T.V

→ $f(x_1, x_2 \ldots x_n)$

→ dependent variable or function by one or more independent variables

→ a physical quantity that varies with time

classmate Date 23/07/25 Page

i) Basic understanding of signals — time implicit, amplitude

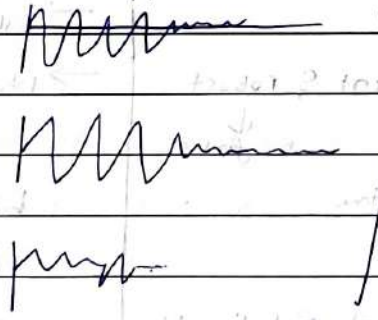
ii) Find meaning out of these signals — how to do it

- which protocols
- which technique
- if they work
- how much
- Where they fail?

iii) No attendance — No 75%
No midsem — 5%
No endsem — 15%
⇒ Continous evaluation (No makeup quiz)
• (2 quizzes/week)

iv) 40% weightage — Assignment : everything 'uttered'
⇒ (C-prereq) ⇒ Needs to be implemented

v) Probability concepts applied like,
for hello :



We can have all these 3 variations
but, how can we identify if all 3 corresponds to 'hello'
⇒ for that we look at the probability produced that Signals

vi) Speech Recognition System : Final Dployed project

vii) Books : Rabiner & Juang → (Read this)
* Fundamental of Speech Recognition

(ii) Rabiner & Sch for
Digital Processing of Speech Signals

(iii) Vector Quantization & Signal Compression — Gersho & Gray

(iv) Khalid Sayood — Data Compression

viii) Skills to develop for assignment

QR code on Notice Board in CSE dept ←
- Visual Studio 2010
- Cooledit 2000 – for editing the speech waveform
-

Q Drone are detectless. Can we build a system to detect? → Signatures of different drones (frequency)
→ And do something (.....)

12/08/25

Revise

1) What are the things that go into SP? → Speech processing (CL4PS)

Areas:

A) Signal Processing (SP2): It involves extracting relevant parameters from the speech signal
→ It should be efficient & robust
SP algorithms → real time; → no failure

B) Physics: (Acoustics) Relationship between the physical speech signal & physiological mechanism – vocal tract, hearing mechanism

C) Pattern Recognition: These are algorithms to cluster data and create templates or models and matching procedures.

D) Communication and Information Theory: Procedures for estimating parameters of statistical models, detecting the presence or absence of speech patterns, coding & decoding

Fact1

- 1998 – Microsoft bought some (Entropic) Company which was working on speech recognition
do it (exam)
→ What type of software? ESPS/Waves+ Toolkit ↑ for analyzing speech & signals
→ What did they do?
↓↓
which was open sourced
HTK – Hidden Markov Toolkit
- (ANSI C – Code)
- not properly documented

E) Linguistic Knowledge : Relationship between sounds (phonology), words in a language (which is a syntax), meaning (semantic), and use derived from meaning (pragmatics)

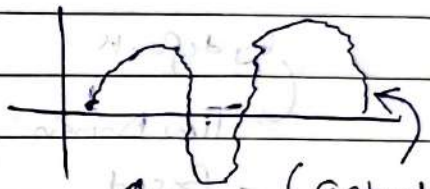
F) Physiology : Understanding higher order mechanisms within the human central nervous system that account for speech production & perception in humans (embed this knowledge in ANNs)

G) Computer Science : Study of efficient algorithms for implementing in software & hardware, the various methods of speech processing systems

H) Psychology : Understanding the factors that enable human to use speech processing systems for practical tasks

CD stores music digitally & not mp3.

- What is mp3's format?
- pulse code modulated (PCM) data : cd stores
  - → no compression
  - → grammophone req for it



(actual grooves)

digital

Primitive media

⇒ Compression : Huffman coding

types

lossy coding (mp3)

lossless coding

Speech requires language //

19/08/25

Different Approaches to Speech Processing

i) Acoustic Phonetic Approach
ii) Pattern Recognition Approach
iii) AI based Approach

- DSP

- Speech → Mathematical form
  - Physical form

Step 1: Speech Production (Speaker)

- message formulation
- Speak - selection of a language : language code
- neuro-muscular action

→ Homework ←

give us signature {Regularity of an alphabet in spacing} ⇒ (pitch period)

{→ Draw the wave for each alphabets as you speak → Record → from wave (select some) → save selection → in ASCII format

Step 2 : Speech Perception (Listener)

- Capture of (logitudinal waves) air waves through ear

Modular steps in step 2

→ starts off with the acoustic wave, hits the basilar membrane,
→ does a neural transduction
→ then language code is decoded
→ finally message comprehension

Assignment

(i) record
(ii) select some part - the signature (search it up) : pitch period
(iii) save in ASCII format
(iv) run it again
(v) write characteristics for spectrogram & the wave

by default
→ Time Domain Based Approach
another is: frequency
how many oscillations per sec

for this : wave + spectral view

Spectrogram: 3D view of same signal
↓ consists of
dark band : Resonances of Speech (no single frequency, multiple frequency) (high energy)
(can be done with GPU as well)

1.

Simplify: theorem — maximum frequency that can be achieved at sampling rate x is x/2

Go to analyse → ( )
↳ instantaneous analysis
↑ 2D graph for 3D view
↑ peaks are the resonating frequency

After opening, the cool edit
- F10 btn - press : Red color bar
  - measured in decibles
  - to measure noise (on record btn press)
- Sample rate - sample/s
  ↳ as digital so (....)
- Mono channel - single channel
- each sample 16 bit - resolution

How to deal with the noise (background noise)

What does the negative represent?
- diaphgram movement

⇒ (Homework)
- Record vowels - 10 times
  ↑ 10 different files
  ⇒ A E I O U
  ⇒ {अ आ इ ई उ ऊ ओं ं} (??)

Assignment 1: finding out difference bet? yes & no

→ build a speech recognition system which differentiate yes & no

→ Algorithm will be provided

→ how does yes & no look like chuk? ← it doesn't have

1 Logic: = ↓ has ssh... noise (hissing sound)

→ Clip off the noise

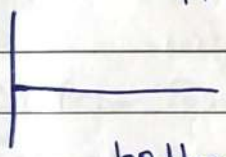
- Energy = $\left(\frac{1}{N}\right) \sum_{i=1}^{N} x_i^2$ ↑ sample height

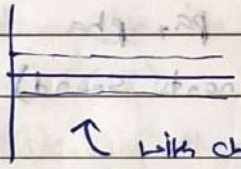
(↑ We can remove as well) (like N – fix to 100)

- Zero Crossing Rate (ZCR) = +ve to −ve : +1 or −ve to +ve : +1 } but if +ve to +ve : 0, −ve to −ve

20/08/25

Steps to take before recording

a) 
battery


↑ with charger

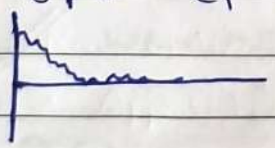
↑ Can be corrected by DC shift correction

(i) (find it out) before recording

So to substract

(ii) get a good mic

(iii) a big pulse before the actual recording


↑ ignore starting data

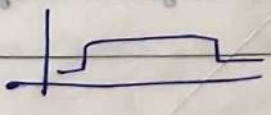
↑ no sound; why does it come?

⇒ When tap on Start button, the electromagnetic pulse results in that case

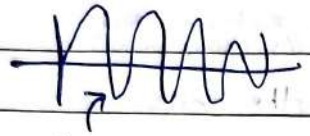
⇒ Cut it out & then throw it (like 1 sec etc)

(iv) finding out safe limit: not too low or too low (close to the mouth & away from nose) right near to chin

↓



(iv) Normalisation : (i) find higest or lowest no. in the records
of data
(ii) Scale it to ± 5000
(This is known as equal loudness)
→ (here we only control volume not signal distortion)



this can overflow on
(- - - - - -)

Different Types of Sounds & Their Characteristics

i) Vowels
→ how is a vowel pronounced — Quiz
→ Steady Sound { like tongue movement etc, alignment of vocal chord etc
→ regular pitch periods

• Condenser Microphone : in built power supply

ii) Consonants — pa, pha
→ plosives (consonants sound)

iii) Fricatives — /s/ , /sh/
→ no vibration of vocal chord
→ closure somewhere in the vocal track : noisy sound

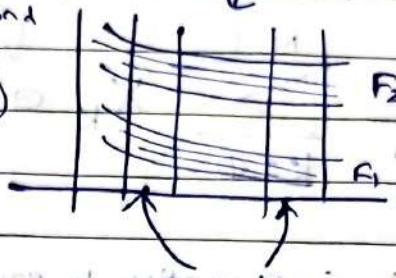
Voiced
iv) ^ Fricatives

v) Nasals — m, n, ɳ (ni)
→ both mouth & nose used
→ anti resonance : Waves coming from mouth & nose
if have same freq → & opposite in sign
⇒ they cancel each other
→ can be used for identification of ppl

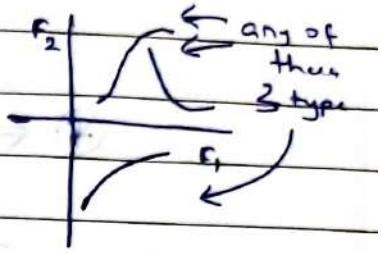
vi) Diphthong → two vowel sounds that are pronounced together to make one sound

- Sliding formant (??)

frequency



⇒ Plot F_1 & F_2 graph



(vii) Semi Vowel → aka leturals, (य ल व)

How to parametrically represent ⇄ waveform ⇄ LPC coefficient Speech Sounds?

i) Energy

ii) ZCR — extremely low in vowels, high in fricatives & plosives.

iii) Short term Spectral Envelope — instead of point eval, do in intervals

- Time Domain Based Approach
- Frequency Domain " "

} either one of them only used in deployment

↳ based on fourier transform.

→ give the frequency content of a signal

Discrete Fourier Transform (DFT)

For a discrete signal $x(n)$, if the signal is periodic with period N

↳ signals having finite values at fixed point of time

then $x(n)$ can be written as:

$$\tilde{x}(n) = x(n+N) \qquad -\infty < n < \infty$$

(signal repeats after N time)

(vowels have this property) — i.e periodic when N is periodicity

Then $\tilde{x}(n)$ can be represented by a discrete sum of sinusoids.

The fourier series representation of a periodic sequence is given by,

$$\tilde{X}(k) = \sum_{n=0}^{N-1} \tilde{x}(n)\, e^{-j\frac{2\pi}{N}nk} \quad \text{.....} (*)$$

{cubic operation}

(for a particular k we try to sum over n)

with this we can regenerate $\tilde{x}(n)$

$$\tilde{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}_k \, e^{j\frac{2\pi}{N} \cdot kn} \quad \text{---} \quad (**)$$

($\cos\theta + i\sin\theta$)

(*) $\Longleftrightarrow$ (**) : Convertible to each other

- Fourier Transform (FT) — (not a periodic signal)

The Fourier Transform representation of a signal $x(n)$ is given by :

(j fixed, w only varies)

$$X(e^{jw}) = \sum_{n=-\infty}^{\infty} x(n) \, e^{-jwn} \quad \text{---} \quad (1)$$

(not periodic hence used to observe empty signal)

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{jw}) \cdot e^{jwn} \, dw$$

where w — frequency

NOTE: if noise then in environment — FT not good like in case of fricative

A sufficient condition for the existence of FT is

$$\sum_{n=-\infty}^{\infty} |x(n)| < \infty$$

classmate
Date
Page

↙ gives the overall picture of the system

Z-Transform

- mathematical tool to convert discrete time signals (domain)

to

its Z-domain representation

- $X(z) = \sum_{n=-\infty}^{\infty} x[n]\, z^{-n}$ ← Time Domain Signal sampled periodically

~~Discrete Time Domain~~

Two sided Z transform eqn

~~Z transformation~~

$X(z) = \frac{z}{z-1}$ (Z domain)

(for an integrator system which outputs 1)

- Discrete Time Domain $\Longrightarrow$ $X(z) = \frac{z}{z-1}$

Z domain representation

Part 1

- How to solve Z-transform ~~eqn~~

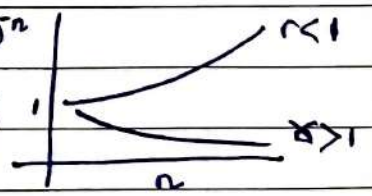
NOTE:

DFT $* r^{-n}$: makes it the Z-transform (a positive real no)

$\Rightarrow \sum_{n=-\infty}^{\infty} x[n] \cdot e^{-i\omega n} \cdot r^{-n}$

$z^{-n} = (e^{i\omega} \cdot r)^{-n}$

$\Rightarrow z = e^{i\omega} \cdot r$



- This r has convergence pt!

classmate
Date 26/08/25
Page

- The Z-transform

It is given by

The complete signal

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) \cdot z^{-n}$$

$$x(n) = \frac{1}{2\pi j} \int_{C} X(z)\, z^{+n-1}\, dz$$

→ Some complex domain

- If we replace $z = j\frac{2\pi}{N} k$ we get Fourier Transform
- depending upon 'z' we can get different transform (or signal)

