\$} B_{p_L-p-\$} + \alpha_{\$-p-p_R} B_{\$-p-p_R} + \alpha_{\$-p-\$} B_{\$-p-\$},$$

Where $\hat{B}_{PL-p-pR}$ is the interpolated density. We constrain the α s to add up to 1 ; hence

$$\alpha_{p_L-p-p_R} + \alpha_{p_L-p-\$} + \alpha_{\$-p-p_R} + \alpha_{\$-p-\$} = 1.$$

The way in which the as are determined is according to the deleted interpolation algorithm.

Other smoothing methods include empirical estimates of the α s based on occurrence count, co-occurrence smoothing based on joint probabilities of pairs of codebook and use of fuzzy VQs in which input spectral vector is coded into two or more codebook symbol

14.9.4 Smoothing and interpolation of continuous densities

It is very difficult to devise a good smoothing or interpolation for continuous density modelling because the acoustic space of different units is inherently different

There are two ways to handle this problem

1. Semicontinuous or tied mixture modelling
   In which each PLU uses a fixed set of mixture means and variance, and the only variables are the mixture gains for each model.
2. Tied-mixture approach on the CI unit level
   Here we design a separate codebook of densities for each CIPLU and then constraint each derived CD unit to use the same mixture means and variance but with independent, mixture gains.

14.9.5 Position dependent Units

- When using both intraword and interword units, it is natural and reasonable to combine occurrences of the same unit independent of whether they occurred within the word or across words.
- The phones within words are significantly more stable, than Phones occurring at word boundaries.
- Thus it seems plausible that the spectral behaviour of the same intraword and interword unit could be considerably different

IDOL

14.9.6 Unit splitting and clustering

- The issue in the design and implementation of large vocabulary is how to efficiently determine the number and character of context dependent units that gives best performance for given training set
- There is no simple answer in this section we will discuss the several proposed methods based on either starting from small set of context-independent units and iteratively splitting the units or start with large set of context dependent unit and merging similar units to reduce the number of units based on some type of clustering procedure

Splitting subword units

Subword splitting unit is illustrated in following figure

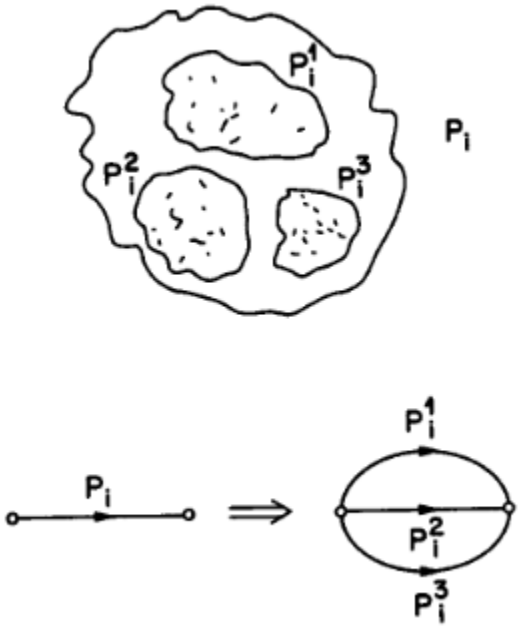


Figure splitting of subword unit p_i into three clusters

- assume that for each sub word unit p_i (with model λ_i), representing a context-independent unit, some inherent internal distribution of training tokens that naturally clusters into two or more cluster
- The cluster represent classes of sounds that are all labelled as p_i. but which have different spectral properties

- Once the separation of p_i into is achieved, effectively created multiple models of the context independent subword unit as shown in bottom of above figure
- The simple procedure for splitting off training tokens with low likelihood scores and creating a new model is as follows

1. For each sub word unit, p_i, which is to be split, all training tokens whose likelihood scores fall below a threshold are split off and used to estimate an additional model for that unit.
2. The segmental k-means training procedure is iterated on the split-off tokens until the new model reaches convergence.
3. The above procedure (steps I and 2) is iterated until the desired number of models, for each sub word unit, is obtained.

Clustering complex dependent unit

- The alternative to model splitting is model clustering in which we initially start with the complete set of context-dependent units and then sequentially merge units so that the decrease in likelihood score is minimized at each step.
- This procedure is iterated either until a desired number of units is reached or until the resulting decrease in likelihoods gets too large.
- Advantage of it is that it is trivial to modify the word lexicon to account for the decrease in units from merging.
- model clustering is inherently simpler to implement than model splitting and therefore has been used more widely in practical systems.

14.9.7 Other factors for creating additional subword units

Source of difficulty in continuous speech recognition are the so called function words, which include words like a, and. For, in, and is.

