Hidden Markov Models

Hoang Le Minh Tien[C4G222]

Eötvös Loránd University, Budapest, Hungary
hoangleminhtien@gmail.com

Abstract. Hidden Markov Models (HMM) was initially introduced in the 1970s and then it become popular because of the rich mathematical base and the model work well and was used for a number of applications. This self-study report is targeting to understand the basic of HMM for probabilistic sequence classification and the application in speech recognition. The feature extraction process is used for extracting information from the acoustic signal. Mel Frequency Cepstral Coefficients is introduced when using HMM for sound recognition task. Then, speech recognition model for isolated word is modeled and analysis.

Keywords: Hidden Markov Model, Speech Recognition, Forward Algorithm, Backward Algorithm, Viterbi Algorithm, Baum-Welch algorithm, Mel Frequency Cepstral Coefficients, Hamming Window, Fast Fourier Transformation.

1 Introduction

Speech recognition is one of the powerful tools for exchanging information by using acoustic signal, which has been researching for many decades. With the development of recurrent neural network and deep learning, it opens a new world for speech recognition tasks. To build the strong base for the audio signal identification, one of the most achievement is obtained and explained by Rabiner et al [1] by using hidden Markov models.

These results are the main supports for effective communication, mostly for people with disabilities. And it can be a subsystem for speech-to-speech translator. During this study, the basic speech recognition system is implemented as a classification task based on labeled training data. The more complex feature extraction, Mel Frequency Cepstral Coefficients, and the application that HMM could be applied is introduced in this report.

2 Background theory

2.1 Motivation

Since most of the instance which can be observed in nature were considered as an iid model. However, in weather forecasting, music composing, language and etc. which showed some dependency between nearby instances. Iid assumption, which mentioned about independences, cannot cover these cases. This brings us with the theory of Markov chain and Hidden Markov models (HMM).

2.2 Markov chain

Markov chain [2] is a formula informs about the probability of sequences of dependent random variables.

Assumed that N is the number of states in discrete time and discrete space, given:

- $S = \{S_1, S_2, \dots, S_N\}$ is a set of states
- $Q = \{q_1, q_2, \dots, q_t\}$ is a sequence of states

The Markov model represent the Markov chain assumption by showing the probability of the sequence:

$$P(q_i = a \mid q_1, q_2, \dots, q_{i-1}) = P(q_i = a \mid q_{i-1}) \quad (1)$$

This can be explained as a sentence: in order to predict the future, it is not depending on the past, only depends on the given present.

To defines a Markov chain, these following requirements are needed:

- A set of states
- A transition probability matrix A
  - Each a_{ij} element of this transition matrix showed the probability of state S_i moving to state S_j, so that:

$$\sum_{j=1}^{n} a_{ij} = 1, \forall i \quad (2)$$

- Initial state probability distribution $\pi = \pi_1, \pi_2, \dots, \pi_N$, where $\pi_i$is a probability starting at state *i*.

$$\sum_{i=1}^{n} \pi_i = 1 \quad (3)$$

A representation of Markov chain can be considered as a following DAG:

$$X_0 \text{ -> } X_1 \text{ -> } X_2 \text{ ->} \dots \text{ -> } X_n \text{ -> } \dots$$

Each node has only one parent, which is the previous observation.

Markov chain can't expect to perfectly observe complete true state of system since there is some hidden information expected under these observations. Hidden Markov models is a solution which hidden/latent variable can be monitored.

2.3 Hidden Markov model (HMM)

HMM is one of the most common models to be used for sequential or temporal data as it is simple enough for us to estimate the parameters and render inferences effective.

Not just this, the layout of HMM is complex enough to accommodate real-world implementation.