- Fundamental frequency - first frequency which is maximum from the lower frequency side
  → It represents the vibration of vocal cord

Prosotics

- Time Domain Analysis of Speech
  - analysis of speech wrt time only (no frequency)
  - largely based on linear Predictive Coding (LPC)
    ↓
    for speech analysis
- Why only LPC?
  ⇒ Cellphones running this algo (very old; it actually works)
  i) it's a good model for periodic parts of speech
  ii) Good Source; Vocal tract Separation
    ↳ lung system / pressure + functioning of vocal chord
    remaining part of vocal tracts & articulators
  iii) it's a mathematically stable model - 0 chance of system crashing

iv) Prosus method (it is)

Deriving LPC

let $s(n)$ be the speech signal at time point n, this can be approximated as a linear combination of past 'p' samples such that,

$$s(n) \simeq a_1 s(n-1) + a_2 s(n-2) + \dots + a_p s(n-p) \quad \dots (1)$$

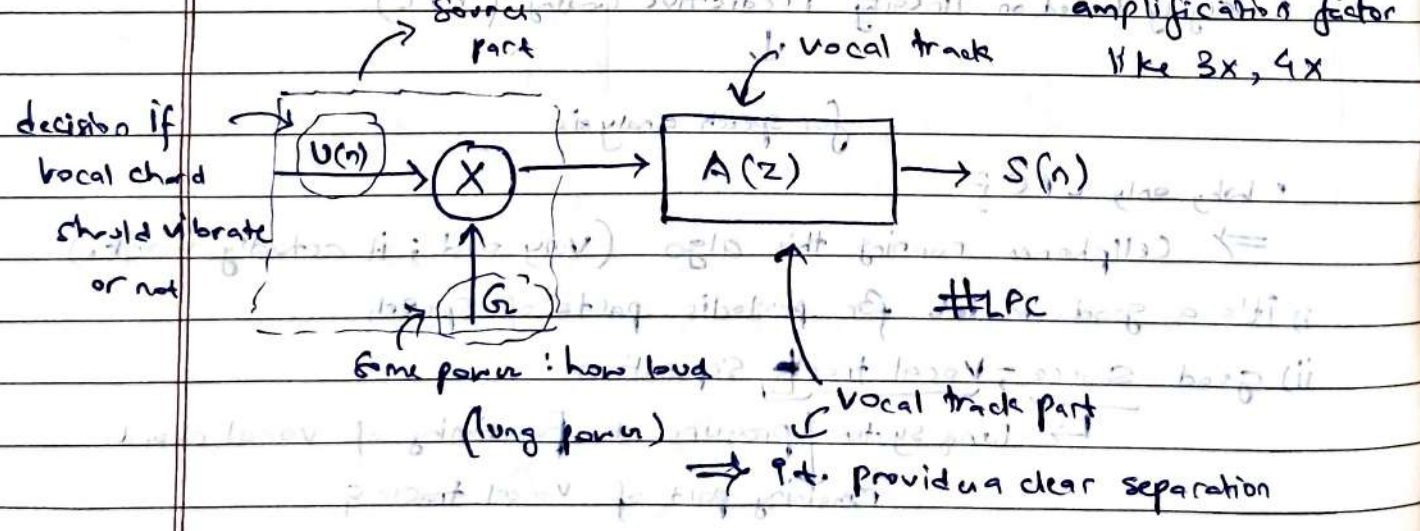
p ← for predicting this, we need past p samples

Where a_i, $(i = 1, 2 \dots p)$ are constants now convert eqn (1) to a equality by including a term called excitation term given by, $-G \cdot v(n)$ giving

$$s(n) = \sum_{i=1}^{p} \underbrace{a_i s(n-i)}_{A(z)} + G \cdot v(n) \quad \dots (2)$$

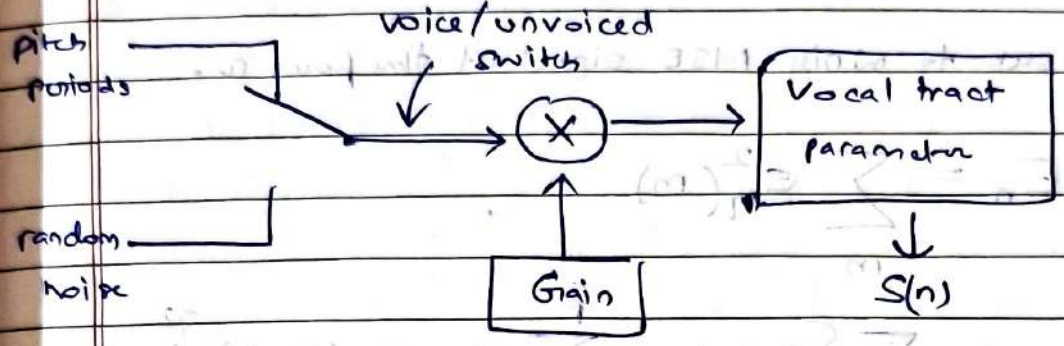
(determined for that analysis frame)

Where $v(n)$ is the normalized excitation & G is the gain term ← amplification factor like 3x, 4x



⇒ i.e. provides a clear separation

Vocoders - read little bit about it (A para)



Let us consider,

$$\tilde{S}(n) = \sum_{i=1}^{p} a_i (s(n-i)) \quad \cdots\cdots \quad (3)$$

(Assuming no source G. $u(n) = 0$)

$\tilde{S}(n)$ → not absolute speech signals

Now the prediction error will be given by,

$$e(n) = S(n) - \tilde{S}(n)$$

$$= S(n) - \sum_{i=1}^{p} a_i S(n-i)$$

for this

how much signals should we analyze?

- We need to analyze the error for a short segment of the speech signal

$S(n)$ is fixed

a_i is variable to make $e(n)$ low

$S(n-i)$ is fixed

- Now let us take: $S_n(m) = S(n+m)$

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

m tells the range of analysis

- Now we seek to minimise MSE signal at time point n, i.e

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

$$= \sum_m \left\{ s_n(m) - \sum_{k=1}^{p} a_k s_n(m-k) \right\}^2$$

- p quantities unknown (a_k)
- p equations {since we want to minimise} $\Rightarrow$ differentiate

Differentiating partially wrt each a_k & setting it equal to zero

$$\frac{\partial E_n}{\partial a_i} = 0 \text{ , will give}$$

$$\Rightarrow \frac{\partial (E_n)}{\partial a_i} = \frac{\partial}{\partial a_i} \left[\sum_m \left\{ s_n(m) - \sum_{k=1}^{p} a_k s_n(m-k) \right\}^2 \right] = 0$$

$$= \frac{\partial}{\partial a_i} \sum_m \left[s_n^2(m) + \left\{ \sum_{k=1}^{p} a_k s_n(m-k) \right\}^2 - 2 s_n(m) \cdot \left\{ \sum_{k=1}^{p} a_k s_n(m-k) \right\} \right] = 0$$

$$= 0 + \frac{\partial}{\partial a_i} \sum_m \left(\sum_{k=1}^{p} a_k^2 s_n^2(m-k) + 2 \sum_{\substack{k=1 \\ k \neq i}}^{p} a_i a_k s_n(m-i) \cdot (s_n(m-k)) \right)$$

$$= \sum_m \left[2 s_n(m) \cdot \frac{\partial}{\partial a_i} \left\{ \sum_{k=1}^{p} a_k s_n(m-k) \right\} \right.$$

these 2 can be clubbed

$$= \sum_m \not{2} a_i (S_n^2 (m-i)$$

$$+ \not{\phi} \sum_m \sum_{\substack{k=1 \\ k \neq i}}^{p} a_k S_n(m-i) (S_n(m-k)$$

$$= \sum_m \not{2} S_n(m) (S_n(m-i)$$

$$\Rightarrow \sum_m \sum_{k=1}^{p} a_k S_n(m-i) \cdot S_n(m-k)$$

$$= \sum_m S_n(m) \quad S_n(m-i)$$

$$= \sum_m S_n(m-0) S_n(m-i)$$

$$\Rightarrow \sum_{k=1}^{p} a_k \phi_n(i,k) = \phi_n(0,i) \quad \text{---} (4)$$

where $\phi_n(i,k) = \sum_m S_n(m-i) \cdot S_n(m-k)$

how to define p?

$\Rightarrow$ **p = 2 to 12**

$i = 1$ to p

p eqns — how many unknowns? — p of them (a_i)

How to find a_i's

classmate
Date 28/09/25
Page

- Autocorrelation Method (aka short term Speech analysis)

→ To find a_i's

i) In this method, we take, $0 \leq m \leq N-1$ ($\Rightarrow$ N speech values) (normally N = 320 samples)

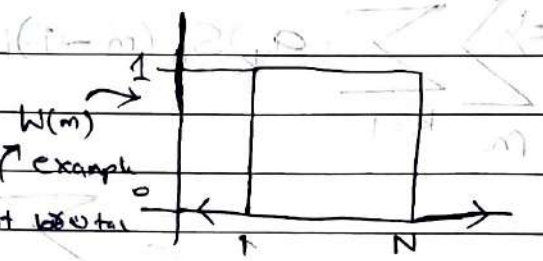
$\Rightarrow$ determinant of cofactor matrix > 0

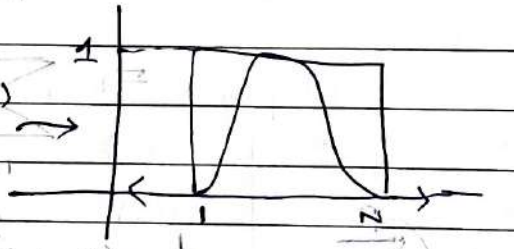
ii) Signal ceases to exist beyond the interval i.e available in the frame only

$$S_n(m) = w(m)\, S(m+n)$$

Controls signal when $m < 0$ or $m \geq N$

aka windowing function





We can now write,

$$\phi_n(i,k) = \sum_{m=0}^{N-1} S_n(m-i)\, S_n(m-k)$$

does the signal ever reach to N-1? (No.)

$\Rightarrow$ as it ceases to exist

$\Rightarrow$ therefore add p : {N-1+p} limit of m

put $m - i = y$

$$\phi_n(i,k) = \sum_{y=0} S_n(y)\, S_n(y+i-k)$$

as $y = -i$ to $(N-1+p-i)$

but not significant $S_n(0)$

but not significant

classmate
Date
Page

$y = \underbrace{-i}_{\downarrow 0}$ to $\underbrace{N-1+p-i}_{\downarrow N-1-(i-k)}$

↑ as beyond $(i-k)$ something not significant

$$\phi_n(i,k) = \sum_{y=0}^{N-1-(i-k)} s_n(y)\, s_n(y+i-k) \quad \text{— (6)}$$

↳ (Autocorrelation fn of $s_n(y)$ at a lag of $i-k$)

Correlation: Degree of agreement bet² two sets

$X:$ $Y:$ } 2 sets

$$R(X,Y) = \frac{1}{N}\sum_{i=1}^{N} x_i y_i$$

↑ random variable; ↑ (actual phenomenon value)

Case 1: If y replaced with X,
→ $R(X,X)$ is energy (possible value) { high value for +ve correlation; low ~~negative~~ " for -ve " }
→ it is the maximum ^ called energy

envelop

Case 2: X_1: o o o o o o o o o o

X_2: o o o o o o o o

$$R(X_1,X_2) = \frac{1}{N}\sum_{i=1}^{N} x_i x_{i+1}$$

↑ Autocorrelation of lag 1

with lag = 0 ⇒ energy

⇒ lag → if increased

Autocorrelation → ⊕ ← pitch period / frequency

↑ decreases

Now we can write (6) is the short term autocorrelation function with lag 'i-k' i.e, $\phi_n(i,k)$

$$\phi_n(i-k) = R_n(i-k) \quad \text{(put modulus operator)}$$
$$= R_n(|i-k|)$$

It is an even function

Finally eqn (4) becomes,

$$\sum_{k=1}^{p} a_k R_n(|i-k|) = R_n(i) \quad i = 1 \ldots p$$

In matrix notation, we get

$$\begin{bmatrix} R_n(0) & R_n(1) & R_n(2) & \cdots & R_n(p-1) \\ R_n(1) & R_n(0) & \cdots & \cdots & R_n(p-2) \\ R_n(2) & R_n(1) & R_n(0) & \cdots & R_n(p-3) \\ \vdots & & & & \\ R_n(p-1) & R_n(p-2) & \cdots & \cdots & R_n(0) \end{bmatrix}_{p \times p} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix}_{p \times 1} = \begin{bmatrix} R_n(1) \\ R_n(2) \\ \vdots \\ R_n(p) \end{bmatrix}_{p \times 1} \quad -7$$

- Cofactor matrix
- dimensionality : $p \times p$ (p by p, not p cross p)

(P1) Symmetric matrix : So can be solved efficiently

(P2)
- diagonal values all same
- next diagonal values — non principal all same

→ Choose any diagonal — all elements same

(P3)
- how many unique elements ? — p
- total no. of elements — p^2

⇒ space footprint reduced

Time complexity = $p^3 \longrightarrow p^2$ (we need) - We use now!!

