

Speech Processing - Application of Probability

Sreenya Chitluri
Roll No: 2020102065

sreenya.c@students.iiit.ac.in

Smruti Biswal
Roll No: 2020112011

smruti.biswal@research.iiit.ac.in

Akshit Gureja
Roll No: 2020112004

akshit.gureja@research.iiit.ac.in

Abstract

Speech processing deals with the study of input speech signals and the methods of processing them. Speech processing involves several components in designing of speech systems, a lot of which use probability models. In this report, we will look into use of Gaussian random variables while modelling and filtering noise that gets added after being passed through an additive noise channel in a communication system, and, the applications of Hidden Markov's models in speech recognition.

1. Introduction

Speech Processing is characterized by several components some of which being

1. Speech Coding
2. Speech Recognition
3. Speech Enhancement, etc.

The probability density function (pdf) of speech signals and noise should be well defined for an effective speech processing system.

A major part of speech enhancement is the removal of noise that gets added through an additive noise channel. This is known as filtering and is a huge application of probability. An additive noise is a very important part of the communication systems and speech processing. They are based on Normal distribution and Gaussian Random Variables. As we remove noise, we can recognize speech better. This helps in efficiency in speech processing.

Speech recognition involves enabling the identification and translation of the spoken language into text by computers. Hidden Markov Models are used as basis on modern

speech processing units in general. The purpose of using Hidden Markov Models (HMMs) is because speech signal can be considered as short-time (or) piece-wise stationary signal i.e. it is assumed that statistical characteristics of the signal don't change with time. HMMs are statistical models whose output is a sequence of Random Variables (quantities or symbols). We assume that speech can be approximated to be a stationary process for a very short period of time, say 10 milliseconds. Thus speech is modelled as a Markov model for a stochastic process (or random processes). HMMs are popular for another reason, this is because they can be 'trained' automatically and are simple and computationally viable in practical application. The HMM output sequence of n -dimensional real valued vectors such that n is a small n . Each output is given output in 10 milliseconds. Thus Hidden Markov Models are important in Speech Recognition part of the speech processing.

2. Theoretic Preliminaries

For recognizing the importance of application of Random Variables and Random Processes we must be familiar with the concepts of Random Variables, Probability Density Function, Gaussian Random Variables, Central Limit Theorem, Additive Noise Channel and Stochastic processes.

2.1. Random Variable

A random variable, abbreviated as RV, is a variable whose values depends on the outcomes of a random phenomenon. If X is a RV, then

$$X : S \mapsto \mathbb{R}$$

such that X is a measurable function. This means that it should be satisfying the following property: Pre-images are of the form $(-\infty, x]$, i.e., $X^{-1}((-\infty, x]) \in \mathcal{F}$ where \mathcal{F} is the event space of the random phenomenon.

2.2. Probability Density Function

For a RV X with Cumulative Distribution Function (CDF) defined as $F_X(\cdot)$ such that

$$F_X(x) = P(X \leq x)$$

If $F_X(\cdot)$ is continuous and say it has a differential, this differential is called as **Probability Density Function** (pdf)

$$f_X(x) = \frac{d}{dx} F_X(x)$$

Here, $f_X(x)$ is called as Probability density function. In probability, pdf of a continuous RV is a function whose value at any given sample is the set of possible values taken by the RV can be interpreted as providing a relative likelihood that the value of the random variable would equal that sample. Probability Density Function is used to specify the probability of the random variable falling within a particular range of values, as opposed to taking on any one value. Also,

$$f_X(x) = \frac{d}{dx} F_X(x)$$

$$\implies F_X(a) = P(X \leq a) = \int_{-\infty}^a f_X(x) dx$$

$$P(a < X \leq b) = F_X(b) - F_X(a) = \int_a^b f_X(x) dx$$

If we plot $f_X(x)$ vs x and consider the area between x and $x + \Delta x$. We can also define $P(x \leq X \leq x + \Delta x) \approx \text{Area of rectangle with width } \Delta \text{ and height } f_X(x)$ provided Δ is very small

$$P(a \leq X \leq b) = f_X(x) \Delta x$$

Properties of pdf If a function has the following pdf then it is said to be a valid pdf of the given RV.