Function word dependent

- One simple idea is to represent function words independently of the rest of the training set, using either whole-word models, Multiple pronunciations in the lexicon (e.g., the, thee). Or special subword units called function word dependent units trained directly from occurrences of the function words within the training set

IDOL

- When function word dependent units are added to standard set of subword units it shows small but consistent improvements in recognition performance

Gender specific model

- Another interesting idea is to create separate sets of units for both male and female talkers.
- Spectral properties of the units are distinct for males and females.
- The problem is that by separating male from female talkers, the amount of training data for each separate gender set is reduced.
- Hence the reliability of the estimates of both sets of models is reduced even further
- Advantage is there is consistent gains in recognition performance using gender specific model

Combination of word models and subword unit

- The idea is that for words that do occur often in the training set creation of whole-word models provides the highest recognition performance.
- For all other words in the lexicon, some type of sub word units is required.
- Hence a combination of word and sub word units would probably lead to the best implementation for many applications4

14.9.8 Acoustic Segment Units

- To create a consistent acoustic framework, it is possible to define a set of acoustic segment units (ASUs) that can be trained from continuous (unlabelled) speech and which form a basis for representing any spoken input.
- For this we need to have a procedure that automatically segments fluent speech into unlabelled sections and then cluster the resulting segments to create a codebook of ASUs.
- The Problem now becomes creating an acoustic lexicon that represents words in recognition vocabulary in terms of appropriate sequence of ASUs.
- Technique for creating the acoustic lexicon exist and appear to work well for the system in which every vocabulary word is seen in training set,
- But for large vocabulary system the problem of automatically creating the acoustic lexicon remain a major problem

IDOL

Creation of vocabulary independent units.

References and further Reading

[1] L.R. Rabiner , B. H. Juang , “Fundamentals of Speech Recognition”, Prentice-Hall, Englewood Cliffs, N.J.

IDOL

Chapter 15

TASK ORIENTED APPLICATION OF AUTOMATIC SPEECH RECOGNITION

15.1 Introduction

In this chapter we will have look on the problem of how to integrate the speech recognition system into a task specific application to perform a useful tasks.

15.2 TASK SPECIFIC VOICE CONTROL AND DIALOG

To understand the problem of integrating the speech recognition system into task specific application we will consider the following figure of task specific voice control and dialog system

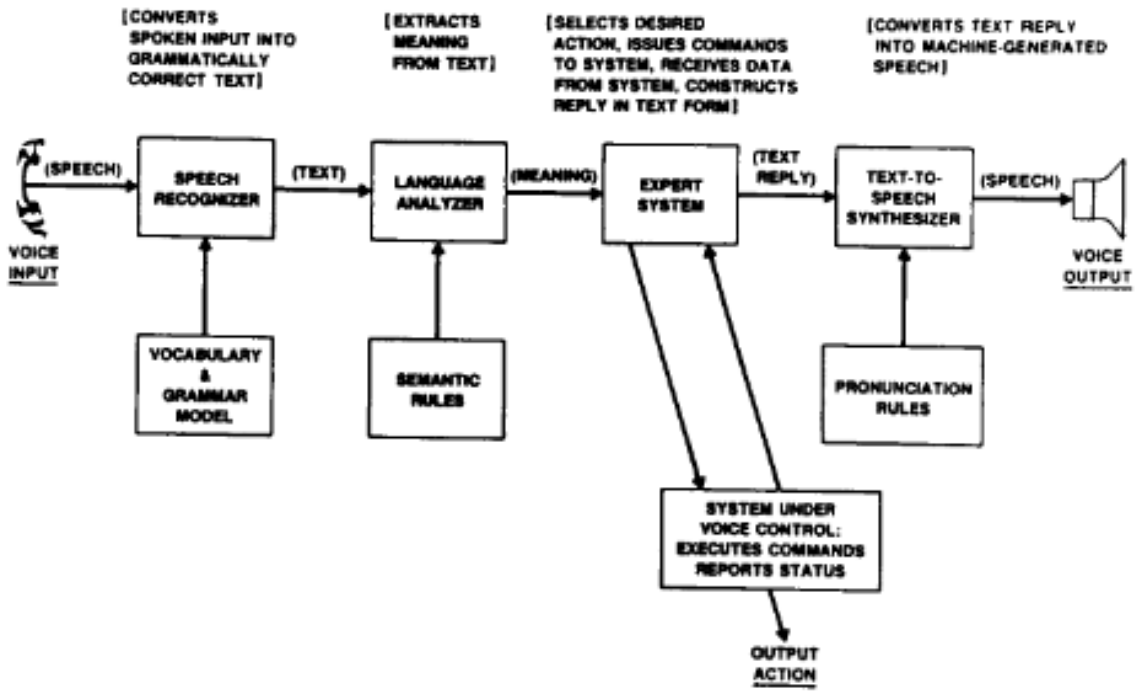


Figure 15.1 Block diagram of task-specific voice control and dialog system

- ` System consists of
  - A speech recognizer
  - A language analyzer
  - An expert system
  - A physical system being controlled by the voice commands
  - Text to speech synthesizer