What is that algorithm which reduces from p^3 to p^2?

- This particular Cofactor matrix has got a special name. Called, Toeplitz matrix.

A very efficient soln to the set of normal equation is provided by Levinson Durbin's Algorithm (can check derivation)

↳ prove it 'emphirically' using $p=2$, $p=3$

(Algorithm)

Initialize → $E^0 = R(0)$ ← energy

$$k_i = \left\{R(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R(i-j)\right\} / E^{(i-1)}$$

$1 \le i \le p$

$\alpha_i^{(i)} = k_i$ ← 2nd matrix

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

$$E^{(i)} = (1 - k_i^2) E^{(i-1)}$$

by law of Induction, it is gonna be true

$\Rightarrow R(0) =$ energy $= E^0$ # Hardcode it

$$k_1 = \frac{R(1)}{R(0)}$$

$\alpha^{(p)} \rightarrow$ soln

Homework

Do for $p=2, 3$

- T.C $= p^2$

Methods to solve/find LPC coefficients

M-1 The Autocorrelation Method

- Levinson Durbin Algorithm
  - error det is not singular
  - if det = 0, error not divided by 0

linear pbs — bottleneck with inverse calculation → in eqn (7)

- When will the big matrix of left is 0?
  ↳ if R(0) = 0 ⇒ Energy = 0 ? no signal at all
  R(0) very small ? ignore it
  ≈ $10^{-4}, 10^{-3}$ —

Normally we will take p = 12 (13) → R(0) to R(12)

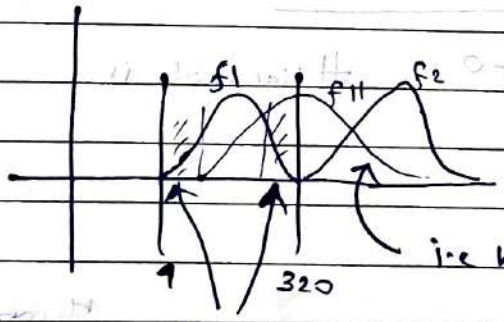
N = 320 (frame size) (20 msec)

Why?

Sampling rate = 16000 Samples/sec

16 bits/sample

Skip = 80 samples → slide by 5ms



i.e why overlapping we should do just to avoid loss
⇒ gives reliable answer
(take 25%)

Exercise

Consider p = 2,

$R = \{ r(0), r(1), r(2) \}$

$a_1, a_2 = ?$

To compute a_1 & a_2 solve this pb using cross multiplication

classmate
Date
Page

method & the levinson Durbin Method.

⇒ eqn for cross multiplication: $\begin{bmatrix} r(0) & r(1) \\ r(1) & r(0) \end{bmatrix} \begin{bmatrix} a[1] \\ a[2] \end{bmatrix} = \begin{bmatrix} r(1) \\ r(2) \end{bmatrix}$

M-2 Covariance Method → (Signal is untouched)

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

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

now we get,

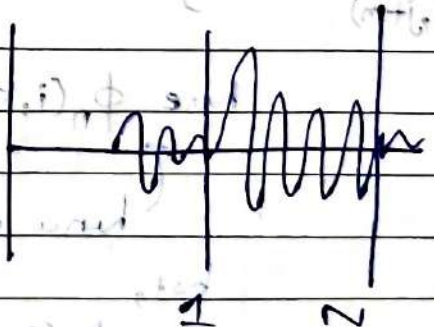
$$\phi_n(i,k) = \sum_{m=-i}^{N-1-i} s_n(m)\, s_n(m+i-k)$$

p extra req ←

or

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

We need a few samples which is prior to the interval under consideration.



now the eqn becomes,

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

$-p \le m \le N-1$

as i & k bounded by p

$i = 1, \ldots, p$

This gives

$$\begin{bmatrix} \phi_n(1,1) & \phi_n(1,2) & \cdots & \phi_n(1,p) \\ \phi_n(2,1) & \phi_n(2,2) & \cdots & \phi_n(2,p) \\ \vdots & & & \\ \phi_n(p,1) & \phi_n(p,2) & \cdots & \phi_n(p,p) \end{bmatrix} \cdot \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_p \end{bmatrix}$$

This is the new cofactor matrix

$$= \begin{bmatrix} \phi_n(1,0) \, i \\ \phi_n(2,0) \\ \vdots \\ \phi_n(p,0) \end{bmatrix}$$

NOTE: $\phi_n(1,1), \phi_n(2,2) \cdots \phi_n(p,p)$ won't be same now

$\phi_n(1,2)$ & $\phi_n(2,1)$ will it be same?

→ yes (identically)

as signal range changes if we do $(i+m, j+m)$

- Symmetric matrix — but not toeplitz

here $\phi_n(i,k) = \phi_n(k,i) \quad \forall i, k = 1, \ldots p$ $i \neq k$

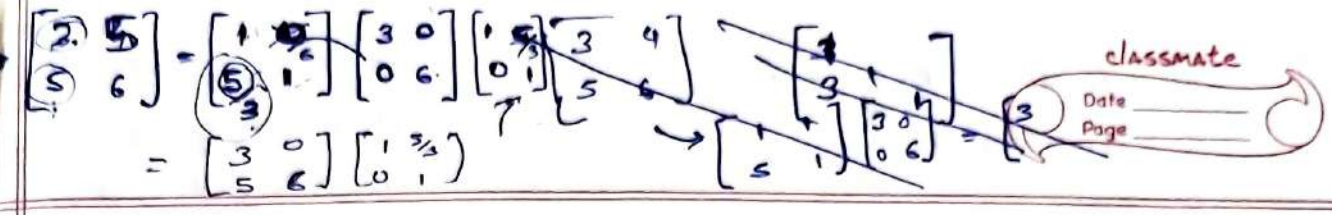
(hence cofactor matrix is symmetric)

but,

$\phi_n(i,j) \neq \phi_n(i+k, j+k)$

⇒ not toeplitz

T.C $= (n^3)$



M-3 Cholesky's Decomposition Method — apply to only symmetric positive definite matrices

$$\underset{p\times p}{\phi}\ \underset{p\times 1}{\alpha} = \underset{p\times 1}{\psi}$$

→ pivotal part of the algo

factorise ϕ as: $\phi = VDV^T$ where

how to do it? → search it up

$V \rightarrow$ lower Δ matrix

$D \rightarrow$ diagonal "

We get:

$$(\underset{p\times p}{V}\underset{p\times p}{D}\underset{p\times p}{V^T})\,\alpha = \psi$$

Let

$$DV'\alpha = y \leftarrow \text{has the variables}$$

Thus,

$$Vy = \psi$$

$$\begin{bmatrix} - & 0 & 0 & 0 \\ - & - & 0 & 0 \\ - & - & - & 0 \\ - & - & - & - \end{bmatrix}$$

→ (Since first row only 1 val ⇒ q_1 will be found & so remaining elements one at a time)

→ in worst case (n^3)

$$L = \begin{bmatrix} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{bmatrix} \quad d = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix}$$

$$d_k = a_{kk} - \sum_{j=1}^{k-1} L_{kj}^2 d_j \quad \text{(kth entry)}$$

$$L_{ik} = \frac{1}{d_k}\left(a_{ik} - \sum_{j=1}^{k-1} L_{ij} L_{kj} d_j\right)$$

$i > k \qquad k = 1, 2, \ldots, n$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 7 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 4 & 5 & 0 \\ 7 & 12 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

$a_1 x + b_1 y = c_1$

$a_2 x + b_2 y = c_2$

$a_1 x + b_1 y + c_1$

$a_2 x + b_2 y + c_2$

$b_1 \quad c_1 \quad a_1 \quad b_1$

$b_2 \quad c_2 \quad a_2 \quad b_2$

$x \qquad y \qquad 1$

$b_1 \quad x \quad c_1 \quad a_1 \quad b_1$

$b_2 \quad c_2 \quad a_2 \quad b_2$

$\frac{x}{b_1 c_2 - b_2 c_1} = \frac{y}{-c_1 a_2 + c_2 a_1} = \frac{1}{a_1 b_2 - a_2 b_1}$

$r(0) = n^2$

$r(1) = n(n-1)$

$r(2) = n(n-2)$

$E_0 = R(0)$

$K_i = R(i) - \sum_{j=1}^{i-1} \alpha_j^{i-1} R(i-j) \Big/ E^{i-1}$

$\alpha_i^i = k_i$

$\alpha_j^i = \alpha_j^{i-1} - k_i \alpha_{i-j}^{i-1}$

$i = 1, 2$

$j = 1 - i - 1$

$E_i = (1 - k_i^2) E_{i-1}$

$E_0 = R(0)$

$K_1 = \frac{R(1)}{R_0}$

$\alpha_1^1 = k_1 = \frac{R(1)}{R_0}$

$\alpha_0^1 = \alpha_0^0 - k_1 \alpha_1^0$

$E_1 = (1 - k_1^2) E_0$

$= (1 - \frac{R(1)^2}{R_0^2}) E_0$

$= R_0 [1 - \frac{R(1)^2}{R_0^2}]$

$i = 2 \quad j = 1$

$K_2 = R(2) - \alpha_1^1 R(1) \Big/ E_1$

$\alpha_2^2 = K_2 = R(2) - R(1)^2 \Big/ R(0)(1 - \frac{R(1)^2}{R_0^2})$

$\alpha_2^2 = K_2$

$\alpha_1^2 = \alpha_1^1 - K_2 \alpha_1^1$

$= (1 - K_2) \alpha_1^1$

$= R(1) \left(1 - R(2) + \frac{R(1)}{R(0)(1 - R(1)^2/R_0^2)}\right)$

$$\alpha_1 = R(1)\left(1 - R(2) + \frac{R(1)^2}{R(0)\left(1 - \frac{R(1)^2}{R_0^2}\right)}\right)$$

classmate
Date
Page

$$\alpha_2 = R(2) - \frac{R(1)^2}{R(0)\left(1 - \frac{R(1)^2}{R_0^2}\right)}$$

LD

$$= \frac{R(0)R(2) - R(0)R(2)R(1)^2 - R(1)^2}{R(0)\left(1 - R(1)^2\right)}$$

$$R(0)\alpha_1 + R(1)\alpha_2 = R(1)$$

$$R(1)\alpha_1 + R(0)\alpha_2 = R(2)$$

$$\frac{\alpha_1}{R(1)R(2) - R(1)R(0)} = \frac{\alpha_2}{R(1)^2 - R(0)R(2)} = \frac{1}{R(0)^2 - R(1)^2}$$

$$\alpha_1 = \frac{R(1)(R(2) - R(0))}{R(0)^2 - R(1)^2} \qquad \alpha_2 = \frac{R_1^2 - R(0)R(2)}{R(0)^2 - R(1)^2}$$

$1 - k_2$ k_2

$$R(0) = \frac{1}{N}\sum x_i^2 \qquad R(1) = \frac{1}{N}\sum x_i x_{i+1}$$

$$R(2) = \frac{1}{N}\sum x_i x_{i+2}$$

$$1 - \frac{R_1^2 - R_0 R_2}{R_0^2 - R_1^2} = \frac{R_2 - R_0}{R_0^2 - R_1^2}$$

$$R_0^2 - R_1^2 - R_1^2 + R_0 R_2 = R_2 - R_0$$

$$R_0(R_0 + R_2) - 2R_1^2 = R_2 - R_0$$

$$R_0(R_0 + 1) - 2R_1^2 + R_2(R_0 - 1) = 0$$

$$\alpha_1 = R_1\left(1 - R_2 + \frac{R_1^2 R_0}{(R_0^2 - R_1^2)}\right)$$

$$R_0^2 - R_1^2 - R_2 R_0^2 - R_2 R_1^2 + R_1^2 R_0$$

CM $\alpha_1 = \frac{-R_1 (R_2 - R_0)}{R_0^2 - R_1^2}$ $\alpha_2 = \frac{-R_1^2 + R_0 R_2}{R_0^2 - R_1^2}$

LD $i=1$ $E_0 = R_0$

$j=1$

$$K_1 = \frac{R_1}{E_0} = \frac{R_1}{R_0}$$

$$\alpha_1^1 = K_1 = \frac{R_1}{R_0}$$

$$E_1 = (1 - K_1^2) E_0 = \left(1 - \frac{R_1^2}{R_0^2}\right) R_0$$

$$= \frac{R_0^2 - R_1^2}{R_0}$$

$i=2$ $j=1$

$$K_2 = \frac{R_2 - \alpha_1^1 R_1}{E_1}$$

$$= \frac{R_2 R_0 - R_1^2}{R_0^2 - R_1^2}$$

$\alpha_2^2 = K_2$ ✓

$$\alpha_1^2 = \alpha_1^1 - K_2 \alpha_1^1$$

$$= \alpha_1^1 (1 - K_2)$$

$$= \frac{R_1}{R_0} \left(1 - \frac{R_2 R_0 - R_1^2}{R_0^2 - R_1^2}\right)$$

$$= \frac{R_1}{R_0} \left(\frac{R_0^2 - R_0 R_2}{R_0^2 - R_1^2}\right)$$

24 y. ea
1
$\frac{5}{1} + a = 5$

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2/7 & 1 & 0 \\ 3/7 & \frac{35}{24} & 1 \end{bmatrix} \begin{bmatrix} 7 & 0 & 0 \\ 0 & \frac{24}{7} & 0 \\ 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} 1 & 2/7 & 3/7 \\ 0 & 1 & 35/24 \\ 0 & 0 & 1 \end{bmatrix}$$

$$d_{ik} = a_{ik} - \sum_{j=1}^{k-1} L^2_{kj} d_j$$

$$L_{ik} = \frac{1}{d_k}\left(a_{ik} - \sum_{j=1}^{k-1} L_{ij} L_{kj} d_j\right)$$

