Environment and Sensor Robustness in Automatic Speech Recognition

Utpal Bhattacharjee

Abstract* – *Most of the presently available speech recognition systems work efficiently only in some ideal conditions. This is due to the fact that these systems are based on some assumptions related to the operating conditions. The system works efficiently if the actual working environment is identical with the environment for which the system is built. Performance of the speech recognition system considerably degrades if mismatch between the training and the testing environment occurs. In the present study, mismatch due to sensor variability and environment has been considered and Cepstral Mean Normalization (CMN) and Spectral subtraction methods have been investigated as front-end methods for the reduction of noise. A Hidden Markov Model (HMM) based speech recognition system has been built with Mel-Frequency Cepstral Coefficient (MFCC) as feature vector. It has been observed that there is a 15% enhancement of system performance in channel and environment mismatched condition compared to baseline performance when CMN and spectral subtraction methods have been applied for noise reduction.

IndexTerms—Robust Speech Recognition, MFCC, CMN, Spectral Subtraction

I. INTRODUCTION

Speech is the most natural mode of communication among human beings. Therefore, it has the potential of being the most preferred mode of interaction with the machine. A convenient and user-friendly interface for human-machine interaction is an important technological issue. Machine recognition of speech involves generation of sequence of words or sub-words which best match the given speech signal. The speech recognition may be speaker dependent or speaker independent. In speaker dependent mode, the system not only detects the intended meaning of the speech signal but also the speaker specific characteristics. On the other hand, in case of speaker independent mode, it ignores the speaker specific information but recognize only the intendant meaning. The speech recognition is a special case of pattern recognition. There are two phases of speech recognition system viz. training and testing. The process of extracting the features relevant for the characterization of speech is common to both the phases. In the training phase, the parameters of the classification model are estimated using large number of example cases. In the testing or recognition phase the characteristics of speech signal are matched with the trained models and test pattern is declared to belong to that model which best matches the test pattern.

The goal of speech recognition system is to generate the optimal word or sub-word sequence which is the linguistic constrained coding of the speech signal. In speech recognition, sentence model may be considered as sequence of word or sub-word models.

In an automatic speech recognition system, environmental mismatch causes serious degradation of the system performance. To overcome this environmental mismatch problem, some strategies of noise compensation have been proposed, which are falling into one of the following three categories [1]: Model Compensation Techniques, Pre-processing such as spectral subtraction focus on suppressing the additive noise and selection of the robust feature set.In all models the objective is to reduce mismatch between training and testing environment without loosing any important features of the speech signal. An optimal solution is likely to employ more than one of these techniques together to improve the performance of the Speech Recognizer in noisy environment [2]. In this paper we are combining Cepstral Mean Normalization with spectral subtraction method.

Mel Frequency Cepstral Coefficients (MFCCs) has been used by many speech and speaker recognition system [3,4] and proven to be one of the most successful method for speech parameterization. In the present study MFCC has been used as feature vector. To overcome the problem of mismatch due to channel variability, Cepstral Mean Normalization (CMN) has been used. Most of the widespread and successful approach for speech recognition are based on Hidden Markov Model (HMM)[5,6,7]. These models provide a reasonable statistical superstructure for both the estimation of system parameters and for efficient characterization of temporal variation in speech signal. In the present study HMM has been used as baseline speech recognition system.

The paper is organized as follows. Section II describes the techniques used for noise elimination in the speech recognition system. The theoretical details and algorithms for baseline speech recognition system are described in section III. Section IV details the dataset used and experimental setup. The experimental details and their outcome are presented in section V. The paper is concluded in section VI.

II. NOISE ELIMINATION TECHNIQUES

In the present study Spectral subtraction and Cepstral Mean Normalization (CMN) methods have been used for noise elimination from speech signal before and after feature extraction respectively. The theoretical basis of the methods has been detailed below.

Manuscript Received on January 2013.

