UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Politècnica Superior d'Enginyeria de Manresa

Hidden Markov Models for Speech Recognition

Probablistic Methods — Master's degree in Machine Learning and Cybersecurity for Internet-Connected Systems

Anass Anhari Talib   Eric Muthomi Gitonga   Pol Vidal Miranda

May 29, 2025

1 Introduction

This report presents our work on building a speech recognition system using **Hidden Markov Models (HMMs)**. Due to a widespread blackout in Spain during our academic term, we lost a full week of classes. Unfortunately, this affected our learning continuity, especially on advanced topics such as HMMs which were not fully covered before this assignment was given.

Despite these difficulties, we approached this project with the aim of gaining practical experience on how to update probabilistic models for prediction. We relied on publicly available repositories and research work to implement and present a technical walk-through of HMM-based speech recognition.

2 What is a Hidden Markov Model?

A Hidden Markov Model (HMM) is a statistical model used to represent systems that are stochastic (having a random probability distribution) and time-dependent, where the underlying states are hidden (unobservable) but can be inferred through observable outputs. It consists of:

- **States (S)**: Represent the hidden process.
- **Observations (O)**: Outputs generated by the hidden states.
- **Transition Probabilities (A)**: Probabilities of moving between states over time.
- **Emission Probabilities (B)**: Probabilities of an observation being generated from a state.
- **Initial State Probabilities** (π): Probabilities of starting in each state.
- **State Space (S)**: The set of all possible hidden states.
- **Observation Space (O)**: The set of all possible observations.

The idea is to learn the parameters of this model from labeled audio data. Once trained, given a new sequence of audio features, we can use the model to determine which word (or phoneme sequence) was likely spoken. HMMs are widely used in applications like speech recognition, bioinformatics, and sequence prediction, where temporal or sequential data plays a key role.

3 Step-by-Step Pipeline

The notebook we followed implements an end-to-end system for speech recognition using HMMs. Below, we summarize and expand each step we took.

1. **Audio Input and Visualization**

   We began by reading '.wav' files using the `scipy.io.wavfile` module. This gives us both:

   - The **sampling frequency** (e.g., 16,000 Hz).
   - The raw **audio waveform**, which is just a sequence of numbers representing sound pressure.

   We plotted the audio waveform to visualize how sound varies over time. This helps us understand the data before processing it. The resulting files were stereo audio of shape (stereo: (n_samples, 2)).

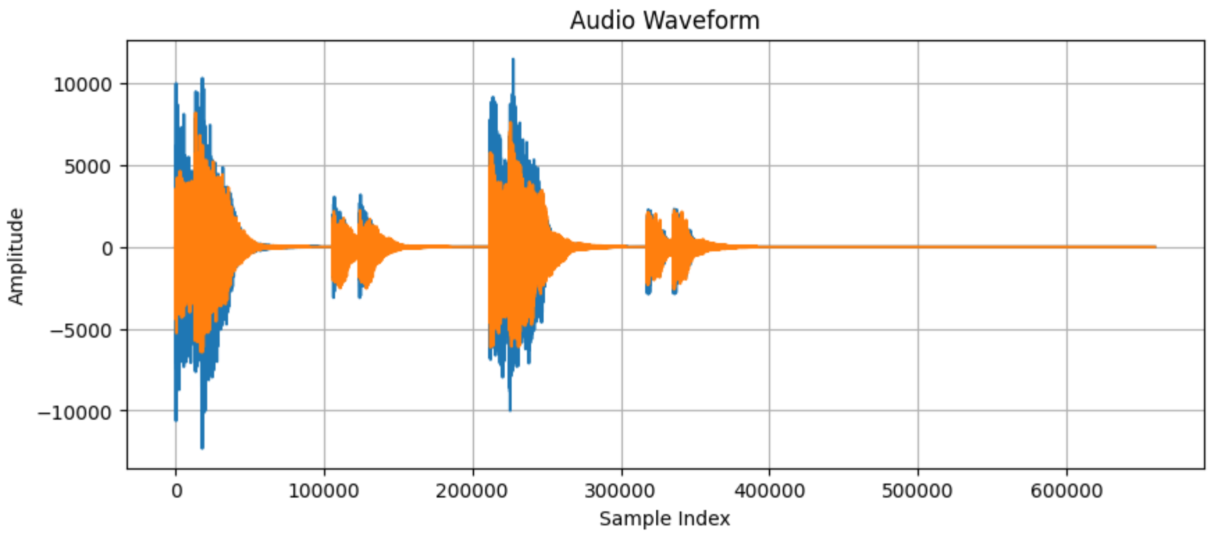

Figure 1: Stereo Audio output

2. **Preprocessing and Normalization**

   - **Normalization**: The audio signal is scaled to a range between '-1' and '1' by dividing by the maximum possible value or the maximum absolute value of the signal.
   - **Root Mean Square (RMS) Scaling**: The RMS of the signal is calculated to represent its energy, and the signal is scaled to a target RMS value (e.g., 0.1) to ensure consistent loudness across all audio samples.
   - **Truncation**: Optionally limits the audio to the first 30 samples, likely for testing or computational efficiency.