$i > k$

$$\begin{bmatrix} 7 & 0 & 0 \\ 2 & \frac{24}{7} & 0 \\ 3 & \frac{5}{a} & 6 \end{bmatrix} \begin{bmatrix} 1 & 2/7 & 3/7 \\ 0 & 1 & 35/24 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\frac{4}{7} + x = 4$$

$$x = 4 - \frac{4}{7}$$

Req symmetric matrix

$$\begin{matrix} 7 & 2 & 3 \\ 2 & 4 & 5+x/7 \\ 3 & \frac{6+5}{7} & \end{matrix}$$

$$y\frac{24}{7} = 5 - \frac{6}{7} \quad \frac{9}{7} + \frac{5\times 35}{24} + 6$$

$$y = \frac{35}{24}$$

$$\Rightarrow \begin{bmatrix} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{bmatrix} \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix}$$

$$\Rightarrow 4 - \frac{(4)^2}{7}$$

$$4 - \frac{16}{7}$$

$$\Rightarrow \frac{28-16}{7} = \frac{12}{7}$$

$$\Rightarrow L_{21} = \frac{1}{a_{11}}\left(a_{21} - \sum_{j=1}^{0}\right) = \frac{2}{7} \qquad d_{11} = a_{11} = 7$$

$$L_{31} = \frac{1}{a_{11}}\left(a_{31} - \sum_{j=1}^{0}\right)$$

$$= \frac{3}{7}$$

$$d_{22} = a_{22} - \sum_{j=1}^{1} L^2_{21} d_1$$

$$= a_{22} - \left(\frac{a_{21}}{a_{11}}\right)^2 a_{11}$$

$$= a_{22} - \frac{a_{21}^2}{a_{11}}$$

$$= 4 - \frac{4}{7} = \frac{24}{7}$$

$$L_{32} = \frac{7}{24}\left(3 - \sum_{j=1}^{1} L_{31} L_{21} d_{22}\right)$$

$$= \frac{7}{24}\left(3 - \frac{3}{7}\times\frac{2}{7}\times\frac{24}{7}\right)$$

$$= \frac{21}{24} - \frac{3\times 7\times 2\times 24}{7\ 24\ 7\ 7} = \frac{21}{24} - \frac{6}{49} = \text{X}$$

(Speed) → (Better)!!

classmate Date 03/09/25 Page

- Analysis for LPC — Durbin — Cholesky

i) Storage

	Durbin	Cholesky
Samples	N	N+p
Matrix	$\propto p$ (proportional to)	$\propto p^2/2$ (since symmetric matrix)
Windowing	N (prestore (extra Storage))	0 (no clamping req)

ii) Multiplications

	Durbin	Cholesky
Windowing	N	0
Correlation	$\propto N \cdot p$	$\propto N \cdot p$

iii) Matrix Solution — $\propto p^2$ — $\propto p^3$

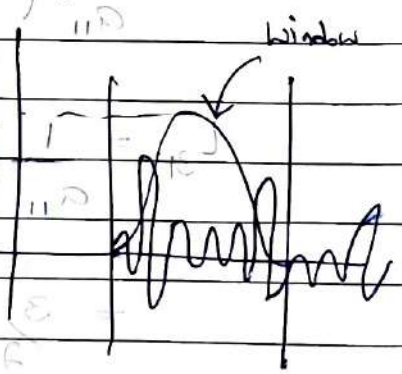
- Another benefit of Durbin
  1) No divide by 0
  ~~ii)~~ ~~X(z) is a transformation of~~ signal)

- What is clamping window?
  → we shall use a window called hamming window.

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

= 0, otherwise

(Plot the graph for this)



NOTE: pre compute & store is here N will be fixed

- Cost a lot of computation!!

What happens when we apply hamming window? → the signal gets compressed on both side → Why necessary?

M-1 • FT

M-2 • LPC a_i $i = 1 \ldots p$

↑ What is the use of a_i ? → (can we generate S ?) : regenerate speech

classmate Date Page

Speech Regeneration

→ Take the signal of N = 320

→ Apply the hamming window

→ find auto correlation

→ Apply levinson - Durbin

→ Take first (p) as it is : generate the (p+1) sample
↑ i.e 13 th sample

(*) → keep the 13th (synthetic sample) in array

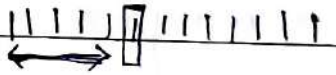
→ Skip the first sample, take next (12) → generate the 14th sample

→ Continue (*) till N = 320 reached : (Synthetic frame will be generated !)

- listen to it ←
- data too small : so nothing

So plot both the original & synthetic samples !!

→ check where varying



NOTE : use original data to generate signal & hearing also !! ← if use synthetic → will be bad !!

M-3 C_i : Cepstral Coefficient ← $C(m)$

↳ LPC coefficient converted to this

$$C(0) = \log \sigma^2$$ where σ is the gain term of the LPC model

Gain = R(0)

$$C(m) = a_m + \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) C_k a_{m-k} \qquad 1 \le m \le p$$

$$= \sum_{k=m-p}^{m-1} \left(\frac{k}{m}\right) C_k a_{m-k} \;, \qquad m > p$$

Why beneficial ?

→ source is intact (whereas previously in speech regeneration source not considered)

→ all a_i included

C_i ; $i = 1, 2, \ldots N$

(we don't use $Q = \frac{3}{2}p$ as well) ← • We are restricting to 13 values per frame

The cepstral coefficient have the following properties:

(i) The lower order C-C are sensitive to the spectral slope

(ii) The higher order " are " to noise, or noise like sounds

In order to diminish the sensitivity, a tapered window is used,

$$w(m) = \left\{1 + \frac{Q}{2} \sin\left(\frac{\pi m}{Q}\right)\right\} \quad 1 \le m \le Q$$

(after calculating w(m) apply it on C(m) giving (p=12) samples not Q)

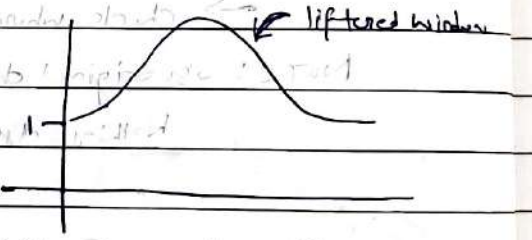
We do, $C(m)\, w(m)$ $m = 1 \ldots Q$

The distortion of signal

: here we are only amplifying it → called raised sine window

⇓ floating point (12 64-bit nos)

This process is sometimes called liftering (opposite of filtering)



- here we allow some part (ie like low freq or high freq)
- but in liftering → (allows all as well as amplify it) – No data loss

→ Mid-sem ←

- No more classes

→ T/F

NTF: Matching

FIB

MCQ

SQ

LQ

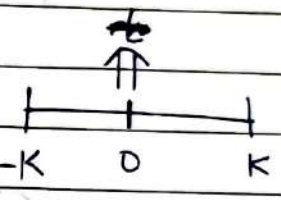
classmate
Date 05/05/25
Page

feature set
Static ^ properties — dynamic

- Instead of 320 samples → $C_0\ C_1\ C_2 \dots C_p$ $\quad \Delta C_0\ \Delta C_1 \dots \Delta C_p$
  $p = 12$
  # Orthogonal polynomial ; how to do it ⊗
  What is the use ✓
  ↑ how C_i changing with time is also considered

- The derivatives or Delta cepstrum is approximately given by,

$$\frac{\partial}{\partial t} C_m(t) = \Delta C_m(t) \simeq \mu \sum_{k=-K}^{K} k \cdot C_m(t+k)$$



This is computed using orthogonal polynomial fit.
Where μ is an appropriate normalisation factor & $(2K+1)$ is the no. of frames over which the computation is performed. (This is computed for current frame & from from back & front)

Usually K is treated as 3 (for analysis = 7)
$\mu = 0.375$ (empirically found)

The complete feature set also includes 13 $\Delta^2 C_i$ or acceleration coefficients.

$[C_0 \dots C_p]\ [\Delta C_0 \dots \Delta C_p]\ [\Delta^2 C_0 \dots \Delta^2 C_p]$ = 39 dimensional space

NOTE: feature vector for speech — industry set

Write a para on, ΔC_p & $\Delta^2 C_p$

Q What can we do with cepstral coefficient?

Assignment 3
Extension → vowel recognition

How to differentiate bet^n 2 speech

- Distance Measures

(Too easy) (i) Euclidean Distance

~~The first distance is the euclidean d~~

$$D_{CEP} = \sum_{i=1}^{Q} \{C_i(t) - C_i(r)\}^2$$

Q → Q is P for our case

(Kepstral) ↑

$t \rightarrow$ test data

$r \rightarrow$ reference data

$\boxed{P = 12}$

- Take 1 frame
- Compute cepstral coefficients
- Store it

} # r

Do the same for test data

→ Repeat for remaining frames

if distance is low ⇒ Same word speech !!

In this distance computation, equal weightage is given to each component.

It is easy to implement & fairly straight forward.

(Too good) (ii) Mahalanobis Distance

- PC Mahalanobis : Write a short note on him (Mid sem)

$$D_{M_{CEP}} = \underbrace{\left(\tilde{C}_t - \tilde{C}_r\right)'}_{1\times Q} \underbrace{V^{-1}}_{Q\times Q} \underbrace{\left(\tilde{C}_t - \tilde{C}_r\right)}_{Q\times 1}$$

↑ transpose

↑ vector

Where,

V = Covariance matrix of test feature vector (C_i's)

NOTE: Inverse take (P^3)

Variance → how far or close
↑ high ↑ small

properties :
i) it is extremely accurate (computation itself wrong)
ii) difficulty in computing the inverse — (cubic) & underflow pb
iii) $I - \{$ ↑ if diagonal, els → 0 $\}$ (P)

(iii) Tokhura's distance
- Take variance $_{i \neq j} = 0$
- He suggested using only the digonal elements of V,

$$D_{T_{CEP}} = \sum_{i=1}^{8} w(i) \left[C_i(t) - C_i(r) \right]^2$$

where,
w(i) is the inverse of the variance of the ith element (computed for test data)

$$w(i) = \frac{1}{\sigma_i^2}, \quad \sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2$$

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

अ - अ | ए - ए - इ - ई ओ - ओ उ - ऊ

classmate Date 10/09/25 Page

Distinguishing speech based on a_i, using prev method not good ∴,

⇒ Distance Measure based on LPCs (a_i's)

(i) Itakura Minimum Prediction Residual Distance

$$D_{I_{LPC}} = \frac{a_r' R_t a_r}{a_t' R_t a_t} - 1$$

error

(here most optimal) a_r → LPCs of ref

⇒ if $R_t = R_p$ ⇒ error is minimum a_t → LPCs of test

(NOTE: $a_r' R_t a_r$ is also error but of reference data) R_t → Correlation matrix

(ii) Itakura Saito Distance

$$D = \left\{ \frac{a_r' R_t a_r}{a_t' R_t a_t} - 1 \right\} + \log \left[\frac{G_r^2}{G_t^2} \right]$$

source component

$$G = \sqrt{a' R a}$$

Assignment Steps

(i) Calc DC shift - with power before reading

(ii) normalise — to ±5000

(iii) preceding & succeeding frames — Own Algo based on ZCR & Energy

(iv) Steady frame

- for each frame calc R_i, a_i, C_i
- shift by - 80
- apply hamming window

(long double)

v) Best sounding vowels (reference C_i) (5)

take distance from test data (----)

from all 5 of ⇒ whichever min (is the same one) ∴ find recognition score

classmate
Date
Page

- NOTE: Don't write the distance calculation code
  → use excel sheet : Put C_0, C_1, & formula's not
  → calc there

24/09/25

How to prove these 12 values are useful
→ vowel recognition } — by calculating the distance!!
→ regenerate the waveform ↳ Accuracy - 85%

LPC
Cepstral Coefficients

Cepstrum : The fourier transform of a signal $x(n)$ is given by,

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j 2\pi/N\, nk} \qquad 0 \le k \le N-1$$

(gives fourier coefficients)

Let,

$$\hat{X}(k) = \log\left[X(k)\right]$$

($\log[\cdot]$ → never neg so take mod $(X(k))$ or only real part)

apply FFT^{-1} on this → original signal ?