1.

$$f_X(x) \geq 0$$

i.e. $f_X(\cdot)$ is monotonically non-decreasing function

2.

$$\int_{-\infty}^{\infty} f_X(x) dx = 1$$

This is because $P(X < \infty)$ hence $F_X(\infty) = 1$

2.3. Gaussian Random Variable

A Gaussian RV is the RV used to describe a Gaussian distribution. The general form of the pdf of a Gaussian RV X is:

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

where μ is the mean, median and the mode of the distribution, and σ^2 is the variance. This RV has a lot of applications in modelling major aspects of signal processing, for eg, noise is represented as a Gaussian RV during speech processing.

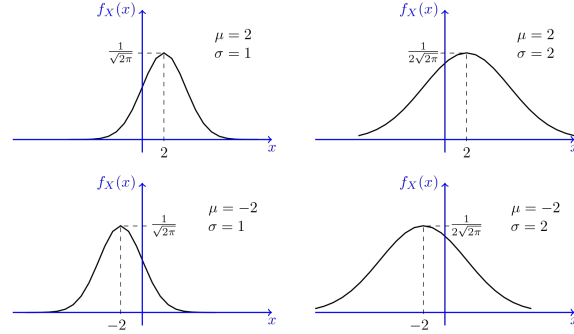


Fig 1: PDFs of example Gaussian RVs

2.4. Central Limit Theorem

The central limit theorem states that when independent random variables are added, their properly normalized sum tends toward a normal (or Gaussian) distribution even if the original variables themselves are not normally distributed. The central limit theorem has several variants. Commonly, the random variables are independent and identically distributed (iid). In other forms, the convergence of the mean to the normal distribution also occurs for non-identical distributions, given they comply with certain conditions.

Theorem: $X_1, X_2, X_3 \dots X_n$ are a sequence of iid RVs where $E(X_i) = \mu$ and $Var(X_i) = \sigma^2$ then

$$Z_n = \frac{\sum_{i=1}^n X_i - E(\sum_{i=1}^n X_i)}{\sqrt{Var(\sum_{i=1}^n X_i)}}$$

This is implied mostly when N is a large number. A simple example of the central limit theorem is rolling many identical, unbiased dice. The distribution of the sum (or average) of the rolled numbers will be well approximated by a normal distribution.

2.5. Additive Noise Channels

Additive Noise Channels are of great importance in communication models. The channel input and the noises are both random processes. *Random process is a mathematical object usually defined as a family of random variables.* Let the input signal be a random process $X(t)$ such that

$$\{X(t); t \in \mathbb{R}\}$$

And let the noise be $N(t)$ such that

$$\{N(t); t \in \mathbb{R}\}$$

. The channel output is another random process $Y(t)$ such that

$$Y(t) = X(t) + N(t)$$

This means that if the time is fixed i.e. for a period of time **Random Variable** Y is equal to *sum of Random Variables* X and N .

$$Y = X + N \text{ given } t \text{ is fixed}$$

We should note that noise in this channel can always be defines at the difference $Y(t) - X(t)$ between the output and the input signal. The notion of additive noise tells us that the processes are independent i.e. *when time is constant the Random Variables formed are Independent in nature*. This nature of the channel incorporates that we know the following about the channel:

1. channel attenuation
2. propagation delay
3. carrier frequency and phase

It is also assumed that all noises are accommodated by the N and the input speech signal is not distorted by any other factors. Additive noise can be easily modelled by a Gaussian Random Variable and is frequently done as so. In other cases, when noise is not directly an Gaussian Random Variable, it is a modified version of Gaussian Random Variable. Most of the noise Random Variables, even though not explicitly mentioned as Gaussian Random Variables, are valid only for Gaussian Random variable and it's pdf.

2.6. Stochastic Process

A stochastic process, in simple terms is a process that deals with probability and values changing with time. A stochastic process involves variables which change at random rate through time. And one of the simplest stochastic processes is a Markov Chain.

3. Applications of Random Variables/Random Processes in Speech Processing

3.1. Speech Enhancement

Speech enhancement aims to improve speech quality by using various methods. After speech passes through additive noise channel, we get disturbances which reduce the quality of speech for processing. So must remove the noise. Noise can be filtered easily if it modelled as a Random Variable (RV).

3.1.1 Modelling Noise as a RV

Noise is often modelled as a Gaussian random variable. This is because noise is a by-product of a large number of

independent small effects which can be represented as random variables. Applying central limit theorem, noise can be mathematically calculated to be a Gaussian random variable. The mean of this Gaussian random variable turns out to be 0.