Speech recognizer:

- The function of this block is to convert speech input into a grammatically correct text.
- It is constrained by the recognizer vocabulary and grammar model.
- The output of this block is the text string
- This text string is input to the next block.

Language analyzer:

- Input to this block is the text string which is sent from speech recognizer
- It extracts the meaning from the text with the help of semantic rules
- The decoded meaning is sent to the expert system.

Expert system:

- First selects the desired action then issues appropriate commands to a physical system under voice control to carry out the action then receives data on the command status
- Ex. Of command status are "command carried out successfully" or "command cannot be carried out"
- After that it constructs the textual reply

Text to speech synthesizer:

- A text reply is converted into a speech message
- Conversion is done with appropriate word pronunciation rules and played back to the user

IDOL

15.3 CHARACTERISTICS OF SPEECH RECOGNITION APPLICATIONS

Following are the requirements to decide whether the proposed task is suitable for speech-recognition development

- Beneficial to the user
- User friendly
- Accurate
- Real time

1. Proposed recognition system must provide a real benefit to the user in the form of
    - Increased productivity
    - Ease of use
    - Better m/c interface or a more natural way of communication
    - If the application is not useful to the user it do not succeed over time

2. The system must be user friendly.
    - User should feel comfortable,
    - it must provide
        - friendly and helpful voice prompts
        - an effective means of communications.

3. The system must be accurate.
    - It must achieve a specified level of performance on the task associated with the recognition decision

4. The recognition system must respond in real time.
    - The response to the query should be very fast so that voice dialog can be maintained

IDOL

15.4 METHODS OF HANDLING RECOGNITION ERROR

The fact is that the speech recognition system will make some error in recognition of spoken input the question is how to deal with it to handle such errors.

Following are the four ways to deal with recognition error:

- Fail soft methods
- Self-detection/correction of errors
- Verification or multilevel decision before proceeding
- Rejection/pass to operator

1. Fail soft methods
   - The cost of recognition error in terms time is low
   - Recognition Error is acceptable
   - The error will be detected and corrected at the later stage
   - If command word is misrecognized and next word is inappropriate then the user can enter into a correction mode to backtrack to the point where the error was made
2. Self-detection/correction of errors
   - The recognition system utilizes known task constraints to automatically detect and correct recognition errors
   - Ex. Spelling of the name from finite list of names, it is easy to detect and correct recognition error in spelled letters because the recognized name is constraint to the set of names within the given list
3. Verification or multilevel decision before proceeding:
   - The recognition system ask the user for help whenever two or more recognition words whose likelihood score is high and it is difficult resolving small differences in the strings
   - The recognizer ask the user to verify the first choice decision; if it is not verified, the recognizer ask the user to verify the second choice
4. Rejection/pass on to operator:
   - By recording all spoken Input in digital format, the system can reduce the error rate by rejecting a small but finite percentage of the spoken strings,
   - Then such strings are passed to a human operator who makes the final decision based on listening to the spoken input

By using all four techniques the accuracy of speech recognizer approaches 100%

IDOL

15.5 BROAD CLASSES OF SPEECH RECOGNITION APPLICATIONS

Five broad classes of applications to which speech recognition applied:

1. Office or business system application include

- Data entry
- Database management and control
- Keyboard enhancement

2. Manufacturing
   Application is used to provide “Eyes-free , hands free” monitoring of manufacturing for quality control.

3. Telephone or telecommunications
   - Many applications are feasible over dialed up telephones including :

- Automation of operator assisted services
- Telemarketing (inbound and outbound)
- Call distribution by voice
- Repertory dialing
- Catalog ordering

4.Medical

The primary application is voice creation and editing of specialized medical reports

5. Other:

- Voice controlled and operated games and toys
- Voice recognition aids for the handicapped
- Voice control in a moving vehicle
- Climate control

15.6 COMMAND AND CONTROL APPLICATIONS

User can control the machines using simple commands over speech

Voice repertory dialer

Voice repertory dialer allows a caller to place a call by speaking the name of someone in the repertory rather than dialing the digit code.