Cepstrum is defined as,

$$\hat{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{X}(k) * e^{j 2\pi/N\, kn} \qquad 0 \le n \le N-1$$

($\hat{x}(n)$ ← IDFT → Inverse discrete fourier transform ; $\hat{X}(k)$ ← {well behaves})

Cepstrum is defined by, {read on it}

$$C(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log\left|X(e^{-jw})\right| e^{jwn}\, dw \qquad -\infty < n < \infty$$

• {Signal → FFT → spectrum (Time domain) → mutate it → apply FFT⁻¹ → capstrum}

~~FFT~~ ~~spectrum~~ ~~{Capstrum}~~

$\hat{X}(k)$ approximately behaves like $X(k)$

↑ sampling rate

Example 1 # raw signal : 10 KHz (10000 samples in 1 sec)

each sample : 16 bits

⇒ 160,000 bps

Capstral analysis ← {let's say p = 10, 100 analysis frames

16 bits vector computation

⇒ 16,000 bps — data rate

NOTE: if we use original data ⇒ 160000 bps

if we use capstral {compression} ⇒ 16000 bps

data rate: i.e 10:1

{how to improve this}

$\pi = \frac{22}{7}$ = {scalar quantization} ⇒ 3.14

(representing decimal in small nos.)

Vector Quantization

$\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{12} \end{bmatrix}$ ⟹ 1 integer

• we would require a 'code book' of approximately 1024 entries (25 variants of 40 basic speech sounds)

{25 variants of 1 speech sound, next 25 for another speech sound}

{# each row is called a Code word / template}

size of codebook

levels

{# here recognition can be done again by computing E.D & checking distance from the table ⇒ & reporting associated speech sound}

Book: Gersho & Gray for vector Quantization (either Ch1, Ch2)

• constituent of codebook

{Granted example 1}

In this situation, there will be 100 vectors per sec

$\Rightarrow$ 1000 bits/sec : as each row by 10 bits

$\Rightarrow 10 \times 100$

↑ Throughput achieved by vector quantization (No. of data bits dealing) ← hence

NOTE: here 1 vector = 1 frame containing all c.c

$\Rightarrow$ to codebook assigns 10 bits/vector & we analyse 100 vector per sec

Advantages:

(i) less storage space is used

(ii) reduced computation for determining similarity of spectral vectors

(iii) discrete representation of speech signals

Drawbacks:

(i) spectral distortion is introduced {(E != 0) error}

(ii) here quantization error is strictly greater than 0

(iii) Storage of codebook vectors may not be non-trivial for large sized codebooks

{Why $\Rightarrow$ NOTE: most of important info in speech is in the spectral c.c reduce it to envelope (formant) not in every sample 1/10}

↓

c.c analysis capture this envelope using just few coefficients

↑ performs (i.e.p)

compressed form representation

Defn: It was proposed by Gray in 1984, Makhoul et al 1985

Let $\tilde{x}$ be K-dimensional vector whose components are real valued random values.

A vector $\tilde{x}$ is mapped onto another K dimensional vector $\tilde{y}$

$\tilde{x} \longrightarrow \tilde{y}$

where $\tilde{y} = q(\tilde{x})$

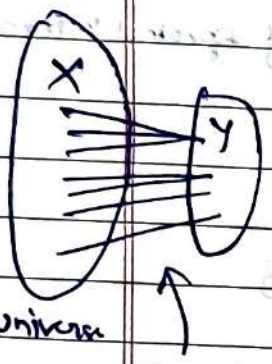
That is $\tilde{y}$ takes on one of a finite set of values,

$Y = \{ \tilde{y}_i \}$ $i = 1, 2, \ldots, K$ ← no. of levels

The set of vectors Y is called the codebook & each $\tilde{y}_i$ is called a codeword or a template.

The size K of the codebook is referred to as the no. of levels

When $\tilde{x}$ is quantized to $\tilde{y}$, a quantization distortion measure, $D(\tilde{x}, \tilde{y})$ can be defined between $\tilde{x}$ & $\tilde{y}$



The over all avg distortion is then represented by,

$$D = \lim_{M \to \infty} \frac{1}{M} \sum_{m=1}^{M} d(x(n), y(n))$$

Distortion ↑ distance ↑

A quantizer is said to be optimal (minimal distortion) if the overall distortion is minimized over all K level quantization

⇒ To make such a codebook, the k-dimensional space of vector ~~vector~~ $\tilde{x}$ is partitioned into K regions, i.e.,

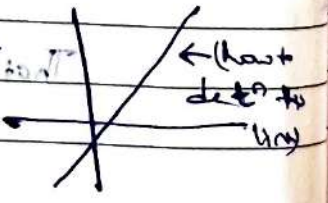
$$\{C_i\} \quad i = 1, \ldots, K$$

with vector $\tilde{y}_i$ being associated with each region C_i

The quantizer then assigns the code vector $\tilde{y}_i$ if $\tilde{x}$ is in C_i that is, $q(\tilde{x}) = \tilde{y}_i$

- Voronoi Diagram / Seeds

→ just diff^n



Conditions necessary for optimality: {of a codevector being assigned}

i) nearest neighbour selection rule

$$q(\tilde{x}) = \tilde{y}_i \quad \text{iff} \quad d(\tilde{x}, \tilde{y}_i) \leq d(\tilde{x}, \tilde{y}_j) \quad \forall\, i \neq j \;\&\; 1 \leq j \leq K$$

(distance)

So quantize $\tilde{x}$ to $\tilde{y}_i$

↓ centre most point if there are no islands in the codebook structure

ii) Centroid Condition

$\tilde{y}_i$ should be chosen to minimise the avg distortion in the region C_i, such a vector is called the centroid of the region C_i

If there are M_i vectors in the region C_i then the centroid will be given by,

$$\tilde{y}_i = \frac{1}{M_i} \sum_{j=1}^{M_i} \tilde{x}_j \quad , \forall\, x_j \in C_i$$

Generalized Lloyd's Algorithm {K-means algorithm}

↑ size of the codebook

Let K be the no. of clusters {i.e. K regions}

that is,

$\{C_i\}$ $i = 1, 2, \ldots K$ as vectors have to be cut in K regions

Step 1: Initialisation

- Set $m = 0$ where m is the iterative index
- Choose a set of initial codevectors, $\{y_i(0), 1 \leq i \leq K\}$ using an adequate method

Step 2: Classification

Let $\{x(n)\}$ $1 \leq n \leq M \leftarrow$ large no

Take each $x(n)$ & put them into cluster $C_i(m)$ using the nearest neighbour selection rule, that is,

$$\tilde{x}(n) \in C_i(m) \text{ iff } d(\tilde{x}, y_i(m)) \leq d(\tilde{x}, y_j(m))$$

$$\forall i \neq j, 1 \leq j \leq K$$

Step 3: Code vector updating

$m \leftarrow m+1$

Update the code vector of every cluster by computing the centroid of each cluster,

$$y_i(m) = \text{centroid}(C_i(m))$$

$$1 \leq i \leq K$$

Compute the distortion measure, $D(m)$

Step 4: Termination

If, $D(m)$ at iteration m relative to $D(m-1)$ is below a certain threshold, stop,

else, go to step 2

classmate
Date 03/10/25
Page

- What should be the starting centroids?
  → random vectors from universe – is this optimal? (No)
  → first K vectors in universe – not optimal as one type of sound
  → 3rd method {missed}
  → Judgment sampling – (?) {we know the speech samples so we choose samples from each type of sound as our initial centroids}

K-Means: Alternative – starting point problem

- Linde Buzo Gray (LBG) Algorithm
  → no decision on starting point {system takes care of it}
  → start with optimal centroids → then expand it by ensuring optimality
  → starting with codebook of size = 1 is centroid of the whole universe
  ⇓
  Split it into 2 centroids → suboptimal {run K means algo}
  ⇓ to make it optimal
  Split it again i.e 4 centroids → {run K means again}

It is also called the binary split algorithm, or modified K-means algorithm.

{Size of code book should be power of 2}

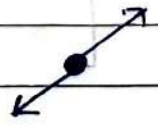
The steps of the algorithm are as follows:

(i) Design a 1 vector codebook. This is the centroid of the entire training vectors

(ii) Double the size of the codebook by splitting each code vector $\tilde{y}_n$ as follows:

$$\tilde{y}_n^+ = \tilde{y}_n(1+\tilde{\epsilon})$$

$$\tilde{y}_n^- = \tilde{y}_n(1-\tilde{\epsilon})$$

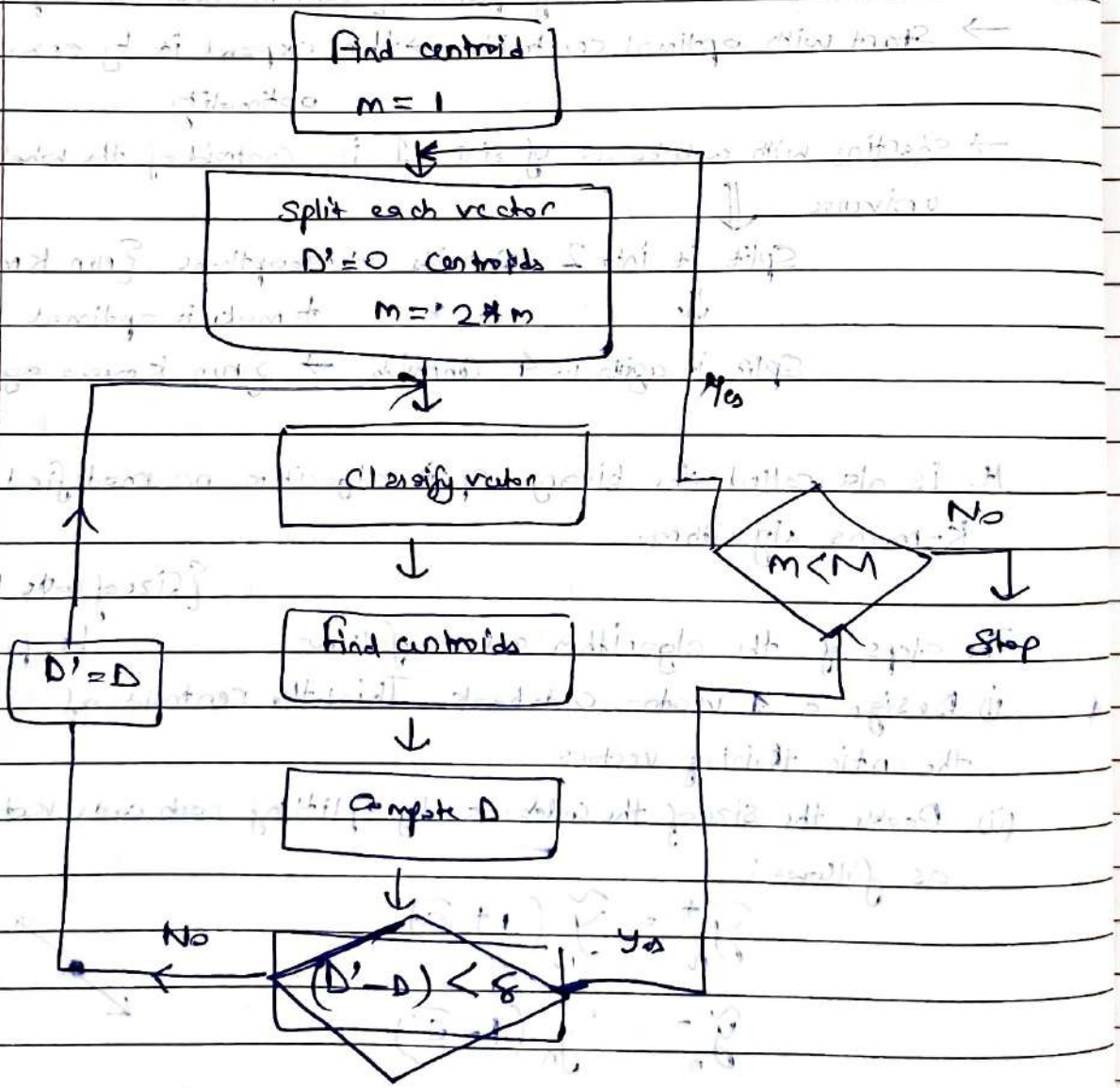


Where n varies from 1 to current size of the codebook
ϵ ϵ is the splitting parameter. It is normally between 0.01 & 0.05

$$0.01 \leq \epsilon \leq 0.05$$

(iii) use K-means algo to get the best set of centroids for the split codebook i.e codebook of twice the size.

(iv) Iterate steps (ii) & (iii) until a codebook of size M (2^0) that is reached.



What is the weak link in this algo?

- The choice of epsilon (ϵ)
  - all points go with 1 centroid & the other is with no points
    - here no new centroid can be computed for the other point
      - as 0/0 → {occurs when ∈ is comparable}

This is called empty cell problem

{can also occur in K-means algo} – but somehow settled

Q How to solve this? → (i) Stop the algorithm for bigger codebook

(ii) throw away the vector which was having no points associated with it, and split the vector with high density & run the ~~algo~~ # K-means again!!