Let X_1, X_2, X_3, \dots be independent random variables which contribute to noise. Using central limit theorem, we can define

$$Z_n = \frac{\sum_{i=1}^n X_i - E(\sum_{i=1}^n X_i)}{\sqrt{Var(\sum_{i=1}^n X_i)}}$$

and say that the sequence Z_1, Z_2, \dots converges to Z which is a Gaussian random variable with mean 0 and variance 1. Using this result, we can define the pdf of Z as

$$f_Z(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$$

using pdf of Gaussian RV as defined before. Here, Z is the noise that gets added to the speech signal after passing through an additive noise channel. As part of speech processing, we must filter out this noise. This is known as speech enhancement.

3.1.2 Filtering of Noise

Noise when modelled as a Gaussian RV has mean equal to 0 as described above. This is a crucial factor for the filtering of noise and getting better quality of speech signals. Let's say our input speech signal is a sinusoidal wave as described in Fig 1a. As it passes through an additive noise channel, noise gets added as shown in Fig 1b.

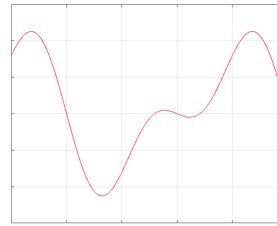


Fig 2a

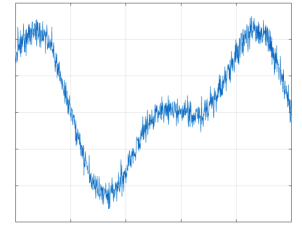


Fig 2b

The filter works by calculating the average of all the values of the signal from a to $a + M$ and mapping it to the $(a + M + 1)^{\text{th}}$ point, where a is any point in the signal domain and M is a constant. We know that

$$n = s + g$$

where n, s, g are the noisy output signal, input speech signal and noise modelled as a Gaussian RV respectively. The filter takes input n and gives an output signal which is expected to be the closest to s . Let this output signal be o . From the above analysis, we know that

$$o_{a+M+1} = \frac{\sum_{i=a}^{a+M} n_i}{M}$$

Replacing n with $s + g$,

$$o_{a+M+1} = \frac{\sum_{i=a}^{a+M} (s_i + g_i)}{M} = \frac{\sum_{i=a}^{a+M} (s_i) + \sum_{i=a}^{a+M} (g_i)}{M}$$

As the value of M increases, the term $\frac{\sum_{i=a}^{a+M} (g_i)}{M}$ tends to $E(g)$. As we modelled g i.e. noise as a Gaussian RV with mean 0 and variance 1, $E(g) \rightarrow 0$ and $o_{a+M+1} \rightarrow \frac{\sum_{i=a}^{a+M} (s_i)}{M}$. We observe that as M increases, o becomes closer to s but a shifted version of it as shown in Fig 2. This is known as an M -order filter.

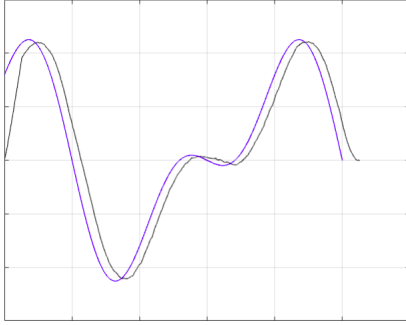


Fig 3: Filtered and original signals

So, by modelling noise as a Gaussian RV we can eliminate noises by using a simple filter as described above. And henceforth we improve the speech signal quality, hence enhancing it. This effective filter is called M -order filter.

3.2. Speech Recognition

Now let's look at one more application of probability in the field of speech processing, which are Markov Chains and Hidden Markov Models.

3.2.1 Markov Chains

A Markov Chain is a memory-less stochastic process, which means that the probability of future events is independent of any previous event till the present. This property of memory-less process is known as the **Markov Property**.

To explain in more clarity how Markov Chains differ from general Stochastic Processes, it would be good if we look at an example.

Suppose we have a bag of 5 Blue Balls, 3 Red Balls and 5 Green Balls, and we have to find the probability that the 3^{rd} ball we pick without replacement is Red in color.

Note that here, the probability that we are looking for depends on the number of Red balls left in the bag when we go for our 3^{rd} pick which is dependent on the balls

that have been picked up in the first 2 picks. Thus, the probability that we aim to find depends on previous events and this is a general stochastic process.

