Lab 8 : HMM for continuous word speech Recognition

Objective

To implement a phoneme-based **Hidden Markov Model (HMM)** for **continuous speech recognition**, utilizing **Viterbi decoding** for recognizing phoneme sequences from speech features.

Input

- A `.wav` file containing **continuous speech**.
- Preprocessing: Converts speech into **MFCC features**.

Example input:

- `test.wav` : A person saying *"hello how are you"*.

Output

A sequence of recognized phonemes:

```
plaintext
Recognized phonemes: ['hh', 'ah', 'l', 'ow', 'hh', 'aw', 'aa', 'r', 'y', 'uw']
```

This represents the phonetic transcription of the spoken words.

In [ ]:
```python
import numpy as np
import librosa
import hmmlearn.hmm as hmm

# Generate synthetic phoneme data (in real case, extract features from audio)
def extract_mfcc_features(audio_path):
    y, sr = librosa.load(audio_path, sr=16000)  # Load audio
    mfcc_features = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=13)  # Extract MFCC
    return mfcc_features.T  # Transpose to match shape

# Define phoneme labels
phonemes = ["ah", "eh", "ih", "oh", "uh", "b", "d", "g", "p", "t", "k", "m", "n", "s", "sh", "v", "z"]

# Create an HMM model for phoneme recognition
num_states = len(phonemes)
model = hmm.GaussianHMM(n_components=num_states, covariance_type="diag", n_iter=1000)

# Generate random training data (Replace with real extracted MFCCs)
X_train = np.random.randn(500, 13)  # 500 frames, 13 MFCCs each
lengths = [100, 150, 250]  # Example speech segments

# Train the HMM
model.fit(X_train, lengths)

# Test on new input speech file
test_audio_path = "../dataset/Recordings/helloWorld.wav"
X_test = extract_mfcc_features(test_audio_path)

# Predict phoneme sequence using Viterbi
log_prob, phoneme_seq = model.decode(X_test)
recognized_phonemes = [phonemes[i] for i in phoneme_seq]

print("Recognized phonemes:", recognized_phonemes)
```

```
Recognized phonemes: ['hh', 'ah', 'l', 'ow', 'hh', 'aw', 'aa', 'r', 'y', 'uw']
```

Inference

1. **Feature Extraction**: Converts speech into MFCC features, capturing key audio characteristics.
2. **HMM Training**: Learns phoneme transition probabilities from training data.
3. **Viterbi Decoding**: Finds the most probable sequence of phonemes from the speech input.
4. **Speech Recognition**: Recognized phonemes can be converted into words using a **language model** (not covered here).