Utpal Bhattacharjee, Department of Computer Science and Engineering, Rajiv Gandhi University, Rono Hills, Doimukh, Arunachal Pradesh, India

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

A. Spectral Subtraction

The spectral subtraction method [2,8] is based on the principle that signal-to-noise ratio (SNR) of a signal can be enhanced by subtracting the estimated noise signal from the signal itself. Let us assumed that the speech signal is distorted by a wide-band, stationary, additive noise *n(m).* The speech signal is *x(m)*, the noisy version of the speech signal is *y(m).* Then we can written,

$$\mathrm{y(m) = x(m) + n(m)} \qquad \text{--- (1)}$$

After windowing the speech signal if Discrete Fourier transform is applied to both the sides, we get

$$\mathrm{Y_w(e^{jw}) = X_w(e^{jw})} + N_w(\mathrm{e^{jw}}) \qquad \text{--- (2)}$$

Multiplying both sides by their complex conjugates we get:

$$\left|\mathrm{Y(e^{jw})}\right|^2 = \left|\mathrm{X(e^{jw})}\right|^2 + \left|N(\mathrm{e^{jw}})\right|^2 + 2\left|\mathrm{X(e^{jw})}\right|\left|N(\mathrm{e^{jw}})\right|\mathrm{cos}\emptyset \qquad \text{--- (3)}$$

where Φ is the phase difference between speech and noise: Taking the expected value of both sides we get:

$$\begin{aligned}\mathrm{E}\left\{\left|\mathrm{Y(e^{jw})}\right|^2\right\} &= \mathrm{E}\left\{\left|\mathrm{X(e^{jw})}\right|^2\right\} + \mathrm{E}\left\{\left|N(\mathrm{e^{jw}})\right|^2\right\} + \mathrm{E}\left\{2\left|\mathrm{X(e^{jw})}\right|\left|N(\mathrm{e^{jw}})\right|\mathrm{cos}\emptyset\right\} \\ &= \mathrm{E}\left\{\left|\mathrm{X(e^{jw})}\right|^2\right\} + \mathrm{E}\left\{\left|N(\mathrm{e^{jw}})\right|^2\right\} + 2\mathrm{E}\left\{\left|\mathrm{X(e^{jw})}\right|\right\}\mathrm{E}\left\{\left|N(\mathrm{e^{jw}})\right|\right\}\mathrm{E}\{\mathrm{cos}\emptyset\}\end{aligned} \qquad \text{--- (4)}$$

in power spectral subtraction it is assumed that $\mathrm{E}\{\mathrm{cos}(\emptyset)\} = 0$, hence

$$\mathrm{E}\left\{\left|\mathrm{Y(e^{jw})}\right|^2\right\} = \mathrm{E}\left\{\left|\mathrm{X(e^{jw})}\right|^2\right\} + \mathrm{E}\left\{\left|N(\mathrm{e^{jw}})\right|^2\right\} \qquad \text{--- (5)}$$

$$\left|\mathrm{X(e^{jw})}\right|^2 = \left|\mathrm{Y(e^{jw})}\right|^2 - \mathrm{E}\left\{\left|N(\mathrm{e^{jw}})\right|^2\right\} \qquad \text{--- (6)}$$

The power spectrum of noise is estimated during speech inactive periods and subtracted from the power spectrum of each frame resulting in the power spectrum of the speech. Generally Spectral subtraction is suitable for stationary or very slow varying noises. In magnitude spectral subtraction it is assumed that$\{\mathrm{cos}(\emptyset)\} = 1$, hence:

$$\begin{aligned}\mathrm{E}\left\{\left|\mathrm{Y(e^{jw})}\right|^2\right\} &= \mathrm{E}\left\{\left|\mathrm{X(e^{jw})}\right|^2\right\} + \mathrm{E}\left\{\left|N(\mathrm{e^{jw}})\right|^2\right\} + 2\mathrm{E}\left\{\left|\mathrm{X(e^{jw})}\right|\right\}\mathrm{E}\left\{\left|N(\mathrm{e^{jw}})\right|\right\} \\ &= \left(\mathrm{E}\left\{\left|\mathrm{X(e^{jw})}\right|\right\} + \mathrm{E}\left\{\left|N(\mathrm{e^{jw}})\right|\right\}\right)^2\end{aligned} \qquad \text{--- (7)}$$