This is most useful in system where the advantage of eyes and hands free dialing of telephone numbers are clear like in mobile phone within a car

DOL

A repertory dialer needs a speaker trained set of vocabulary pattern corresponding to repertory names and their phone number

- It also Needs a speaker independent set of vocabulary patterns corresponding to the digits and set of command words for controlling normal telephone features like off-hook, dial, repeat, hang up
- For most user 10 to 20 repertory names, 10 digits and about 5 to 10 commands word are sufficient

Automated Call-type recognition

- The automation of operator-assisted to toll calls is the interesting telephone application of speech recognition
- Example is Call made from a pay phone that normally require operator assistance, including collect calls, person to person calls, third party billing calls, operator assisted calls and credit card calls
- Since There are only Five options for this service a vocabulary consisting only five words is adequate:
- “Collect” to make collect calls
- “Person” to make person to person calls
- “Third number” to make third party billing calls
- “Operator” to make operator assisted calls
- “Calling card” to make calling card calls

- The system is speaker independent and can work over the standard dialed-up telephone network
- If the customer obeys the voice prompt and spoke one of the command words then the accuracy of the system is more than 99%
- Customer have to use the specific command word

IDOL

- If customer uses the sentence like *collect* call please then keyword spotting technique have to be used to find out the command words embedded within the sentence
- A powerful key spotter is used to spot the command in sentence using key word spotter the overall command accuracy remained about 99%.

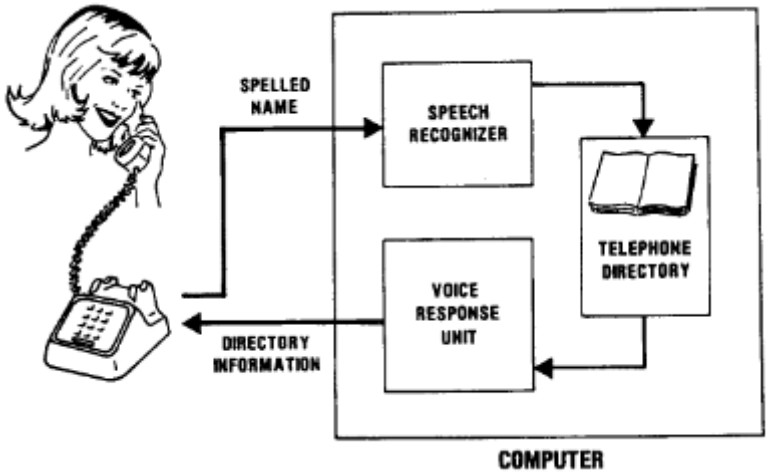
Call distribution by voice commands

Another telephony based task where voice recognition is applied is the area of Call distribution by voice commands

- A call is placed that will normally answered by an operator who then distributed the call to the appropriate location (person) based on the users responses to the questions asked by the attendant
- In this application the attendant function is automated via voice processing
- The voice response system poses a series of menu based questions, and based on the user responses, route the call appropriately
- The examples where such application is used includes railway system, hotels, departmental store, product customer care services etc.

Directory listing retrieval

A block diagram of the access to directory information from spoken spelled names is shown below:



- To access the directory information for a name in the directory, the user spells the name using the word “stop” between the last name and the initials as in

  “Rabiner-stop-LR-stop”

- The speech recognizer determines the name in the given directory which best matches the spoken input and then speaks the directory information for that name to the user
- there may be an error due to similar sounding letters, but the telephone directory provides task syntax that automatically detects and corrects improperly recognized letters
- it not only work with correctly spelled input, the system can handle common misspelling of names with a single insertion or deletion of letter or a single letter substitution

Credit card sales validation

Another application area of telecommunication is the automated validation of credit card sales.

- Here in this application area whenever the credit card sales transaction occurs where Merchant needs credit card validation and does not have automatic card reader then Merchant must call to a specific number and provides an attendant with 10 digit merchant identification number, a 15 digit credit card no. and the amount in dollar or rupees of the transaction.
- In this case a speech recognition system uses a connected digit recognizer to recognize the merchant identification number and credit card no. and a connected word recognizer for the transaction amount
- In this suppose the amount is 189 dollar then there are few varieties of specking this amount as
  - One eighty nine
  - One hundred and eighty nine
  - One eight nine
  - One hundred eight nine

  In this same amount or string can been spoken in variety of ways.

IDOL

- For amount the vocabulary size for recognition is larger than that of need for credit card hence the recognition task is quite bit more difficult

References and further Reading

[1] L.R. Rabiner , B. H. Juang , "Fundamentals of Speech Recognition", Prentice-Hall, Englewood Cliffs, N.J.

IDOL