The model is described as follow: given a sequence of observations $O = o_1 o_2 \dots o_T$ and the random variables (hidden state) $Q = q_1 q_2 \dots q_N$ which respect the following graph:

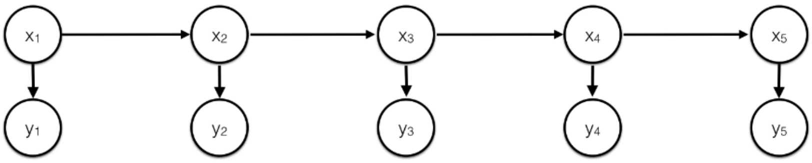


Fig. 1. Trellis diagram or HMM graphical Model where Y are observed variables (which colored gray) and X are hidden/latent variables.

From the graph above and from Markov assumption, we have:

$$p(o_1, \dots, o_n, q_1, \dots, q_n) = p(q_1)p(o_1|q_1) \prod_{k=2}^{n} p(q_k|q_{k-1})p(o_k|q_k) \qquad (4)$$

which is the joint distribution of the Hidden Markov model.

The needed parameters are:

- Transition probabilities: a_{ij} , ANN matrix transition matrix
- Emission matrix: $B = b_i(o_t) = p\ (o_t|q_k = i)$ is the probability of an observation o_t given from state i.
- Initial probability: $\pi = \pi_1, \pi_2, \dots, \pi_N$

2.4 Forward – Backward Algorithm:

By solving this HMM with naïve solution, supposed of having N state in T time, the time complexity of this problem is $O(TN^T)$.

Assume that the emission probabilities, transition matrix and initial distribution are known. By applying the idea of dynamic programing, the target of forward-backward algorithm is to compute the probability of state k given the sequence of observation O.

$$p(q_k|O) \propto p(o_{k+1:n}|q_k, o_{1:k})p(o_k, o_{1:k}) \propto p(o_{k+1:n}|q_k)p(q_k, o_{1:k}) \qquad (5)$$

Forward algorithm:

The target of this state is to compute $p(q_k, o_{1:k}), \forall k = 1, \dots, n$

$$\alpha_k(q_k) = p(q_k, o_{1:k}) = \sum_{q_{k-1}=1}^{m} p(q_k, q_{k-1}, o_{1:k})$$

$$= \sum_{q_{k-1}=1}^{m} p(o_k|q_k, q_{k-1}, o_{1:k-1}) p(q_k|q_{k-1}, o_{1:k-1})\, p(q_{k-1}, o_{1:k-1})$$

$$= \sum_{q_{k-1}=1}^{m} p(o_k|q_k) p(q_k|q_{k-1})\, p(q_{k-1}, o_{1:k-1})$$

$$\alpha_k(q_k) = \textstyle\sum_{q_{k-1}=1}^{m} p(o_k|q_k) p(q_k|q_{k-1}) \alpha_{k-1}(q_{k-1}) \quad (6)$$

where $\alpha_1(q_1) = p(q_1, o_1) = p(q_1)p(o_1|q_1)$

The time complexity $\Theta(nm^2)$

Backward algorithm:

The target of this state is to compute $p(o_{k+1:n}|q_k), \forall k = 1, \dots, n$

$$\beta_k(q_k) = p(o_{k+1:n}|q_k) = \sum_{q_{k+1}=1}^{m} p(o_{k+1:n}, q_{k+1}|q_k)$$

$$= \sum_{q_{k+1}=1}^{m} p(o_{k+2:n}|q_{k+1}, q_k, o_{k+1})\, p(o_{k+1}|q_{k+1}, q_k) p(q_{k+1}|q_k)$$

$$= \sum_{q_{k+1}=1}^{m} p(o_{k+2:n}|q_{k+1})\, p(o_{k+1}|q_{k+1}) p(q_{k+1}|q_k)$$

$$\beta_k(q_k) = \textstyle\sum_{q_{k+1}=1}^{m} \beta_{k+1}(q_{k+1})\, p(o_{k+1}|q_{k+1}) p(q_{k+1}|q_k) \quad (7)$$

for k = 1, …, n-1 where $\beta_n(q_n) = 1\ \forall q_n$

The time complexity $\Theta(nm^2)$

2.5 Viterbi Algorithm

To find a most probable sequence of hidden state Q given the sequence of observed state O, Viterbi algorithm is introduced with the used of forward and backward algorithm