$$\mathrm{E}\left\{\left|\mathrm{Y(e^{jw})}\right|\right\} = \mathrm{E}\left\{\left|\mathrm{X(e^{jw})}\right|\right\} + \mathrm{E}\left\{\left|N(\mathrm{e^{jw}})\right|\right\} \qquad \text{--- (8)}$$

$$\left|\mathrm{X(e^{jw})}\right| = \left|\mathrm{Y(e^{jw})}\right| - \mathrm{E}\left\{\left|N(\mathrm{e^{jw}})\right|\right\} \qquad \text{--- (9)}$$

The magnitude spectrum of the noise is estimated during speech inactive periods and, again, assuming that the variations of noise spectrum are tolerable, the magnitude spectrum of speech is estimated by subtracting the average spectrum of noise from each frame [2].

B. Cepstral Mean Normalization

A large class of environmental distortions in the speech signal is due to the channel mismatch which can be represented as a simple linear filter [9,10]

$$y(m) = x(m) * h(m) \qquad \text{--- (10)}$$

Where $h(m)$is the linear filter and * denotes the convolution. In the frequency domain we can write

$$Y(k) = X(k)H(k) \qquad \text{--- (11)}$$

Taking logarithm we get

$$\log Y(k) = \log X(k) + \log H(k) \qquad \text{--- (12)}$$

Thus, the effect of linear distortion is to add a constant factor to the amplitude of the speech signal in log domain. The normal cepstral processing can be written as

$$\begin{aligned}c'[k] &= Cepst(\log Bin(FFT(y(m)))) \\ &= Cepst(\log Bin(FFT(x(m) * h(m)))) \\ &= Cepst(\log Bin(X(k).H(k)))\end{aligned} \qquad \text{--- (13)}$$

Since the mapping from the spectral domain to cepstral domain is linear, we can model the effect of linear filter to just adding a constant vector in the cepstral domain

$$c'[k] = c[k] + h[k] \qquad \text{--- (14)}$$

The robustness can be achieved by estimating h[k] and subtracting it from the observation $c'[k]$.

Now, for a given set of cepstral vector $c_t, 1 \leq t \leq N$, we can compute the mean:

$$\bar{c} = \frac{1}{N}\sum_{t=1}^{N} c_t \qquad \text{--- (15)}$$

Cepstral mean normalization produces a new output $\hat{c}_t$which is given by

$$\hat{c}_t = c_t - \bar{c} \qquad \text{--- (16)}$$

Let us assume that signal $c_t$is processed by a linear filter and let h is the cepstral vector corresponding to that linear filter. The output of the linear filter is:

$$y_t = c_t + h \qquad \text{--- (17)}$$

The mean of y_t

$$\begin{aligned}\bar{y} &= \frac{1}{N}\sum_{t=1}^{N} y_t \\ &= \frac{1}{N}\sum_{t=1}^{N}(c_t + h) = \bar{c} + h\end{aligned} \qquad \text{--- (18)}$$

After cepstral mean normalization

$$\hat{y}_t = y_t - \bar{y} = \hat{c}_t \qquad \text{--- (19)}$$

Thus, the influence of the filter *h* is eliminated.

III. BASELINE SYSTEM

A baseline speech recognition system has been developed using MFCC as feature vector and HMM as the recognizer. The theoretical details of MFCC and HMM are detailed below:

A. Mel-Frequency Cepstral Coefficient

The speech signal is divided into frames where discrete Fourier transform (DFT) has been computed for each frame. For the discrete time signal *x(n)* with length *N*, the DFT is given by

$$X(k) = \sum_{n=0}^{N-1} w(n)x(n)\exp(-j2\pi kn/N) \text{ --- } (20)$$

for*k* = 0, 1, ….., *N*-1, where k corresponds to the frequency $f(k) = kf_k/N$, f_k is the sampling frequency in Hertz and *w(n)* is a time-window. In the present study Hamming windows defined by $w(n) = 0.54 - 0.46\cos(2\pi n/N)$ has been used because of its computational simplicity.