3. **Feature Extraction MFCCs**

   Rather than using the raw waveform (which has thousands of values), we extracted features called **MFCCs** (Mel-Frequency Cepstral Coefficients).

   MFCCs are compact numerical representations of the speech signal that:

- Reduce the dimensionality of the data while retaining perceptually important information.
- Are commonly used in speech and audio classification tasks.

The first 13 MFCCs (Mel-Frequency Cepstral Coefficients) are extracted from an audio signal to represent its spectral characteristics.

MFCC Index	**Interpretation**
MFCC 1	Represents the overall energy of the signal (similar to loudness).
MFCC 2	Captures broad spectral shape, related to the formant structure.
MFCC 3	Represents finer details of the first formant (F1).
MFCC 4	Captures information about the second formant (F2).
MFCC 5	Describes additional spectral shape variations.
MFCC 6	Captures higher frequency spectral characteristics.
MFCC 7	Contains mid-range frequency variations.
MFCC 8	Represents slightly higher frequency spectral shape.
MFCC 9	Captures subtle frequency variations important for distinguishing phonemes.
MFCC 10	Represents rapid spectral transitions.
MFCC 11	Captures fine-grained variations in spectral features.
MFCC 12	Sensitive to high-frequency spectral changes.
MFCC 13	Represents the finest details of the speech signal.

Table 1: MFCC Index and Interpretation

Note that MFCC 1 is different from the others because it represents energy rather than spectral shape. MFCCs 2-13 primarily describe the shape of the spectral envelope, which helps in differentiating phonemes in speech recognition and identifying different sounds.

Why Consider Only few Out of 39 Features in MFCC?

MFCCs are commonly used in speech and sound processing. Typically, 39 features are extracted per frame, but many applications reduce this to only 13 15 features.

4. **Data Augmentation**

   To improve model robustness and reduce overfitting, we artificially created more audio data using:

   - **Pitch shifting** (making audio sound higher or lower).
   - **Speed changes** (making speech faster or slower).
   - **Time stretching** (changing duration without altering pitch).

   This helped the model generalize better to different voices and recording conditions.

5. **HMM Training per Word**

   We trained a separate HMM for each word in the dataset. That means:

   - All audio examples for the word yes were used to train one HMM.
   - All examples for no trained a different HMM.

   Each HMM learned the statistical structure (patterns) of the audio features representing its corresponding word.

6. **Classification via Scoring**

   Once all HMMs were trained, we used them to classify new audio samples:

   - For each sample, we calculated the **log-likelihood score** under every HMM.
   - The model that gave the highest score was assumed to be the predicted class.

   To improve this further, these scores were used as features for a simple **logistic regression classifier**, providing a final decision.

4 Results and Evaluation

We evaluated the system using:

- **Confusion matrices**: Shows whether the model's predictions converged or diverged from the actual true labels. The model showed promising results. However, this was not void of some false positives (FP) and false negatives (FN). The model on multiple occasions predicted 'kiwi' as being an 'apple' and vice versa, which showed the limitations of the model.