Let's twitch the example a bit and now we are picking balls with replacements. So after each step, the number of balls in each color is independent of any picks we do at any step. This is analogous to a memory-less system in the sense that an future event is independent of any past probability or trial. This forms an example of a Markov Chain.

3.2.2 Directed Graph Representation

One of the most common Markov chains are time-homogenous Markov Chains which are represented using directed graphs, like the one given below. In such cases, the probability of any state is independent of time.

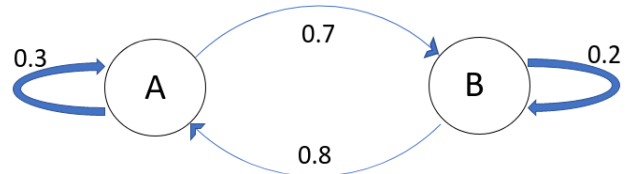


Fig 4

In figure 4, each directed arrow is labelled which denotes the probability of going from one state to the one pointed to. So the probability of going from A to B is 0.7, and staying at A is 0.3. One important point to visualize is that the sum of the probabilities of all the arrows going out from a state must be 1 for a valid process.

Suppose we are on 'B' as the starting point, and we need to move to A in 2 steps.

This can be done, if in the first step we stay at B and move to A in 2nd step, the probability of which is

$$= 0.2 * 0.8 = 0.16$$

Another way is we move to A in the first step and stay at A during the 2nd, the probability corresponding to this is

$$= 0.8 * 0.3 = 0.24$$

So, the total probability becomes

$$0.16 + 0.24 = 0.4$$

A Markov Chain can then be assumed to be a sequence of random variables satisfying the rules of conditional independence. Thus, Markov Property can be stated formally as:-

If X_1, X_2, \dots, X_n are 'n' random variables and a_1, a_2, \dots, a_n depict the possible states. Then,

$$P(X_n = a_n | X_{n-1} = a_{n-1}) = P(X_n = a_n | X_1 = a_1, X_2 = a_2, X_3 = a_3, \dots, X_{n-1} = a_{n-1})$$

In other words, knowledge of the previous states is all that is necessary to determine the probability distribution of the current state.

Note: This doesn't contradict our memory-less property, the probability of transitioning from one state to another is independent of any previous states we have been through. If we wish to get the probability distribution, we just need the knowledge of where we have been for analysis.

3.2.3 Transition Matrix Representation

A transition matrix (P_t) for a Markov Chain at any time 't' holds all the probabilities of transition between all possible states. The $(i, j)^{th}$ element of this matrix P_t is given by:

$$(P_t)_{(i,j)} = P(X_{t+1} = j, X_t = i)$$

Thus each element of this matrix holds the conditional probability of moving from state 'i' which is the current position/ state of a random variable X at time 't' to state 'j' in the next step. So, corresponding to our example in figure 4, the transition matrix can be given as -

$$\begin{bmatrix} 0.3 & 0.7 \\ 0.8 & 0.2 \end{bmatrix}$$

3.2.4 Hidden Markov Models

Hidden Markov Model is a model which assumes that when a data of a system is being observed, what is being observed is not the state the system is in, but the data generated by some underlying hidden states.

For example, suppose we are listening to sound coming from a speaker. The sound which we listen to and is entering our ears is different from the state of the the speaker, which is the sound generating system.

In many real life applications, it is extremely difficult to predict the state of a system by just looking at the observations at hand. Hence, we use functions to map states to observations. Thus,

$$f(s) = o$$

Where 's' is used to denote the state of a system, which is mapped by the function 'f(x)' to an observation 'o'. Thus, 'o' is the observation we make when function f(x) is applied to the state 's' of a system.

It is perfectly possible to have 2 different states s_1, s_2 being mapped to the same observation, in which case it

forms a many-one function. Generally, in such cases, it is impossible to invert perfectly, because $f^{-1}(o)$ can be s_1 or s_2 .

A Hidden Markov Model, is similar as when given an observation it can estimate the probability distribution over the possible states at any time, because many state sequences could lead to the same observation sequence.

In the field of Speech processing, the whole application of Hidden Markov Model in speech recognition revolves around the problem is estimating the system state (word spoken) behind the observations (sounds).