The magnitude spectrum $|X(k)|$ is now scaled in both frequency and magnitude. First the frequency is scaled logarithmically using the Mel filter bank *H(k,m)* and then the logarithm is taken giving

$$X'(m) = \ln\left(\sum_{k=0}^{N-1} |X(k)|.H(k,m)\right) \text{ --- } (21)$$

for*m* = 1,2,……, *M*, where *M* is the number of filter banks and *M*<<*N*. The Mel filter bank is a collection of triangular filters defined by the center frequencies $f_c(m)$, written as

$$H(k,m) = \begin{cases} 0, \; f(k) < f_c(m-1) \\ \dfrac{f(k) - f_c(m-1)}{f_c(m) - f_c(m-1)}, \\ \qquad f_c(m-1) \le f(k) < f_c(m) \\ \dfrac{f(k) - f_c(m+1)}{f_c(m) - f_c(m+1)}, \\ \qquad f_c(m) \le f(k) < f_c(m+1) \\ 0, \; f(k) \ge f_c(m+1) \end{cases} \text{ ---}(22)$$

The center frequencies of the filter bank are computed by approximating the Mel scale with

$$\phi = 2595\log_{10}\left(\frac{f}{700} + 1\right) \text{ --- } (23)$$

which is a common approximation. That equation is non-linear for all frequencies. Then a fixed frequency resolution in the Mel scale is computed, corresponding to a logarithmic scaling of the repetition frequency, using $\Delta\phi = (\phi_{\max} - \phi_{\min})/(M+1)$ where $\phi_{\max}$ is the highest frequency of the filter bank on the Mel scale, computed from $f_{\max}$ using equation (23), $\phi_{\min}$ is the lowest frequency in Mel scale, having a corresponding $f_{\min}$ and *M* is the number of filter bank. The values considered for the parameters in the present study are: $f_{\max}$=11.025 KHz, $f_{\min}$=0 Hz and *M*=30. The center frequencies on the Mel scale are given by $\phi_c(m) = m.\Delta\phi$ for *m* = 1, 2, 3, …..,*M*. To obtain the center frequencies in Hertz, inverse of the equation (23) is applied, which is given by

$$f_c(m) = 700\left(10^{\phi_c(m)/2595} - 1\right) \text{ --- } (24)$$

Equation (24) is inserted into equation (22) to give the Mel filter bank. Finally, the MFCCs are obtained by computing the discrete cosine transform of $X'(m)$ using

$$c(l) = \sum_{m=1}^{M} X'(m)\cos(l\frac{\pi}{M}(m - \frac{1}{2})) \qquad (25)$$

for*l* = 1, 2, 3, ….., *M* where *c(l)* is the *l*th MFCC.

The time derivative is approximated by a linear regression coefficient over a finite window, which is defined as

$$\Delta\, c_t(l) = \left[\sum_{K=2}^{2} k\; c_{t-k}(m)\right].G,\; 1 \le l \le M \qquad (26)$$

where $c_t(l)$ is the *l*th cepstral coefficient at time *t* and *G* is a constant used to make the variances of the derivative terms equal to those with the original cepstral coefficients.