(iii) gather some more data when empty cell encountered {increase the size}

#General Points (i) #12 dimensions

(ii) Since at each split, size of codebook decreases ⇒ Make ϵ logarithmic so that it also dec per split	{randomise which ϵ wrt each dimensions} i.e may be for d_1 → both pos d_2 → pos & neg d_3 → both neg

(iii) $\epsilon \propto$ std deviations of the clusters

{data drives the algo} ⇒ Centroid of data chosen initially

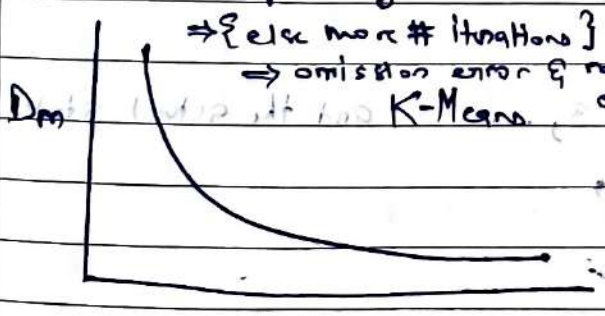
07/10/25

LBG Algorithm

↳ ϵ optimally choose

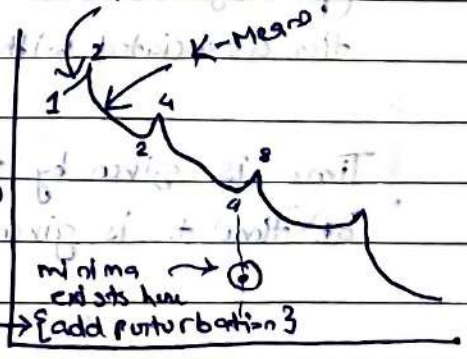
⇒ {else more # iterations}

⇒ omission error & rounding off error



split

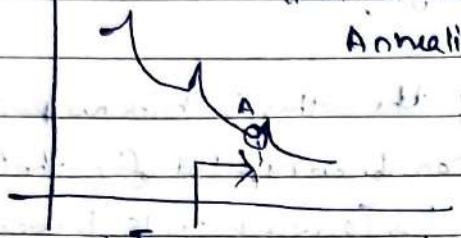
K-Means



LBG

{it helps in improving codebook}

Simulated Annealing



⇒ To get a point over hur, repeat
↑ local optima
Known at A

⇒ by adding perturbation {16th & 17th bit change}

Annealing of Glass
↑ why does it happen?
Physical process - chemical process chuts

→ Make a codebook of size 8 using K-Means LBG

input ↓
⇒ Match C_i with the codebooks
⇓
gives the index value of the codebook entry which matches perfectly!

⇒ per frame, 1 no.!

Statistical Analysis (smtg)
Discrete Markov Process

Consider a system which may be described to be in one of any distinct N states, S_i $(i = 1, 2, \ldots, N)$

At regular intervals the system undergoes a change of state, (possibly to the same state) according to a set of probabilities associated with the state.

Time is given by $t = 1, 2, 3, \ldots$ and the actual state at time t is given by q_t,

$$P(q_t = S_j \mid q_{t-1} = S_i, q_{t-2} = S_k, \dots)$$

$$= P(q_t = S_j \mid q_{t-1} = S_i)$$

$$= a_{ij}$$

1st order markov process

$$1 \leq i, j \leq N$$

Properties:

(i) a_{ij} is a probability, hence $0 \leq a_{ij} \leq 1$

(ii) $\sum_{j=1}^{N} a_{ij} = 1 \mid 1 \leq i \leq N$

Let us consider a weather forecasting problem,

3 states (i) Raining
(ii) Cloudy
(iii) Sunny

Q What's the probability according to the given model that the weather for 8 consecutive days will be Sunny - Sunny - Sunny - rainy - rainy - rainy - Sunny - cloudy - Sunny?

Today is Sunny & the model is,

$$A = \{a_{ij}\} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

Today

Yesterday

This type of matrix is stochastic matrix

$\{\text{row sum} = 1\} \Leftarrow$

classmate
Date
Page

$P(3)$ → Probability it is Sunny today

$= 1$

$P(\underset{\sim}{O}/Model)$

$= P(3,3,3,1,1,3,2,3 / Model)$ (Day 1 … Day 8)

$= P(3) * P(3/3) * P(3/3) * P(1/3) * P(1/1) * P(3/1) * P(2/3) * P(3/2)$

Probability it is 3, yesterday it was 3

This probability given

$= 1 * 0.8 * 0.8 * 0.1 * 0.4 * 0.3 * 1 * 0.1 * 0.2$

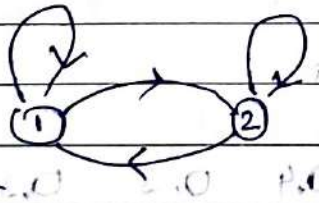
$= 1.536 * 10^{-4}$

08/10/25

- If 1 coin, $P(H) = \frac{H}{H+T}$

$P(T) = \frac{T}{H+T} = 1 - P(H)$

- for 2-Coins, it will need which coins picked first info + probabilities (transition matrix)

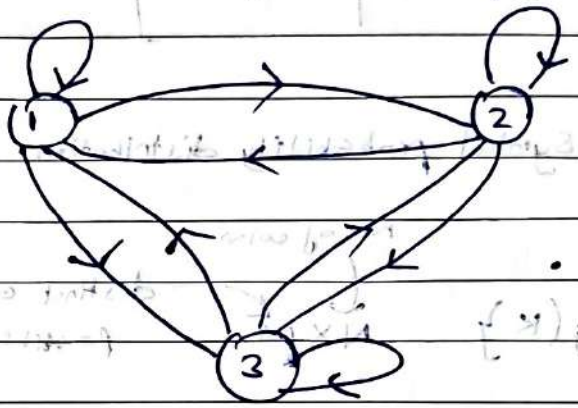


(i) $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ → Transition probability matrix

(ii) Characteristic probabilities $B = \begin{cases} P(H_1) = P & P(H_2) = P \\ P(T_1) = 1 - P(H_1) & P(T_2) = 1 - P(H_2) \end{cases}$

(iii) Initial probabilities (at $t = 1$) $\pi = (\pi_1, \pi_2)$

What if coins increased to 3 ?



• Fully connected graph

→ This can be increased to n no. of coins.

Hidden Markov Models (HMMs)

(remember wordby word)

It is a doubly embedded ~~Shoty~~ Stochastic process with an underlying stochastic process that is not observable, (that is hidden).

It can be observed through another set of stochastic processes that produces the seq of observations.

Game,
i) How many coins } unknown
ii) Nature of coins }
⇒ unObservable

example. {either by ghost game or 2 more that's following}

Elements of an HMM → no. of coins

① N - no. of states, $S = \{S_1, S_2, S_3, \ldots S_N\}$

② At time $t \rightarrow q_t$ (state is) $q_t \in S$

② M - no. of distinct observation symbols per state

i.e codebook size - power of 2 (1024, 512, 256....)

for us: 32

industry: 128

(CCITT) std

If diagonal values high ⇒ inertia of rest!

③ A — The state transition probability distribution

Stochastic matrix → $= [a_{ij}]_{N \times N}$

$= P\left[q_{t+1} = S_j \mid q_t = S_i\right] \quad 1 \le i, j \le N$

④ Observation Symbol probability distribution in state S_j

Stochastic matrix → $B = \{b_j(K)\}$ $N \times M$ (N: no. of coins, M: distinct outcomes possible)

where $b_j(K)$ is the probability of observing K at time t provided $q_t = S_j$

Kind of saying: if C_1 is picked what is the probability of getting H/T

$b_j(K) = P(V_K \text{ at time } t \mid q_t = S_j)$

$1 \le j \le N$

$1 \le K \le M$

- Row sum of A & B are 1 for each row

#2 stochastic matrix

⑤ Initial state distribution

$\Pi = \{\Pi_i\}$ where $\{$probability of state$\}$

$\Pi_i = P(q_1 = S_i) \quad i = 1, 2, 3, \ldots, N$

q_1: state at time point 1

$$\boxed{\sum_{i=1}^{N} \Pi_i = 1}$$

We can write an HMM as,

$\lambda = (A, B, \Pi)$ Where N & M are implicit
A: $N \times N$, B: $N \times M$ {already known}

$$A = [a_{ij}]_{N \times N} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ \vdots & & & \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix}$$

$$B = [b_j(K)]_{N \times M} = \begin{bmatrix} b_1(1) & b_1(2) & b_1(3) & \cdots & b_1(M) \\ \vdots & & & & \\ b_N(1) & b_N(2) & \cdots & & b_N(M) \end{bmatrix}$$

09/10/25

Ball and Urn Model

i) Pick an urn out of N based on prob
ii) Pick a ball from it.
iii) Report the color of ball — {the observation}

Memory
- N — A
- M — B
- NXM — matrix A
- NXM — " B
- N — Π

NOTE: Π — initial prob to pick an urn
A — given current state, what prob to move to next?
B — what observations per state? — how 4 colour balls per urn

Objective is: Determine how many no. of urns present

3 Problems of HMM

Pb1 : Evaluation problem (Scoring Problem)

Given a model λ and a sequence of observations, how to compute that the observation sequence was produced from the model? Also it's called the scoring problem.

{Kind of Compute pb}

How well a given model matches an observation sequence.

$$\max_i P[\underset{\sim}{O} / \lambda_i]$$: probability of observation seq provided model λ_i

Pb 2 : 'Uncovering' Problem

We try to decipher the "hidden" part of the model that is to find the "correct" state sequence.

{As in which State the out-come came from?}

We try to use some optimality criterions to solve this pb as best as possible.

Pb3 : Reestimation Problem

We try to optimize the model parameters to best describe how a given observation sequence comes about.

The observation sequence used to adjust the model parameters is called the training sequence.

classmate
Date 15/10/25
Page

- W = 10

Pb3 will make the model : 96 models

Pb2 will help in evaluating the model - testing : helps in redesigning the model

Pb1 will helps us determining from which model it came!

↳ for deployment : only 1/3rd of the whole system

but in HMM, we used the whole model during deployment/training

↳ bigger size compared to HMM

Solution to Pb1 {Forward Process}

M1 : We want to calculate,

$P(\underset{\sim}{O}/\lambda)$ probability of Observation seq given λ

where $\underset{\sim}{O}$ is the observation seq going from;

$$\underset{\sim}{O} = O_1, O_2, O_3, \ldots O_t, \ldots O_T$$

↳ depends of no. of frames

M = 32

In order to find a solution we enumerate all possible state seq of length T, total no. of state seq = N^T

T — obs

N — state

Let one of the state seq be,

$$\underset{\sim}{q} = (q_1, q_2, \ldots, q_T)$$

↳ for a single observation (O_t)

q_1 → initial state

Now the probability of $\underset{\sim}{O}$ given the above state seq $\underset{\sim}{q}$ can be written as,

$$P(\underset{\sim}{O}/\underset{\sim}{q}, \lambda) = \prod_{t=1}^{T} P(O_t/q_t, \lambda)$$

(here O_i should be independent)

$$P(\underset{\sim}{O}/q,\lambda) = b_{q_1}(O_1) \cdot b_{q_2}(O_2) \cdots \cdots$$
$$b_{q_T}(O_T)$$

Also, the probability of such a state seq $\underset{\sim}{q}$ is given by,

$$P(\underset{\sim}{q}/\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots \cdots$$
$$a_{q_{T-1} q_T}$$

Now the joint probability of $\underset{\sim}{O}$ & $\underset{\sim}{q}$,

By Bayes Theorem $\rightarrow P(\underset{\sim}{O}, \underset{\sim}{q}/\lambda) = P(\underset{\sim}{O}/\underset{\sim}{q},\lambda) \cdot P(\underset{\sim}{q},\lambda)$

$= \pi_{q_1} \cdot b_{q_1}(O_1)\, a_{q_1 q_2} \cdot b_{q_2}(O_2) \cdots \cdots a_{q_{T-1} q_T} b_{q_T}(O_T)$

Finally,

$P(\underset{\sim}{O}/\lambda)$ is obtained by summing the joint probability $P(\underset{\sim}{O},\underset{\sim}{q}/\lambda)$ over all possible state seq $\underset{\sim}{q}$,

$$\Rightarrow P(\underset{\sim}{O}/\lambda) = \sum_{\text{all } \underset{\sim}{q}} P(\underset{\sim}{O}/\underset{\sim}{q},\lambda) \cdot P(\underset{\sim}{q}/\lambda) \cdots *$$

$\prod_{t=1}^{T}$

Time complexity $= 2T \cdot N^T$
$= O(N^T)$

classmate
Date

Now horder to do this operation,

$N = 5$

$T = 100$

It is not feasible for a real time implementation.

M2: The forward Procedure

Consider a forward variable $\alpha_t(i)$ as follows

$$\alpha_t(i) = P[O_1 O_2 O_3 \ldots O_t, q_t = S_i / \lambda]$$

partial forward probability

$i = 1 \ldots N$

$t = 1 \ldots T$

That is the probability of the partial observation sequence, $O_1, O_2 \ldots O_t$ & the state S_i at time t given the model λ.