Given observation sequence $O = o_1 o_2 \dots o_T$ and assumed that the emission distribution, transition matrix and the initialization probability is known. The goal is to find the most likely sequence q*

$$q^* = argmax\, p(q|o) = argmax\, p(q, o)$$

$$v_k(q_k) = max_{q_{1:k-1}} p(q_{1:k}, o_{1:k})$$

$$= max_{q_{1:k-1}} p(o_k|q_k) p(q_k|q_{k-1}) p(q_{1:k-1}, o_{1:k-1})$$

$$= max_{q_{k-1}} [p(o_k|q_k) p(q_k|q_{k-1})\, max_{q_{1:k-2}}\, p(q_{1:k-1}, o_{1:k-1})]$$

$$v_k(q_k) = max_{q_{k-1}}[p(o_k|q_k)p(q_k|q_{k-1})v_{k-1}(q_{k-1})\,] \tag{8}$$

where $v_1(q_1) = p(q1, o1) = p(q_1)p(o_1|q_1)$

For each cell $v_t(j)$ of a Viterbi trellis, which represents the likelihood in state j after seeing the first t observations, can be calculated by recursive following the formal definition of Viterbi algorithm [2]:

1. Initialization:

$$v_1(j) = \pi_j b_j(o_1) \quad 1 \le j \le N \tag{9}$$

$$bt_1(j) = 0 \quad 1 \le j \le N \tag{10}$$

2. Recursion:

$$v_1(j) = max_{i=1}^{N}\, v_{t-1}(i)a_{ij}b_j(o_t); \quad 1 \le j \le N \tag{11}$$

$$bt_t(j) = argmax_{i=1}^{N}\, v_{t-1}(i)a_{ij}b_j(o_t) \quad 1 \le j \le N \tag{12}$$

3. Termination:
   The best score: $P* = max_{i=1}^{N}\, v_T(i)$
   The start of back trace: $q_{T*} = argmax_{i=1}^{N}\, v_T(i)$

2.6 Baum-Welch algorithm

Supposed that $\boldsymbol{\lambda}$ is a collection of parameters of a hidden Markov model. Those parameters are *N*, the number of states; *M*, the number of different observations; *A*, transition probability matrix; *B*, emission probability; π, the initialization probability.

3 fundamental problems of HHM [2] are:

1. (Likelihood) Looking for the probability generated by a model given a sequence of observation. Forward-backward algorithm is applied to solve this problem with much faster time complexity comparing with the naïve approach.
2. (Decoding) Looking for the sequence of state that could explain the sequence of observation. Viterbi algorithm is one of the solutions to choose the most likely sequence.
3. (Learning) Given a set of observation $O = \{o_1, o_2, \dots o_T\}$, how could the HMM parameters $\lambda = (A, B, \pi)$ be learned. This problem brings the used of Baum-Welch algorithm.

From forward algorithm, $\alpha_t(i)$, which is the probability given in state *S_i* at time *t* given all the observation that come before $O = \{o_1, o_2, \dots o_t\}$. $\beta_t(i)$, which is the probability in state S_i at time *t* given all the observation that come in the future $O = \{o_{t+1}, o_{t+2}, \dots o_T\}$, could be found from backward algorithm. And $\gamma_t(i)$ is the probability at state S_i at time *t* given all observations.

Assumed that, $\xi_t(i,j)$ is the parameter that is used for capturing the likelihood at time t being at S$_i$ and at time t+1 being at S$_j$.

$$\xi_t(i,j) = p(q_t = S_i, q_{t+1} = S_j | O, \beta_t(i))$$

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_t(i)}{P(O|\lambda)} \tag{13}$$

Then $\xi_t(i,j)$ is related to $\gamma_t(i)$ which showed below:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{14}$$

Then sum $\gamma_t(i)$ over all time *t,* the result is expected at the number of times Si is ever visited. And similarly, doing the same with $\xi_t(i,j)$, the algorithm is proposed:

$$\bar{\pi} = \gamma_1(i) \tag{15}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{16}$$

$$\bar{b}_j(k) = \frac{\sum_{t=1}^{T} \gamma_t(i) \ (observe\ v_k)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{17}$$

The Baum-Welch algorithm is an Expectation-Maximization (EM) algorithm which converges to a local optimum. Then iterations are needed to find the global optimum while this algorithm is being used.