3.2.5 Rules of Markov Models

1. Markov Property :

The first rule is that the model must satisfy the Markov Property, according to which the model must be memory-less and the probability to transition from the current state to next depends only on the current state and independent of any past event or state the system as been through.

$$P(s_t | s_{t-1}) = P(s_t | s_1, s_2, s_3, \dots, s_{t-1})$$

2. Emit Observation After Each Transition

This rules for Markov Models states that after each transition from one state to another, the system must emit an observation dependent only on the current state. That is

$$P(o_t | s_t) = P(o_t | s_t, o_{t-1}, s_{t-1}, \dots, o_1, s_1)$$

Now, the observer gets to know only the observations and is unaware of the states and transitions that generated them; thus our hidden to the observer, giving the model its name – Hidden Markov Model(HMM).

An HMM is fully characterized by 5 parameters:

1. Set of N states

$$Q = s_1, s_2, s_3, \dots, s_N$$

2. Set of T observations

$$O = o_1, o_2, o_3, \dots, o_T$$

3. Probabilities of State Transition

probability of transition from s_i to s_j denoted by $a_{i,j}$ and stored in the transition probability matrix A.

4. Conditional Observation Probabilities (Emission probabilities)

Probability that a model or system will emit an observation o_t when in state s_i . That is $P(o_t|s_i)$ and is denoted by b_i . We take B as the set of all these emission probabilities.

5. Probability of State Initialization:

The probability that the states of an HMM begin from the state s_i and is denoted by π_i .

We are now fairly equipped to study about the applications of Markov Chains and Hidden Markov Models in Speech Recognition.

3.2.6 Hidden Markov Models in Speech Recognition

To move to Speech Recognition, we need to first understand phonemes.

What are Phonemes? A phoneme is a unit of sound as small as 20 to 4 milliseconds, so it's super super short. It is this unit which helps distinguish one word from another in a particular language. It is the tiniest part of the word, changing them changes the meaning of the word. Words like "thumb" and "dumb" are 2 different words and can be distinguished by just substituting one phoneme 'th' in thumb to 'd'.

Where do HMM's come here? Phoneme are put together in the correct sequence by using a Hidden Markov Model which does this task by applying statistical probabilities. So we check how probable it is that one phoneme follows the first. This can be put in 3 basic ideas or tasks.

1. Represent speech as a sequence of observations that a system has made.
2. Use HMM to model some unit of that speech, a word for example.
3. Concatenate a single unit into larger units, phonemes to words and words to sentences for example.

Lawrence Rabiner in 1989, characterized these as the 3 fundamental problems.

- Problem 1 (Likelihood):
Given an HMM λ and an observation Sequence O , find the likelihood $P(O|\lambda)$.
- Problem 2 (Decoding) :
Given Observation Sequence O and an HMM, estimate the best hidden state sequence Q , which maximizes probability of observation sequence
- Problem 3 (Training) :
Adjust parameters to maximize the probability of observed sequences.

We will look at the solutions to all these problems one by one.

3.2.7 Likelihood Problem

Often referred to as the Evaluation problem, we have the task to find $P(O|\lambda)$ when given the observation sequence $O = o_1, o_2, \dots, o_T$. Let's begin with a general approach to the task at hand. We have an acoustic observation sequence $O = o_1, o_2, \dots, o_T$ and our goal is to estimate the word sequence spoken $W = w_1, w_2, \dots, w_M$. Let the estimated word be represented by \hat{W} .

$$\hat{W} = \underset{W}{\operatorname{argmax}} \frac{P(O|W)P(W)}{P(O)}$$

$$\hat{W} = \underset{W}{\operatorname{argmax}} P(O|W)P(W)$$

However, this leads to give an exponential time complexity of $O(N^T)$, N being the possible states which can be reached. This is not favourable to be used in real life as it makes the computations slow, and increasingly time-consuming.

For which we use algorithms based on Dynamic Programming called as Forward Procedure. This algorithm uses a variable $\alpha_t(i)$ representing the probability to be in state i after seeing the $t - \text{length}$ observation sequence o_1, o_2, \dots, o_t . Each probability $\alpha_t(i)$ is computed by summing the probabilities over paths that lead to the current state, and this lends to the algorithm its recursive nature which greatly reduces the time complexity. Thus the whole algorithm can be summarized in 3 steps:

$$\text{Initialization} = \alpha_t(i) = \pi_i \cdot b_i(o_i) \quad i \in [1, N]$$

$$\text{Recursion} = \alpha_{t+1}(j) = \sum_{i=1}^N \alpha_t(i) a_{i,j} b_j(o_{t+1}) \quad t \in [1, T-1], j \in [1, N]$$

$$\text{Termination} = P(O|\lambda) = \sum_{j=1}^N \alpha_T(j)$$

3.2.8 Decoding Problem

The algorithm we use for decoding is the Viterbi algorithm. Using this algorithm we consider only the optimal decoding sequence corresponding to maximum likelihood and not all potential paths. This algorithm is also based on dynamic programming like the forward procedure used in the previous step.