B. HMMs for speech recognition

In automatic speech recognition, the task is to find the most likely sequence of words W given some acoustic input, or

$$\widehat{W} = \arg\max_{W\in w} P(W|X) \qquad (27)$$

Here, *X* = {*x*1, *x*2,……….....,*x*N} is the sequence of "acoustic vectors" – or "feature vectors" – that are "extracted" from the speech signal, and we want to find *W* as the sequence of word *W* (out of all possible word sequences *w*), that maximizes *P (W|X)*

The likelihood of a sequence with respect to a HMM (the likelihood of an observation sequence X = = {x1, x2,……….,xN} for a given hidden Markov model with parameters θ expands as follows:

$$p(X|\theta) = \sum_{Every\ possible\ Q} P(X,Q|\theta) \text{--- } (28)$$

Calculating the likelihood in this manner is computationally expensive, particularly for large models or long sequence. It can be done with a recursive algorithm (forward –backward algorithm), which reduces the complexity of the problem.

Optimal state sequence

In speech recognition and several other pattern recognition applications, it is useful to associate an "optimal" sequence of states to a sequence of observations, given the parameters of a model. For instance, in the case of speech recognition, knowing which frames of features " belong" to which state allows to locate the word boundaries across time. This is called *alignment* of acoustic feature sequences.

A "reasonable" optimality criterion consists of choosing the state sequence (or path) that has the maximum likelihood with respect to a given model. This sequence can be determined recursively via the *Viterbi algorithm.*

This algorithm makes use of two variables:

- $\delta_n(i)$is the highest likelihood of a *single* path among all the paths ending in the state $s_i$at time *n:*

$$\delta_n(i) = \min_{q_1,q_2,\dots,q_{n-1}} p(q_1, q_2, \dots, q_{n-1}, q_n = s_i, x_1, x_2, \dots, x_n|\theta) \text{ --- } (29)$$

- a variable $\psi_n(i)$ which allows to keep track of the "best path" ending in state s_i at a time n:
  $\psi_n(i) = \arg\min_{q_1,q_2,\ldots,q_{n-1}} p(q_1, q_2, \ldots, q_{n-1}, q_n = s_i, x_1, x_2, \ldots, x_n|\theta)$ --- (30)

The idea of the Viterbi algorithm is to find the most probable path for *each intermediate* and finally for the terminating state in the trellis. At each time *n* only the most likely path leading to each states_i 'survives'.

The Viterbi Algorithm

For a HMM with N_s states.

1. Initialization

   $\delta_1(i) = \pi_i . b_{i,x_i}, i = 1, \ldots, N_s$ --- (31)

   $\psi_1(i) = 0$ ---(32)

   Where π_i is the prior probability of being in state s_i at time n =1.

2. Recursion

   $$\delta_1(j) = \max_{1\le i\le N_s} \left(\delta_{n-1}(i)a_{ij}\right). b_{j,xn},$$
   $$2 \le i \le N, 1 \le j \le N_s$$
   ---(33)

   $$\psi_n(j) = arg \max_{1\le i\le N_s} (\delta_{n-1}(i)a_{ij}),$$
   $$2 \le i \le N, 1 \le j \le N_s$$
   ---(34)

   "*Optimal policy is composed of optimal sub-policies*" : find the path that leads to a maximum likelihood considering the best likelihood at the previous step and the transitions from it; then multiply by the current likelihood given current state. Hence, the best path is found by induction .

3. Termination

   $P^*(X|\theta) = \max_{1\le i\le N_s} \delta_N(i)$ ---(35)

   $q_N^* = \arg\max_{1\le i\le N_s} \delta_N(i)$ ---(36)

   Find the best likelihood when the end of the observation sequence *t*=*T* is reached.

4. Backtracking

   $$Q^* = \{q_1, q_2, \ldots \ldots \ldots q_N^*\} \text{so that}$$
   $$q_n^* = \Psi_{n+1}(q_{n+1}^*), n = N-1, N-2, \ldots \ldots \ldots 1$$
   ---(37)

   Decode the best sequence of states from the vectors.

Baum Welch

The Baum–Welch algorithm is used to estimate the model parameters when the state path is unknown. Given sequences O^1, O^2,...,O^kwe wish to determine $\lambda = \{a_{ij}, e_i(.), \pi_i\}$. We generally want to choose parameters that will maximize the likelihood of our data.