In the E step of the algorithm, $\lambda = (A, B, \pi)$ and O are given to compute $\alpha_t(i)$, $\beta_t(i)$, $\gamma_t(i)$ and $\xi_t(i,j)$. And then in the M step of the algorithm, given those parameters, $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ is recomputed. Iteratively working on this EM algorithm will converge to the local optimum.

When recursion is performed, under-flow can occur for long speech segments, because the probability comes closer to 0 when the sequence is long. Hence, in applications, log-probabilities and log arithmetic are used to avoid this problem [3].

3 Speech Recognition system

The speech recognition system is built by using speech signal convey to words or message. The core of speech recognition system depends on the extracting and modelling method which can distinguish words or phones from the others.

The basic flow chart of the system

3.1 Feature extraction:

This stage of speech recognition is to give a compact representation of the speech waveform as a preprocessing process for the system. The form should be minimizing in term of information loss and separate well words, discriminates between words and by the acoustic model, the good match with distributional assumption should be made [4].

One of the ways to implementing the feature extraction process is using Mel Frequency Cepstral Coefficients (MFCC) to extract speech features for all speech sample [5]. To compute MFCCs, truncated discrete cosine transformation (DCT) is applied to a log spectral estimate, which is computed by smoothing FFT around 20 frequency bins distributed non-linearly in all of the speech spectrum. Then these given features were sent to be trained to create HMM model. Then Viterbi decoding algorithm is used for selecting maximum likelihood word.

Mel Frequency Cepstral Coefficients:

The target of this step is to transfer audio waveform signal to parametric representation, extract unique features for speech samples. Linearly and logarithmically spaced filters are used in MFCC technique. Mel scale, which is the non-linear frequency scale, is used for capturing the phonetically important characteristic of human speech. The unit of the scale is Mel. This scale is a linear mapping bellow 1000Hz and showed as logarithmically scale above 1000Hz.

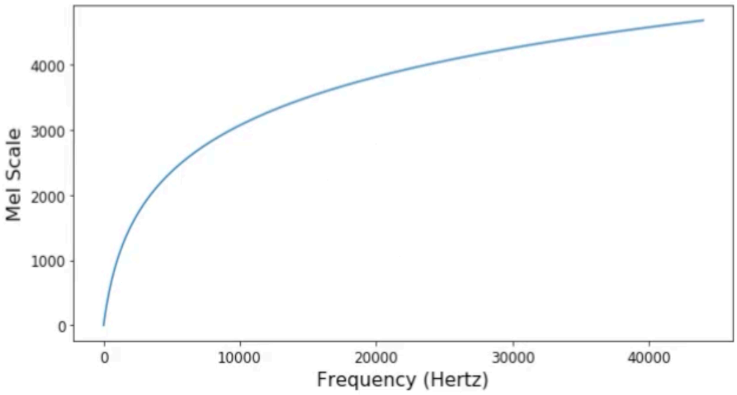


Fig. 2. Mel Scale

Transformation function from normal scale to Mel scale can be seen as followed:

$$Mel = 2595 \log_{10}(1 + \frac{f}{700}) \quad (18)$$

There are 6 computation steps in MFCC [6][7]:

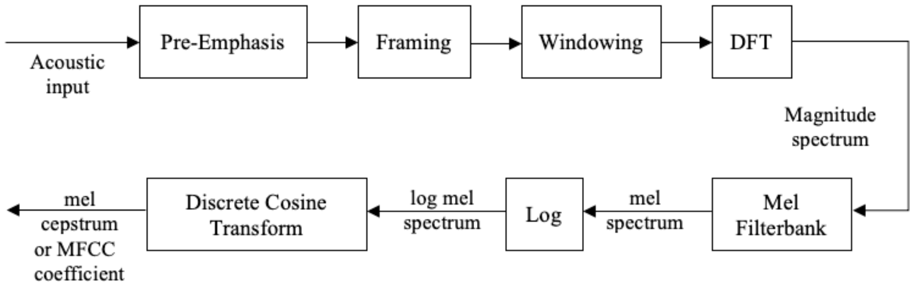


Fig. 3. MFCCs steps

Step 1: Pre-emphasis:

This is one of the first step to reduce noise for the signal. The weaker, higher frequencies signal is boosted before being transmitted or recorded. The energy of high frequency is increased in this process which result to higher the signal-to-noise ratio (SNR).

Step 2: Framing

In order to achieve stationarity, dividing signal into frames is often used in speech processing. This is the process of segmenting the voice samples into N smaller frames. There is an overlapping or adjacent part of the frames, which are denoted by M (where M<N)

Step 3: Hamming windowing

In order to minimize the discontinuities of the signal at the beginning and end of each frame, each frame is windowed. Window function transforms value outside of a chosen interval to be zero, and normally symmetric around the middle inside. In MFCC, Hamming window is proposed and integrates all the closet frequency lines. The reason why using a window function, such as Hamming, instead of using rectangular because the number of sidelobes in this case is less, and it makes the result smoother and more appropriate for frequency-selective study.

Step 4: Fast Fourier Transformation (FFT)

Sound signal is a complex combination of multiple single-frequency sound waves, since sound is recorded, only resultant amplitudes is recorded. Fourier Transformation (FT) is the concept that could decompose a signal into its constituent frequencies and also shows the magnitude of it.

Fast Fourier Transformation calculates the Discrete Fourier Transform (DFT) of the sequence, which instead of taking continuous signal like FT, it considers discrete input signal. DFT and FFT convert time domain to frequency domain.

Step 5: Mel Filter Bank Processing

In FFT spectrum, the frequencies range is large and voice signal scale is not a linear scale. Filter bank on Mel-scale is performed.

In a filter bank, each filter is a triangular which has value of 1 at the center and linearly decreased toward 0 where it meets the other two adjacent filter center frequencies. Each output filter is the sum of its filtered spectral components. Then algorithm is taken to get log Mel spectrum.

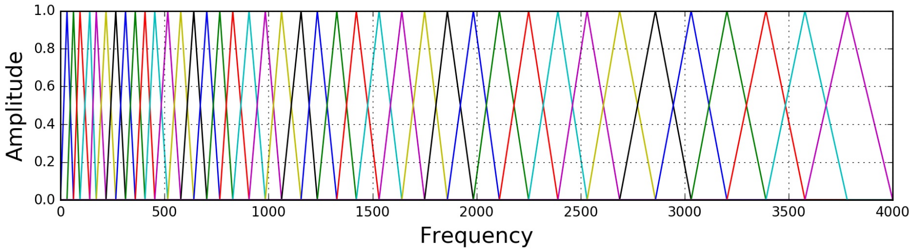


Fig. 4. Filter bank on Mel-Scale

Step 6: Discrete Cosine Transform (DCT)[8]:

Log Mel spectrum is converted to time domain by DCT in this step which output the MFCCs. The set of coefficients is named acoustic vector, which will be linked into a sequence of acoustic vector.

3.2 HHM approach:

After extraction steps, HHM model is train and build with the help of forward and backward or Baum-Welch algorithm. K – mean algorithm which is used for generating the features of voice sample. And Viterbi algorithm is used for decoding, which means choosing the maximum probability sequence in this model.

4 Isolated-word speech recognition [10]

One of the simple word recognition is implemented to learn about the work of HMM. The task is to recognize pronounced word with labeled sound data.