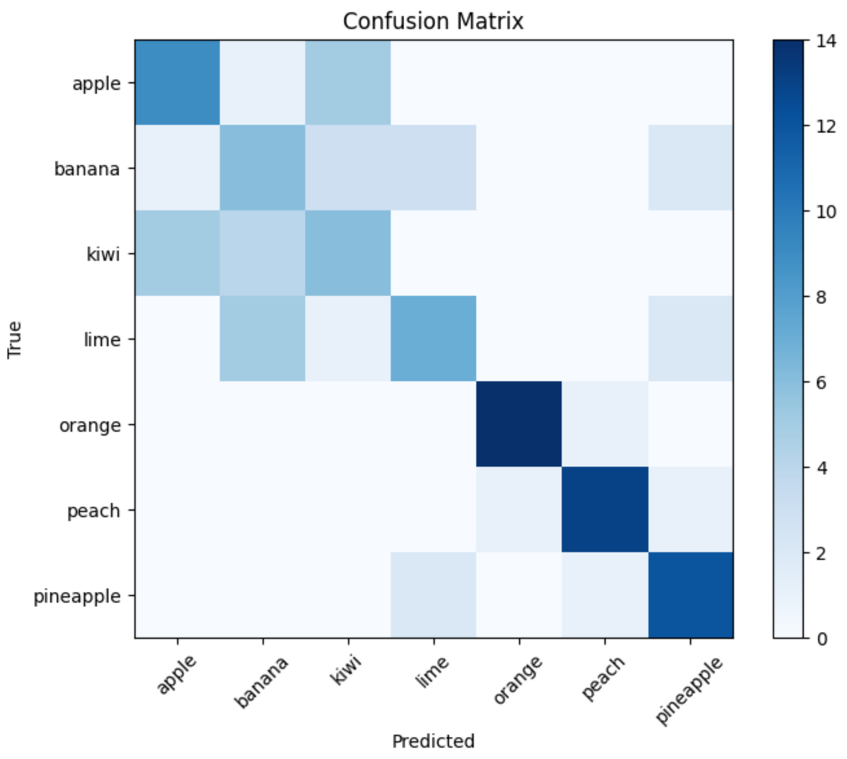

Figure 2: Confusion Matrix for audio files

- **Accuracy scores**: The model got an acceptable accuracy score of 0.64.

Despite the complexity of speech recognition and our short timeline, the model showed encouraging results and worked reliably for basic commands.

5 Challenges of HMM models

HMM often struggles to classify different sounds using a single model due to several key limitations:

1. **HMM Assumes Stationarity and Markovian Properties**
   - HMM assumes that the state transitions depend only on the previous state, which is a **first-order Markov assumption**.
   - However, sound signals often have **long-term dependencies** and require context beyond the previous state.
2. **HMM is Generative, Not Discriminative**
   - HMM is a **generative model**, meaning it estimates the **joint probability** of observations and hidden states.
   - Classification tasks are better suited for **discriminative models** like Deep Neural Networks (DNNs).
3. **Limited Feature Representation**
   - HMM uses simple **Gaussian Mixture Models (GMMs)** or discrete emission probabilities for feature modeling.
   - This representation is often inadequate for complex sound patterns where deep learning models such as CNNs, BiLSTMs can extract **better hierarchical features**.
4. **Noise and Overlapping Sounds**
   - HMMs assume **clean and structured sequences**, making them **prone to misclassification** in real-world noisy environments.

6 Application Scope

- For traditional tasks: HMM, GMM, and Kalman Filters are reliable.
- For modern approaches: RNNs, Transformers, and CNNs offer greater flexibility and accuracy but require more data and computation.
- Emerging trends: Models like WaveNet and Transformers are setting new standards in speech and audio processing.

7 Current state-of-the-art in speech recognition

The field of speech recognition has advanced significantly in recent years, with several key trends and technologies shaping the current state-of-the-art:

- **Deep Learning Architectures**: Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformers have become the backbone of modern speech recognition systems. These architectures can learn complex patterns in audio data, leading to improved accuracy.

- **End-to-End Models**: Systems like DeepSpeech and Wav2Vec 2.0 use end-to-end training, eliminating the need for traditional feature extraction methods like MFCCs. They directly map raw audio to text, simplifying the pipeline.
- **Transfer Learning**: Pre-trained models on large datasets (e.g., LibriSpeech) can be fine-tuned for specific tasks or languages, significantly reducing the amount of labeled data required for training.

For example in this case, we could've used a tensorflow model treating this problem as an image classification problem (https://www.tensorflow.org/tutorials/audio/simple_audio), where the audio is converted to a spectrogram and then classified using a CNN or a Transformer model.

In summary, the current state-of-the-art in speech recognition is characterized by deep learning techniques, end-to-end models, and self-supervised learning approaches that leverage large datasets and advanced architectures to achieve high accuracy and robustness across various languages and domains.

8 Reflections and Conclusion

This project was a valuable learning experience. Working under the pressure of a lost week due to a national power blackout with reduced access to classes, guidance, and time was extremely challenging.

We did not fully understand HMMs at the start. The theory is complex and we lacked time to deeply study the math behind it. However, by following a structured and practical approach, we were able to build a working system and, in the process, gained a meaningful understanding of how HMMs apply real-world audio for speech recognition.

8.1 Things We Learned

- The importance of preprocessing and normalization in audio tasks.
- How MFCCs transform raw sound into meaningful features.
- The intuition of HMMs modeling sequences of data with state transitions.
- That data augmentation is key for performance, even in simple models.

8.2 Final Thoughts

This report does not present a perfect or deeply academic solution, but rather a sincere and resilient attempt by students facing real obstacles. We hope our work reflects a spirit of effort, learning, and adaptation.