16/10/25

We can solve for $\alpha_t(i)$ inductively as follows:

(i) initialisation

$$\alpha_1(i) = \pi_i \; b_i(O_1) \qquad i = 1, 2, 3, \ldots, N$$

(ii) induction

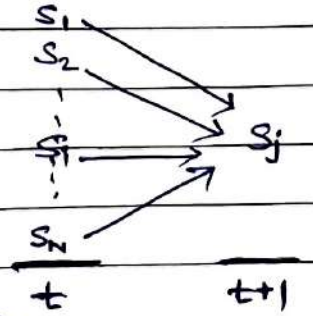
$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) \, a_{ij}\right] b_j(O_{t+1})$$

$1 \le t \le T-1$

$1 \le j \le N$

(iii) termination

$$P(O/\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$



Time Complexity : $O(N^2 T)$ / per model

$\Rightarrow$ for 10 models $\rightarrow 10 N^2 T$

classmate
Date
Page

The Backward Procedure

Consider a backward variable,

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \ldots, O_T / q_t = S_i, \lambda)$$

↑ It has already been observed

→ { Probability of partial observation seq for $O_{t+1}, O_{t+2}, \ldots, O_T$ & at time t, $q_t = S_i$ }

i) Initialization:

$$\beta_T(i) = 1, \quad 1 \leq i \leq N$$

ii) Induction:

$$\beta_t(i) = \left[\sum_{j=1}^{N} a_{ij}\, b_j(O_{t+1}) \right] \beta_{t+1}(j)$$

↖ just for interpretation (in book not there)

classmate

classmate
Date 04/11/25
Page

Soln 2:

$$\gamma_t(i) = P(q_t = S_i / \underset{\sim}{O}, \lambda)$$

That is the pb of being in State S_i at time t given observation seq O & model λ

Using the defn of fwd & bwd probabilities, we may write,

$$\gamma_t(i) = P(q_t = S_i / \underset{\sim}{O}, \lambda]$$

$$= \frac{P(q_t = S_i, \underset{\sim}{O} / \lambda)}{P(\underset{\sim}{O}/\lambda)}$$

$$= \frac{\alpha_t(i)\, \beta_t(i)}{P(\underset{\sim}{O}/\lambda)}$$

$$= \frac{\alpha_t(i)\, \beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\, \beta_t(i)}, \quad t = 1, 2, 3, \ldots, T$$

$\alpha_t(i)$ takes into account the pb of the observation seq $O_1, O_2 \ldots O_t$ & state S_i at time t

$\beta_t(i)$ takes into account observation $O_{t+1} O_{t+2} \ldots$ given state S_i at time t

Hence $\sum_{i=1}^{N} \gamma_t(i) = 1$

Using $\gamma_t(i)$ we can solve for the individually most likely state q_t at time t as follows:

$$q_t = \underset{1 \le i \le N}{\arg\max} \left[\gamma_t(i) \right], \quad 1 \le t \le T$$

($1 \le i \le N$: here transitions are not considered, i.e. $a_{i5} = 0$ but we may get 5th state as very likely)

($\gamma_t(i)$: 2D matrix)

Some a_{ij}'s may be zeroes for some i & j resulting in an optimal state sequence which will be invalid.

One possible sol^n is to modify the optimality criterion for example: Solve for state sequence that maximises the expected no. of correct pairs of states (q_t, q_{t+1}) or triple of states (q_t, q_{t+1}, q_{t+2})

$\beta_t(i)$ takes into account observation $O_{t+1} O_{t+2} \cdots$ given state i at time t

$$\max \sum_{i=1}^{N} \gamma_t(i) = 1$$

We need to maximize Pb of,

$P(\underset{\sim}{q} / \underset{\sim}{O}, \lambda)$ which is equivalent to maximizing Pb of,

observation seq

$$\frac{P(\underset{\sim}{q}, \underset{\sim}{O} / \lambda)}{P(O/\lambda)}$$

A formal technique exists & is based on dp method.

Viterbi Algorithm

To find the single best state seq q,

$$Q = \{q_1, q_2, \ldots, q_T\}$$

for a given observation seq O,

$$O = \{O_1, O_2, \ldots, O_T\}$$

We need to define the quantity,

Scalar quantity $\rightarrow$

$$\delta_t(i) = \max_{q_1, q_2 \ldots q_{t-1}} P[q_1, q_2, \ldots, q_t = S_i, O_1, O_2 \ldots O_t / \lambda]$$

That is $\delta_t(i)$ is the best score (highest Pb) along a single path, at time t which accounts for the

the first t observations & ends in state S_i

By induction we have,

$$\delta_{t+1}(j) = \max_i \left[\delta_t(i)\, a_{ij}\right] b_j(O_{t+1})$$

To actually retrieve the state seq we need to keep track of the argument that maximised, $\delta_{t+1}(j)$ i.e i for each t & j

We do this via the array $\psi_t(j)$ → {keeps track of i which are optimal}

The complete procedure for finding the best state seq can be now stated as,

1. Initialisation: $\left.\begin{aligned} \delta_1(i) &= \pi_i b_i(O_1) \\ \psi_1(i) &= 0 \end{aligned}\right\}, \quad 1 \le i \le N$

2. Recursion: $$\delta_t(j) = \max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij}\right] \cdot b_j(O_t)$$

$$\psi_t(j) = \operatorname{argmax}_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij}\right]$$

$$\left.\begin{aligned} 2 \le t \le T \\ 1 \le j \le N \end{aligned}\right\}$$

3. Termination :

- max probability of the given state seq & observation (i.e of complete path) * : $P^* = \max_{1 \le i \le N} [\delta_T(i)]$

$q_T^* = \operatorname{argmax}_{1 \le i \le N} [\delta_T(i)]$

most optimal state at the last state / time T

4. State sequence (Path) backtracking :

** optimal state seq & it's probability available :

$$\begin{cases} q_t^* = \psi_{t+1}(q_{t+1}^*) \\ t = T-1, T-2, \dots, 2, 1 \end{cases}$$

This is similar to fwd procedure except for the maximisation step.

input :
observation seq
model

output :
globally optimal state seq.

Solution to Problem 3 : The Reestimation Problem

In this problem, we want to determine a set of parameters that is,

$$\lambda = (A, B, \pi)$$

to maximise the probability of occurence of the observation seq given the model

There is no known way of solving this problem explicitly. We can choose a set,

$\lambda = (A, B, \pi)$ such that,

$P(O/\lambda)$ probability of observation seq given λ is locally optimised

We use the Baum - Welch method (or equivalently the Expectation Modification ~~Meth~~ Algorithm or Gradient Techniques (~ Hill Climbing Problem))

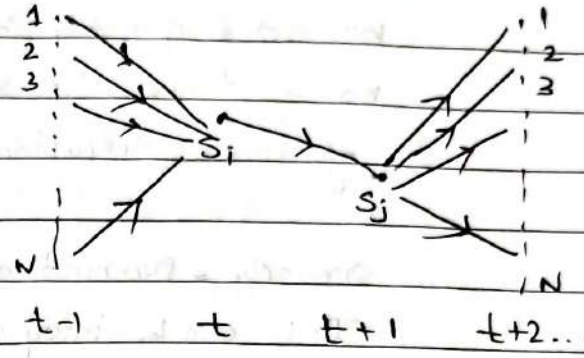
↑ it can be used in any situation : where temporal data is used

The Solution : To solve this problem using iterative update & improvement of HMM parameters we define a variable,

$\xi_t(i,j)$ {the probability of being in state S_i at time t and state S_j at time $t+1$ given the model λ & the observation seq i.e,

$$\xi_t(i,j) = P[q_t = S_i, q_{t+1} = S_j / O, \lambda]$$

Using the defn of fwd & bkd pb, we can write, $\xi_t(i,j)$ as follows,



$$\xi_t(i,j) = \frac{\alpha_t(i)\, a_{ij}\, b_j(O_{t+1})\, \beta_{t+1}(j)}{P(O/\lambda)}$$

$$= \frac{\alpha_t(i)\, a_{ij}\, b_j(O_{t+1})\, \beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i)\, a_{ij}\, b_j(O_{t+1})\, \beta_{t+1}(j)}$$

The denominator makes $\xi_t(i,j)$ into a probability measure.

$$\gamma_t(i) =$$

pb of observation seq ending in state S_i at time t given O & λ

We can now relate $\gamma_t(i)$ & $\xi_t(i,j)$ as follows,

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j)$$

$t = 1, 2, \ldots, T$

(#Paradigm Shift)

If we sum $\gamma_t(i)$ over t, the time index from 1 to T-1,

We get a quantity which can be interpreted as the expected no. of times the state S_i is visited or expected no. of times transitions ~~for~~ are made from S_i

Similarly, summation of $\xi_t(i,j)$ over time t, from 1 to T-1 can be interpreted as,
expected no. of transitions from state S_i to S_j

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{expected no. of transitions from state } S_i$$

~~is the~~

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{expected no. of transitions from state } S_i \text{ to } S_j$$

Using this formulae, we give a reasonable set of reestimation formulae for Π, A, B as follows:

$$\bar{\pi}_i = \text{Expected freq in state } S_i \text{ at time } t=1 = \gamma_1(i), \quad i = 1, 2, \ldots, N$$

$$\bar{a}_{ij} = \frac{\text{expected no. of transitions from state } S_i \text{ to } S_j}{\text{expected no. of transitions from state } S_i} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \quad i,j = 1,2,\ldots,N$$

$\bar{b}_j(k)$ = expected no. of times in state S_j and observing symbol V_k

expected no. of times in state S_j

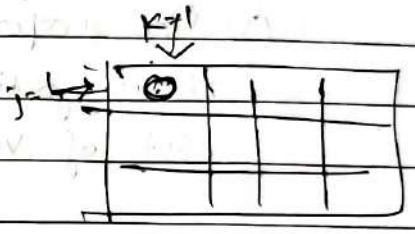
$$= \sum_{t=1 \text{ s.t } O_t = V_k}^{T} \gamma_t(j)$$

(V_k: output of vector quantized)

$K = 1 - - \ldots M$

$j = 1 - - \ldots N$

$$\sum_{t=1}^{T} \gamma_t(j)$$



If we define $\lambda = (A, B, \pi)$ as the current model & compute new models $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ by using the above set of formulae it is proved that, either

$t = 1 \ldots T$ $\quad O_t = V_k$

(i) λ defines a critical point of the likelihood function in which case, $\lambda = \bar{\lambda}$

(ii) $\bar{\lambda}$ is more likely than model λ in the sense that,

$$P(\underset{\sim}{O}/\bar{\lambda}) \geq P(\underset{\sim}{O}/\lambda)$$

What should be starting λ?

(i) Random

(ii) Set it as: $\left[\left[\frac{1}{N}\right]\right]$ for A, $\left[\left[\frac{1}{M}\right]\right]$ for B, $\left[\frac{1}{N}\right]$ for π — No information model (unbiased) — fully connected state machine

NOTE: until $\lambda = \bar{\lambda}$, repeat the process i.e if $P(O/\bar{\lambda}) \geq P(O/\lambda)$ set $\lambda = \bar{\lambda}$ & repeat

The Three problems in HMM – 4 algorithms

Advantages of HMM

(i) It is very accurate

(ii) Time alignment is done automatically & optimally

(iii) It is mathematically tractible & is flexible in modelling real life phenomenon

(iv) Once properly trained, it is statistically stable & performs the recognition task at near ~~of~~ or real time over a large set of varied utterances ($O(N^2T)$ worst case)

(v) It can be used in speaker-independent case

Disadvantages

(i) The no. of states have to be carefully selected as once finalized it cannot be changed for that model

(ii) Training of HMMs require huge amt of training data over a large no. of speakers or utterances

Typically 500 or more.