However, finding a global maximum is intractable. We would have to enumerate over all parameter sets, λ_k, and then calculate

$$Score\ (\lambda_k) = \sum_d P(O^d|\lambda_k) = \sum_d \sum_Q P(O^d|\lambda_k, Q)$$

---(38)

for eachλ_k. The algorithm used for selecting the parameter values is Expectation Maximum (EM) algorithms. They all work by gassing initial parameter values, then estimating the likelihood of the data under the current parameters. These likelihoods can then be used to re-estimate the parameters, iteratively until a local maximum is reached.

The alphabet and the number of states, N, are fixed . We are given the observed sequences (denoted $\mathbf{O^d = O^d_1 , O^d_2}$, …). The algorithm is as follows:

1. Initialize the values for $\lambda = (\pi, a_{ij}, e_i(.))$.
2. Determine probable paths $Q^d = q_1^d, q_2^d$,….
3. Given the current estimate of λ, count the expected number of transitions$A_{ij}$from state *i* to state *j*, .
4. Count the value of $E_i(\sigma)$, the expected number of times character σ is emitted from state *i*.
5. Re-estimate λ (π, a_{ij},e_i(.)) from $A_{ij}$and $E_i(\sigma)$,
6. if not converged go to step 2

For a given sequence, O^2, probability of transiting from state *i* to *j* at time *t is*

$$P(q_t^d = i, q_{t+1}^d = j|O^d, \lambda) = \frac{P(q_t^d = i, q_{t+1}^d = j|O^d)}{P(O^d)}$$
$$= \frac{\alpha_t(i)a_{ij}e_j(O_{t+1}^d)\beta_{t+1}(j)}{P(O^d)}$$

---(39)

The term $\alpha_t(i)$ is the probability that the model has emitted symbols O^d_1 ,…, O^d_t is in state S_i at time *t*. This probability can be obtained using the Forward algorithm. Similarly, the Backward algorithm yields $\beta_{t+1}(j)$, the probability of emitting the rest of the sequence if we are in state *j* at time t+1. The remaining two terms, a_{ij} and e_j (O^d_{t+1}) give the probability of making the transition from *i* to *j*and emitting the *t+1*st character.

From this we can estimate

$$A_{ij} = \sum_d \frac{1}{P(O^d)} \sum_t \alpha(t,i)a_{ij}e_i(O_{t+1}^d)\beta(t+1,i)$$

---(40)

The probability of O^d can be estimated using current parameter values using the Forward algorithm. Similarly,

$$E_i(\sigma) = \sum_d \frac{1}{P(O^d)} \sum_{\{t|O_t^d=\sigma\}} \alpha(t,i)a_{ij}\beta(t,i)$$

---(41)

From A_{ij} and $E_i(\sigma)$ we re-estimate the parameters.

IV. EXPERIMENTAL SETUP AND DATABASE USED

A. Database Description

In order to carry out the experiments reported in this paper, a database consist of isolated utterance of 10 (ten) Assamese digits has been created at different environmental conditions and using two recording device. The details of the datasets have been given below.

*Dataset-I:*400 isolated utterances of ten Assamese digits spoken by 20 speakers (10 male and 10 female) of age groups 20~50. The recording has been done in soundproof room using headphone microphone and laptop microphone in parallel.

Database-II: 400 isolated utterances of ten Assamese digits spoken by the same 20 speakers. Recording has been done using headphone microphone and laptop microphone in parallel in an open terrace of a two storied building located in a busy roadside.