This algorithm uses a variable $v_t(i)$ which represents the probability of being in state i after looking at the t observations and going through most probably state sequence

s_1, s_2, \dots, s_{t-1} . Thus,

$$v_t(i) = \max_{s_1, s_2, \dots, s_{t-1}} (P(s_1, o_1, s_2, o_2, \dots, s_{t-1}, o_{t-1}, o_t))$$

Similar to the forward procedure, here we compute the Viterbi probability by getting the highest probability states which lead to the current state of the system. This lends to the decoding it's recursive nature and we can say-

$$v_t(i) = \max_{i=1 \text{ to } N} (v_{t-1}(i) a_{i,j} b_j(o_t))$$

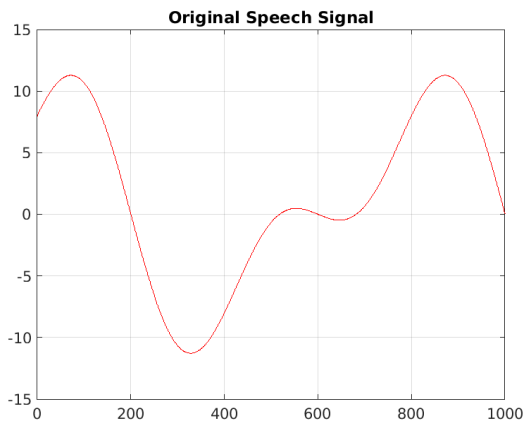
3.2.9 Training and Learning Problem

This is the toughest part of the process which is done using the Baum-Welch Algorithm. This algorithm helps the system learn about the transition probabilities and the probabilities of emission of the observation. It gives the model the flexibility to adjust the parameters to get the best Hidden Markov Model for a given sequence.

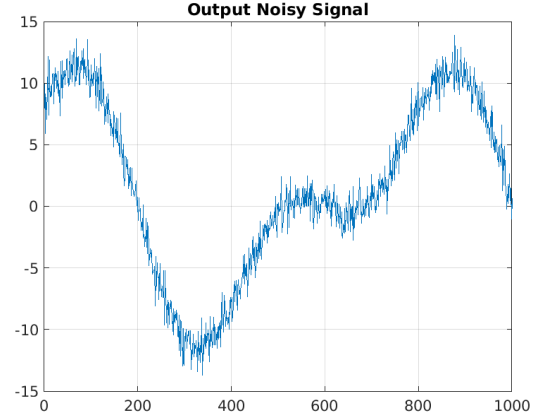
The algorithm works on the idea that if we know the probability distribution of the states at the time 't', we can get the transition and emission probability. However, more detail into the algorithm goes beyond the scope of our report.

This sums up the application of Markov Models in the speech recognition systems. In most of the modern day state-of-the-art systems used in speech and audio recognition a hybrid of HMM's and Neural Networks is used to deliver the best performance.

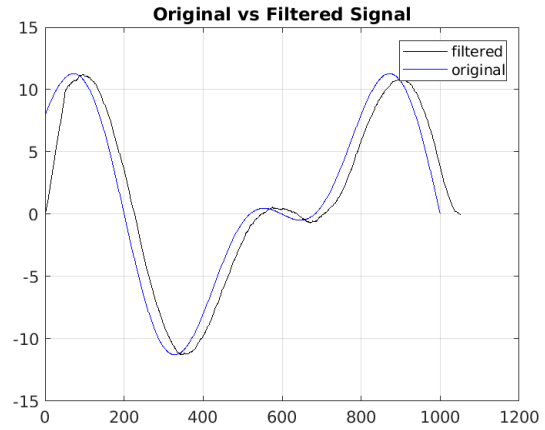
4. Results and Discussion



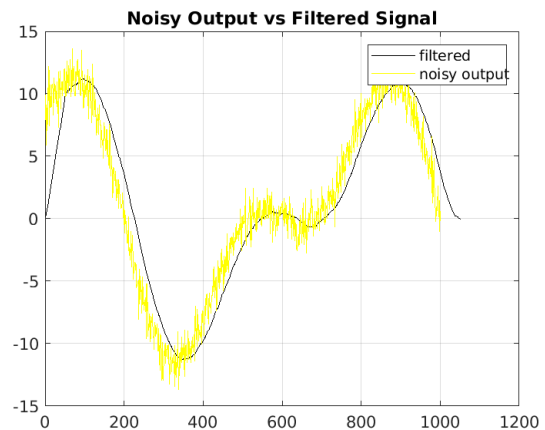
This is the original speech signal before it passes through any noisy channel.