Digit Recognition System

N = 5

N = 7 – 9 ; for sentence & phrase based detection

(iii) Choice of (starting model) is critical for the proper convergence of the model

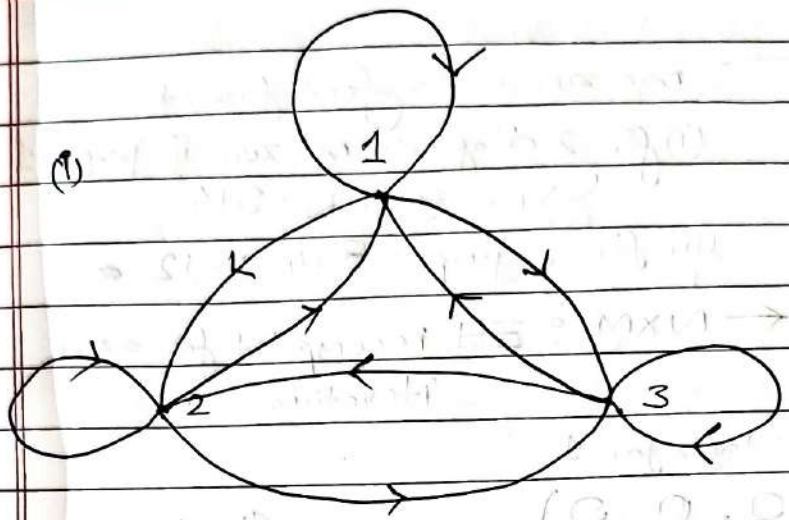
Usually a no. of starting HMMs are used & the model that gives the best performance (given by $P(O/\lambda)$) is chosen

1 2 3 4 5 5 classmates

Date
Page

Types of HMM Model
- Based on state topology

(i)



(Ergodic Model)

Fully connected Model
- Most flexible that can be built

Challenge: converging ⟸ model identification

(for image recognition system)

(ii)

$$A = \left[\left[\tfrac{1}{3}\right]\right]_{3\times 3} \qquad \Pi = \left(\tfrac{1}{3}, \tfrac{1}{3}, \tfrac{1}{3}\right)$$

$$B = \left[\left[\tfrac{1}{4}\right]\right]_{4\times 4}$$

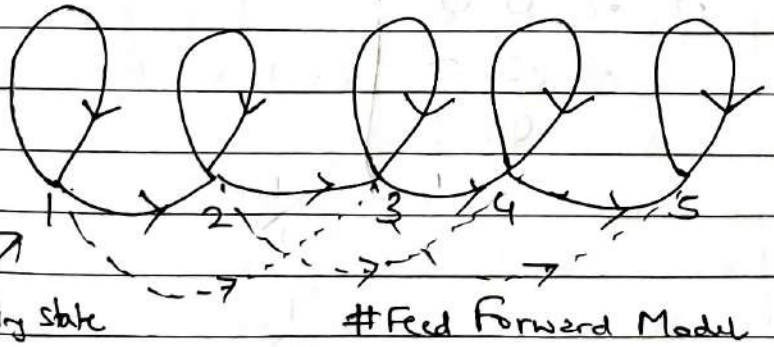
$N = 3$

$M = 4$

(Unbiased Model)

- input: any observation sq which is vector quantization

(iii)



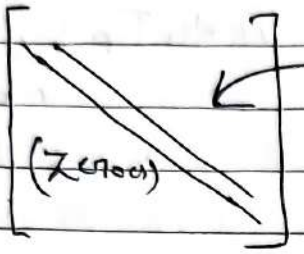
Relevant in speech

Feed Forward Model (Moving forward)

(Bakis Model) - for 1 step moving fwd

NOTE: Data — if 2 steps skipped ⇒ 2 Skipping F.F.M

Both will be in different state due to model constraint



non zeroes as fwd forward

(i) for 2 diagonals non zero if jump of 1
↳ i.e 9 states : 5+4

(ii) for 2 jumps : 5+4+3 = 12

B ← N×M : recomputed for every iterations

Choosing first Model (Bakis Model) as it starts from 1

$\Pi = (1, 0, 0, 0, 0)$

& P(O/λ)

unbiased Model

$$A = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$b_{ij} = \frac{1}{M} \quad \forall i, j$

Adding bias (inertia of rest) (inertia Model)

$$A = \begin{bmatrix} .8 & .2 & 0 & 0 & 0 \\ & .8 & .2 & 0 & 0 \\ & & .8 & .2 & 0 \\ & & & 0.8 & .2 \\ & & & & 1 \end{bmatrix}$$

Architecture of Speech Recognition System

(# Fill in the blanks of source code)!

Soln 1: $\underset{\sim}{O}, \lambda \rightarrow P(\underset{\sim}{O}, \lambda)$

Soln 2: $\underset{\sim}{O}, \lambda \rightarrow p^*, q_t^*, t = 1, \ldots, T$

Soln 3: $\underset{\sim}{O}, \lambda \rightarrow \bar{\lambda} \quad P(\underset{\sim}{O}/\bar{\lambda}) > P(\underset{\sim}{O}, \lambda)$

(# ALL VARS GLOBAL)

Step 1: Make a vocabulary : W words - 10 digits

Step 2: # Training
- First utterance of zero is the training seq

$M = 32, \quad N = 5$

↑ Code book size

Weak law of large numbers
- Req 29 samples atleast to follow normal distribution

*to build the model

⇒ Keep 30 for training and 10 for testing

Step 3: Use trial/starting (λ)
→ use the solution to problem 2
→ print q^* (p^* will be very small)
(1 2 3 4 5 5 ... 5)

Step 4: Compute Soln to problem 1 - α & β

[ ]

$M \times 12$

Step 5: use Soln to pb 3 → get new $\lambda \rightarrow \bar{\lambda}$
go back to Step 3 for ~~checking~~ estimating how good the model is
Check if better (print q^*, p^*)
write old model to a text file; $\lambda = \bar{\lambda}$

Repeat Step 2: 30 models !!! (30 λ for digits)
after we see convergence in Step 3 //

- for 30 models, generate 1 R vs model for O
  - → we get stochastic model – as all stochastic matrices //

use this as starting model & repeat
→ new avg find (run it 3 times)
↓
get the avg → i.e final Model
run 1 more time ←
⇒ pick one of the 30 models which has the highest probability

repeat for 10 digits — (20 models) final!

Test: HEC → run on 20 models → get 20 prob
↓
pick highest & report!!

(observation: P* should increase) # Adjustment of o/p matrix

- After computation of A, check if row sum is 1
  (if not) → add the fraction value which makes it 1 to highest pb
- for B matrix, to get non zero value in every row
  - at every iteration → take it as floor value
    - if val < 10^{-20} → mark it as 10^{-20}
  - non stochastic matrix anymore
    - ⇒ do row sum
    - ⇒ remove the extra value from highest pb!!

- For 30 models, generate 1 avg model for O
  → we get stochastic model – as all stochastic matrices //

use this as starting model & repeat
→ new avg find (run it 3 times)
↓
get the avg → i.e final model M2

run 1 more time ←
⇒ pick one of the 30 models
which has the highest !!
probability

repeat for 10 digits — (20 models) final!

Test : HEC → run on 20 models → get 20 prob
↓
pick highest & report !!

(observation: P* ←] Adjustment of o/p matrix
should increase)

- After computation of A, check if row sum is 1
  (if not) → add the fraction value which makes it 1 to highest pb

- for B matrix, to get non zero value in every row
  at every iteration → take it as floor value
  if val < 10^{-20} → mark it as 10^{-20}
  } not stochastic matrix anymore
  ⇒ do row sum
  ⇒ remove the extra value from highest pb !!

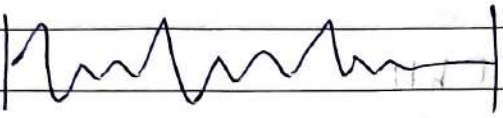
Word Matching

Text Based – DDIF } we can use
TTIF }

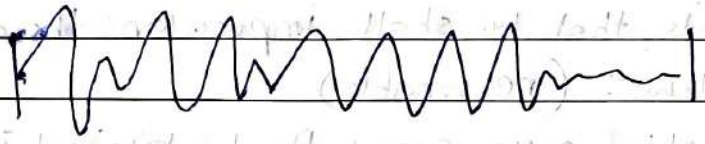
- Speech: not linear

padding can be done to make both same

M2 # Matching 2 unequal speech utterance: Linear Time Warping

ref: 

- Vowel – steady state (Pb?)

test: 

drop frames from test data not ref

If test small → add Silence

M1: Add silence at the end of smaller utterance

(?) Stress points – vowels – match in both & adjust the frames

↑ energy peaks

Dynamic Time Warping (DTW) – 1970's – 80's – Vishnu Atal

It involves two concepts

(i) computation of the features

(ii) distances

Some form of metric has to be used in order to obtain a matched path. There are 2 types of distances:

(a) local distance: it is a computational distance between ~~(b)~~ ~~(c)~~ a feature on one signal with the feature of another signal

↳ Tokhura / Manhattan etc.

(b) Global distance: it is the overall computational distance between an entire signal & another signal of possibly different length

2 algorithms

(i) Symmetrical DTW

example Ref: SPEECH

Test: SSPEEhH

The constraints that we shall impose on the algorithm are as follows: (reasonable)

a) Matching paths cannot go backward in time

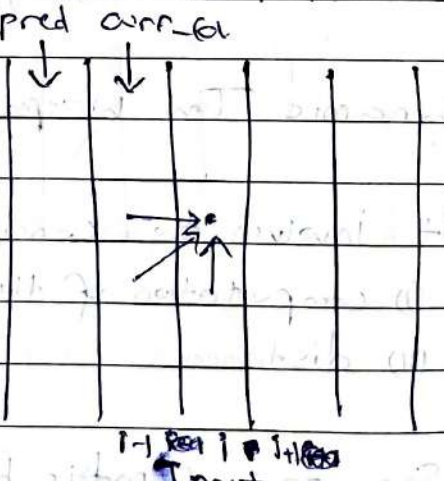
b) every frame in the input must be used in the matching path

c) local distance scores are combined by adding to give a global score

- We will have a matrix, Time - Time Matrix

x-axis : input

y-axis : ref



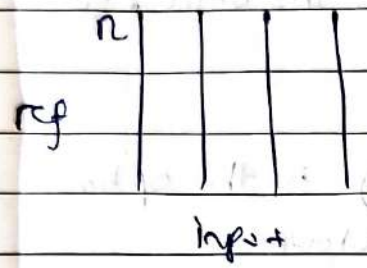
→ Choose from $(i-1, j-1)$ or $(i-1, j)$ or $(i, j-1)$

→ $D(i,j)$ is the global distance upto (i,j) and local distance at (i,j) is given by $d(i,j)$

- no jumps allowed
- No skipping of cells
- movement shown by arrows

$$D(i,j) = \min\left[D(i-1,j-1) , D(i-1,j) , D(i,j-1) \right] + d(i,j) \quad \text{—— } *$$

Distance Matrix ~~(N x N)~~ $(N \times n)$ — input to the algorithm
- compute distance from all point of input to all point of ~~ref~~ ref
- $d(i,j)$



The algorithm to compute the least global cost
- Load only first 2 columns of TT-Matrix
  - ↳ n x 2 points

(i) We need to start at cell (1,1) and go upto D(N,n)

Step 1 (ii) Compute $D(1,1) = d(1,1)$

(iii) populate predecessor column: ~~$D(j,1) = D(j,0) + d(j,1)$ $j \geq 2$~~

$$D(1,j) = D(1,j-1) + d(1,j) \quad j \geq 2$$

now,

for $D(2,j)$: first fill $D(2,1)$

$$D(2,j) = \underset{(*)}{\min}\left(D(1,j-1), D(1,j) \right)$$

Store corr in pred & load the next colin corr repeat 2 onwards

Step (3)

Step (2) populate current column & calculate at (i,j) the local distance = d(ij) + min(global cost at either (i-1, j-1) or (i-1, j) or (i, j-1))

Step (4) Curr column is now assigned to prev col & the steps are repeated. Do this until all columns are populated

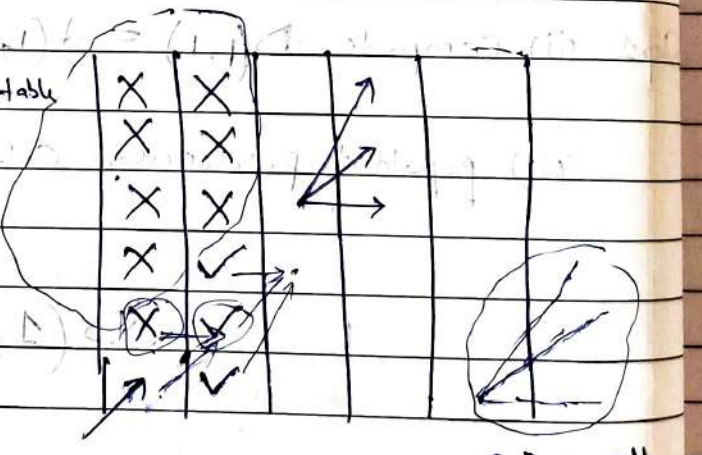
Step (5) The global cost at D(N, n) is the optimal least distance betn the two utterances

(ii) Asymmetrical DTW

There are 3 movements allowed

(we can have many! for sentence recog)

← • a skip 1 jump in ref
• no skip on the input is allowed
} reason of asymmetrical



$$\alpha_{t+1}(j) = \sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\, b_j(o_{t+1})$$

$$\beta_t(j) = \sum_{j=1}^{N} a_{ij}\, b_j(o_{t+1})\, \beta_{t+1}(j)$$

$$\delta_t(i) = \max_{q_1 q_2 \cdots q_t} \left(\delta_{t-1}(j) a_{ji}\right) b_i(o_t)$$

$$\psi_t(i) = \arg\max_j \left(\delta_{t-1}(j) a_{ji}\right)$$

$$P^* = \max_i \delta_T(i)$$

$$q_T^* = \arg\max_i \delta_T(i)$$

$$q_t^* = \psi_{t+1}(q_{t+1}^*)$$

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j)$$

$$\xi = \frac{\alpha_t(i)\, \beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\, \beta_t(i)}$$

$$b_j(k) = \frac{\sum_{t=1}^{T-1} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

$$\xi_t(i,j) = \frac{\alpha_t(i)\, a_{ij}\, b_j(o_{t+1})\, \beta_{t+1}(j)}{\sum\sum}$$

$$\pi_i = \gamma_1(i)$$

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$