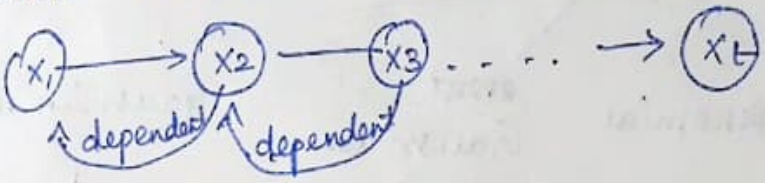
AML

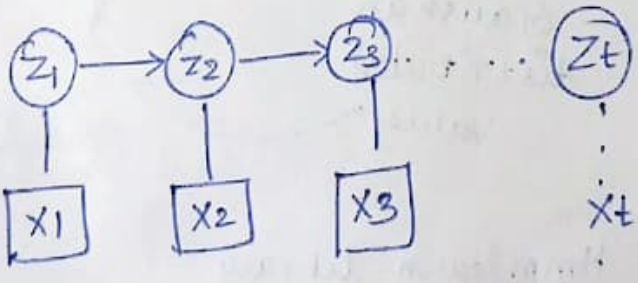
Unit-2 12marks

① Discuss in detail about Building Hidden markov models for sequential data using python.

→ Markov process: Stochastic model describing sequence of possible events in which the probability of each event depends on the previous state.



→ Hidden markov model most important modification to markov process ⟹ the actual system states assumes to be unobservable or hidden.



unable to observe system Z &
can only observe the observable
Process X.

Important parameters

→ Initial hidden state probability

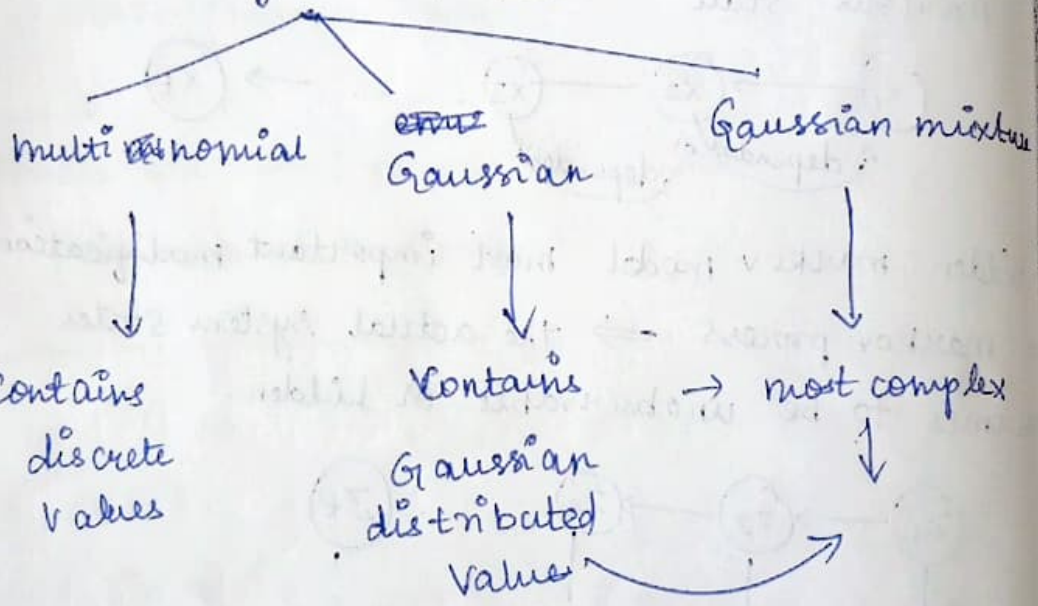
$$\pi = [\pi_0, \pi_1, \pi_2 \ldots].$$

→ Hidden state transition matrix is A.

→ observable probability $\theta = [\theta_0, \theta_1, \theta_2 \ldots]$

→ The hmmlearn library → implements HMM in python.

It gives 3 models



~~Hmm model~~ Hmmlearn library

```
def_init_( self, model_name = "Gaussian HMM"
n_components = 4, cov_type = 'diag')
```

In HMM we use Viterbi algorithm.

↓

max probability to any path at time t.

---

(2) Extract the features of Audio signal using MFCC.

-> MFCC -> widely used extracting audio signals

* Steps:

Pre-processing:

- framing into short intervals.
- Usually overlapped for continuity.

windowing:

each frame is windowed by Hamming/ Hanning.

FFT: Fast fourier transform is applied.

Mel filterbank: linear freq to non-linear freq.

Logarithmic compression: - compress the signals.

Discrete cosine transform: is applied.

normalization: normalised to have zero mean

Feature extraction:

(3) Build a speech recognition model using Hidden markov model.

Step1 - collection & pre-processing of data

↓

MFFC.

Step 2: Define HMM topology

↓

defining structure of the model

Probabilities

Step 3: Training using Baum-welch algo.

↓

used to find the unknown parameters

Step 4: Evaluate.

Step 5: Use the speech recogni

```
import numpy as np
import hmmlearn.hmm as hmm
# Loading dataset.
train_data = np.load('train_data.npy')
# Define HMM
    n_component = 5
model = hmm.GaussianHMM(n_components =
                        n_components)
# Train data
model.fit(train_data)
```

Stock.

test data

test_data = np.load ('test_data.npy)

④ Stock market Prediction using HMM

Step 1 : Data collection : historical stock Prices, financial

→ choosing hidden states depending on the complexity.
since its stock market we need 2 bull / bear . (up & down)

→ stock.npy

⑤ Speech recognition mechanism for infotainment system.

→ . Both h/w & s/w

Step 1:

Suitable mic

↓

Suitable voice recognition Package

↓

Userfriendly UI for user

↓

from mic to s/w (a good software needed)

↓

intent recognition (adjusting volumes)

↓

System response

↓

error handling.

```
import speech_recognition as sr
    r = sr.recogniser()
with sr.mic() as source:
    print("say something")
    audio = r.listen(source)
```

U-3

Explain the building conditional Random fields for sequential text data.

Describe the time series and sequential data series with examp.

Time series:

→ Data collected over time, where obs is recorded at regular intervals.

→ order imp since it must go by time

Step-1 pre processing

Step-2 Choosing model (ARIMA)/(HMM).

Step-3 - Fit the model.

Step 4- Evaluate performance

Eg: Stock market explain

Sequential data:

→ Obs are x.

→ mostly NLP like Speech recog.

→ To build

Step 1: collection & pre-processing

Step 2: Choose modelling approach.

Step 3: Build the model

Step 4: Train & evaluate

Eg: ~~spe~~ ~~audio~~ Speech recognitor

explain

Building CRF for Sequential text data

→ CRF are class of probabilistic model used to make predictions based on sequential data.

→ Usually uses NLP.

steps

1) collection & preprocessing data

splitting into words / numbers.

Step2:

Define the features. they may include current / next / previous words / letters.

Step 3:

Split into training / test data.

Step 4:

Build using python library such as pystruct

Step 5:

Evaluate by accuracy, precision, recall & F1-Score