After being passed through an additive noise channel, the signal appears to look like this. The noise added is normally distributed and can be modelled as a Gaussian RV with mean 0 and variance 1.



Taking into advantage that the mean of the noise is 0, the output of the M -order filter with $M = 51$ resembles the original signal but it is shifted right.



This is the output of the M -order filter vs the output of the additive noise channel.

```
Command Window

Hi
  "Hi"    "I"

  "Hi"    "I"    "am"

  "Hi"    "I"    "am"    "a"

  "Hi"    "I"    "am"    "a"    "dog"

Final order of words/Most likely sentence given detected words:
  "Hi"    "I"    "am"    "a"    "dog"

>>
```

Given the detected set of words, using Markov's model of speech recognition, we can predict the center.

5. Conclusion

Therefore, we have identified many crucial applications of probability in speech processing. Initially we started with Speech Enhancement which included filtering of signal as it comes out of additive noise channel. We applied the concepts of Gaussian Random Variables, Central Limit Theorem and Random Processes at constant time. We related the input, noise and output of a additive noise channel as Random Processes, then when we view them under constant time they can be considered as Random Variables. Noise, as we know, is a result of several independent Random Processes, and hence Central Limit Theorem is used to model Noise as a Gaussian RV with mean μ tending to 0. Taking this fact as an advantage, we design a filter which produces the outputs by taking the average value of certain points such that the Gaussian Noise is removed. This filter is simple yet efficient. This is called an M -order filter.

We have also been introduced to the concepts of Markov Chains, Hidden Markov Models and their applications in the field of speech processing and speech recognition. We started from the basics of Markov Chains and built ideas on top of them. We applied our knowledge of Markov Chains and Stochastic Processes to get more complex Models like the Hidden Markov Models which relate a sequence of states to the observations seen and help us determine the hidden states the system would have passed through. This process was distributed into three fundamental problems of Likelihood, Decoding and Training and we studied in brief how the Forward procedure, Viterbi, Baum-Welch algorithms give solutions to these problems.

6. References

1. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-450-principles-of-digital-communications-i-fa-lecture-notes/book_7.pdf
2. https://web.sonoma.edu/users/f/farahman/sonoma/courses/ces540/lectures/Chapter6_Dig_Random_Proc.pdf
3. https://en.wikipedia.org/wiki/Speech_enhancement
4. https://en.wikipedia.org/wiki/Central_limit_theorem#Applications_and_examples
5. https://en.wikipedia.org/wiki/Probability_density_function
6. <https://brilliant.org/wiki/hidden-markov-models/>
7. <https://brilliant.org/wiki/markov-chains/>
8. https://www.cse.iitb.ac.in/~nirav06/i/HMM_Report.pdf
9. <https://web.stanford.edu/~jurafsky/slp3/A.pdf>
10. <https://towardsdatascience.com/markov-chains-and-hmms-ceaf2c854788>
11. <https://towardsdatascience.com/markov-and-hidden-markov-model-3eec42298d75>
12. <https://jonathan-hui.medium.com/speech-recognition-gmm-hmm-8bb5eff8b196>

Link To GitHub Repository:

[https://github.com/AkshitGureja/
PNRP-Project-Team-17.git](https://github.com/AkshitGureja/PNRP-Project-Team-17.git)

Contributions:

- Sreenya Chitluri
 1. Theoretic Preliminaries
 2. Speech Enhancement
 3. Introduction, Results
 4. Coding: M-order filter, Markov Chains - I
- Smruti Biswal
 1. Theoretic Preliminaries
 2. Speech Enhancement
 3. Introduction, Results
 4. Coding: Gaussian noise, Markov Chains - II
- Akshit Gureja
 1. Theoretic Preliminaries
 2. Speech Recognition
 3. Results
 4. Coding: Markov Chains - III