The flow chart of the system is following:

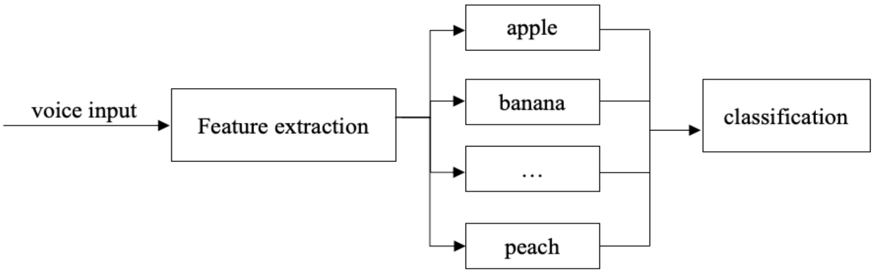


Fig. 5. Flow chart of isolated-word speech recognition system. The feature extracted from voice input will be sent throw each word model to select the best matched.

The dataset is used here is recorded from 1 speaker, 15 utterances for each of 7 words from 1 author is recorded.

4.1 Feature extraction

Base on theory, speech recognition from audio waveform can be done directly. Nevertheless, there is a large variability in acoustic digitalized signal, then feature extraction is preferred to get rid of this variability [9].
The input is sampled at 16 bits, 8000Hz. The feature vector is computed every 10 ms which corresponding to 80 samples, overlap with 20 samples on each side. Hamming window is applied to reduce spectral leakage causing by the framing of a signal before doing FFT.

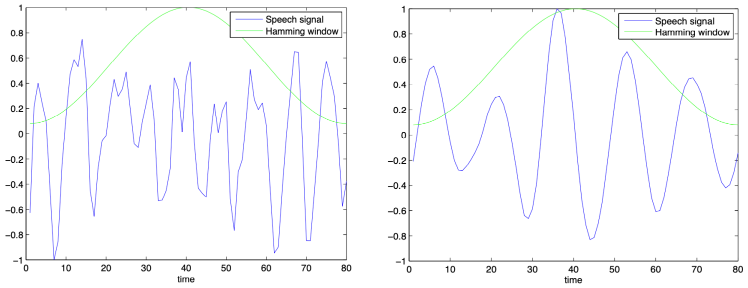


Fig. 6. Speech signal and Hamming window of unvoiced part (on the left) and voiced part (on the right)

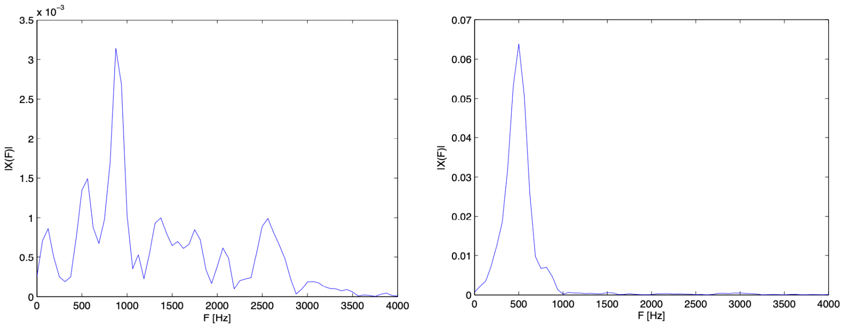


Fig. 7. Single-sided magnitude spectrum of the unvoiced part (on the left) and voiced part (on the right) of a speech signal multiplied by the Hamming window.

Fig.6 showed that the unvoiced part of the speech contains more noise which is at higher frequencies range. This graph is showed an important role of noise reduction in speech recognition task and an important of SNR.

Considering as extra features, the D largest local maxima from the single-sided magnitude spectrum are chosen for each frame. This number is considered as one of the hyperparameter for this model. Because of the simple of this dataset (only one speaker), so the number of local peaks is considered as features. MFCC should be applied for more complex tasks.

4.2 Training

The combination of both unsupervised and supervised techniques was performed. One hidden Markov model for each labeled word was train. The number of different states, which target to represent a phoneme in each word, is chosen.

Baum-Welch algorithm was used to train HMM. The unsupervised task is performing clustering of Gaussians, which depend on the initial values using for Baum-Welch algorithm. 7 separate HMM models, one for each word was trained. The highest output probability, which guess for what word was spoken, is chosen.

Random variables for transition matrix A and initial values π are used in initial step.

4.3 Classification

After the model was trained, the selection is performed as:

$$predicted\ word = argmax\ f(o_1, \dots, o_T; \lambda_i) \tag{19}$$

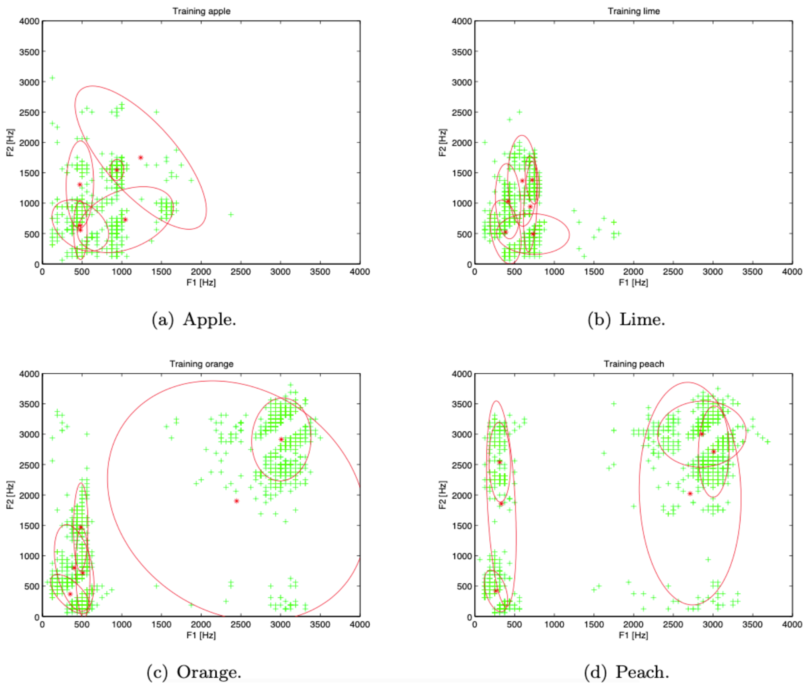
Where $f(o_1, \dots, o_T; \lambda_i)$ is computed by the forward algorithm.

4.4 Results

With the dataset of 105 examples divided to 7 different words, the result of the experiments showed two most important parameters which is the top D local frequencies get from frames, and the amount of hidden states that is used, N. The accuracy of the system is measured by using five-fold cross validation.

N\D	2	3	4	5	6
2			21.9%		8.6%
3	21.0%	15.2%	9.5%	12.4%	1.9%
4	16.2%	11.4%	8.6%	5.7%	3.8%
5	13.3%	8.6%	9.5%	4.8%	2.9%
6	12.4%	10.5%	3.8%	5.7%	7.6%

Table 1. The mislabeled percentage of HMM model given different N and D. The best performance can be found at N = 3, D = 6



(a) Apple.

(b) Lime.

(c) Orange.

(d) Peach.

Fig. 8. The Gaussians distribution model is fitted after 10 iterations of the Baum-Welch algorithm. Six center and its noise range were performed. The green plus present a frame on training data. And the ellipses indicate 75% interval confidence. It is clearly showed that the higher frequency range are performed in those word contain unvoiced word ('dʒ' sound in 'orange' and 'tʃ' sound in 'peach')

4.5 Discussion

Even though the training is performed in the very small dataset with simple extraction phase, the result is good enough for a simple speech recognition system. Since the dataset is recorded from only from one speaker by one person, so this model is over fit for this speech system, collecting more data and recording from multiple sources, and from different gender can solve this problem.

With the small model (N is small), the model does not have enough parameter to separate misclassified. The more complex HMM can be applied to improve the performance of this model. Beside of that, the sampling number in each frame is also important. It is hard for the model to learn meaningful feature when the frame is too small, and there will be lost information when it is large.

5 Outlook and Ideas

From the simple speech recognition discussed in 4th section, HMM is proved to be a robust method having the capability of detecting hidden states from a given observation. More complex HMM models, such as Input-output HMM, Factorial HMM, Coupled HMM, Switching HMM architecture can be implemented.

Since all speech pronouncement are symbolized by concatenating a sequence of phones, then voice modelling is denoted as *beads-on-a-string* model. In real world, context-dependent sequence takes a vast potion, which this *monophones* model fails to handle. *Triphones* model, which take into account every possible pair on the left and on the right, is used to mitigate this problem [4]. Signal preprocessing is an important task having a huge impact to the performance of the model. Reducing the noise an increasing the signal are the main target, which correlated with the height of signal-per-noise ratio. It should be mentioned about the priority of pre-processing and feature extraction. For a former part, high, low frequency filtering or lock-in technique can be applied to filter out unnecessary information for the model. And for the latter, MFCC can be considered to improve the model performance.

HMM is implemented in a lot of application, not only in speech recognition, but also for all of sequence data in general. My proposal is about applying HMM model to the predict the variation of gases monitoring by the photoacoustic system. Continuous concentration measurement of gases is necessary for environmental research and monitoring of industrial processes. Certain applications require short response time measurements. Photoacoustic (PA) spectroscopy-based systems are widely used as gas analyzers and have a typical response time of several seconds, which is mainly limited by the maximum operating flow rate of the PA cells (<1 l/min). High flow rate in a standard PA cell can cause turbulent noise that does not allow the measurements to be made. In order to solve this problems, large volume open PA cell is designed. The major problem of this cell is long response time. HMM can be applied to learn the gas concentration changing sequence in a given variable (like in weather prediction) and leverage the advantages of MFCC, more information could be collected by the PA system, can be used for multiple gases concentration monitoring in different environment condition.

6 Conclusion

This report has reviewed the basic mathematical theory based on probability and showed the advantages of applying dynamic programming theory to reduce the complexity of forward algorithm and backward algorithm. HMMs which is used for sequence classification are showed the relation of observation sequence and hidden or latten state. The Viterbi algorithm is normally used for decoding or inference process, which is to break out the sequence of hidden states given the sequence of observation. Then learning the A transition probability and B emission matrix can be trained with forward-backward or Baum-Welch algorithm.

References

1. L. R. Rabiner, “A tutorial on Hidden Markov models and selected applications in speech recognition,” Proceedings of the IEEE, vol. 77, pp. 257–286, (1989).
2. Daniel Jurafsky and James H. Martin. Speech and Language Processing (2nd Edition). Prentice-Hall, Inc., USA, (2009).
3. N. G. Kingsbury and P. J. W. Rayner, “Digital filtering using logarithmic arithmetic,” Electronics Letters, vol. 7, no. 2, pp. 56–58, (1971).
4. M. Gales and S. Young. The application of Hidden Markov models in speech recognition. Found. Trends Signal Process., 1(3):195–304, (2007).
5. S. B. Davis and P. Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” IEEE Transactions on Acoustics Speech and Signal Processing, vol. 28, no. 4, pp. 357–366, (1980).
6. Rupali, Ms. & Sable, Ganesh. An Overview of Speech Recognition Using HMM. 233-238, (2013).
7. Speech Processing for Machine Learning: https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html, last accessed 2020/05/13.
8. Abushariah, Ahmad & Gunawan, Teddy & Khalifa, Othman & Abushariah, Mohammad. English digits speech recognition system based on Hidden Markov Models, (2010).
9. Shrawankar, Urmila & Thakare, V. M. Techniques for Feature Extraction in Speech Recognition System: A Comparative Study, (2013).
10. HMM Speech Recognition: https://code.google.com/archive/p/hmm-speech-recognition, last accessed 2020/05/13