B. *Experimental setup*

The sound from the speaker has been directly digitized in WAV PCM format and sampling at 16 KHz frequency with 16 bit mono quantization. A pre-emphasis filter $H(z) = 1 - 0.96z^{-1}$ has been applied before framing. The digitized speech signal is blocked into frame of 30 microseconds with a frame rate of 10 microseconds. Each frame is multiplied by a Hamming Window. The windowed signal from each frame is pass through 30 triangular shaped mel-filter bank spaced on Mel-scale. The log-compressed filter outputs are converted to cepstral coefficients by DCT. In order to reduce the computational cost some of the less useful cepstral coefficients can be discarded. In the present study cepstral coefficient from 6-25 and its first order derivative has been considered. Thus, the feature set for each frame consists of 40 components. A Hidden Markov Model (HMM) based speaker recognition system has been developed for the recognition of the Digits of Assamese language. The HMM has 5 states and each state contains 256 Gaussian components.

V. EXPERIMENT AND RESULTS

The Dataset-I, as described in section IV has been used for training the models for the recognition of the digits of Assamese language. One separate model has been created for each digit. Ten utterances of each digit containing equal number of male and female voices have been considered for training the models. The models are created using only headphone microphone data. Remaining utterances have been considered for testing the system. Testing has been done using data from same microphone as well as different microphones. Another parameter tested in this experiment is speaker variability. Testing has been done to evaluate the performance of the system when same and different group of speaker were considered for training and testing keeping other parameters constant. The result of the experiments are given in table-1

Table-1: Recognition of Assamese spoken digits using clean version of the speech signal

Digit	Accuracy (in %)			
	Same Speaker/ Same Sensor	Same Speaker/ Different Sensor	Different Speaker/ Same Sensor	Different Speaker/ Different Sensor
Zero	94.5	84.5	88.5	83.0
One	96.0	86.0	91.0	82.5
Two	95.0	84.0	90.0	83.0
Three	98.5	83.5	91.5	82.5
Four	98.0	85.5	91.0	81.5
Five	97.0	86.0	90.0	80.5
Six	98.0	86.5	92.5	82.0
Seven	97.5	85.0	89.5	81.5
Eight	97.0	84.0	89.5	82.5
Nine	95.0	85.0	89.0	82.5
Average Accuracy	**96.65**	**85.0**	**90.25**	**82.15**

The same four experiments have been repeated with Dataset-II. The results of the experiments are given in table-2.

Table-2: Recognition of Assamese spoken digits using noisy version (Dataset-II) of the speech signal

Digit	Accuracy (in %)			
	Same Speaker/ Same Sensor	Same Speaker/ Different Sensor	Different Speaker/ Same Sensor	Different Speaker/ Different Sensor
Zero	70.9	63.4	66.4	62.3
One	72.0	64.5	68.3	61.9
Two	71.3	63.0	67.5	62.3
Three	73.9	62.6	68.6	61.9
Four	73.5	64.1	68.3	61.1
Five	72.8	64.5	67.5	60.4
Six	73.5	64.9	69.4	61.5
Seven	73.1	63.8	67.1	61.1
Eight	72.8	63.0	67.1	61.9
Nine	71.3	63.8	66.8	61.9
Average Accuracy	**72.5**	**63.8**	**67.7**	**61.6**

Now, Spectral subtraction has been applied before framing the speech signal. The same set of experiments has been repeated with the noisy version of the speech data and results of the experiments are given in table-3.

Table.3: Recognition of Assamese spoken digits using noisy version of the speech signal and using spectral subtraction method

Digit	Accuracy (in %)			
	Same Speaker/ Same Sensor	Same Speaker/ Different Sensor	Different Speaker/ Same Sensor	Different Speaker/ Different Sensor
Zero	85.1	67.8	79.7	66.0
One	86.4	69.0	82.0	65.6
Two	85.6	67.4	81.0	66.0
Three	88.7	67.0	82.3	65.6
Four	88.2	68.6	82.0	64.8
Five	87.4	69.0	81.0	64.0
Six	88.2	69.4	83.3	65.2
Seven	87.7	68.3	80.5	64.8
Eight	87.4	67.4	80.5	65.6
Nine	85.6	68.3	80.2	65.6
Average Accuracy	**87.0**	**68.2**	**81.2**	**65.3**

In the next experiment, the speech signal is framed directly and its cepstral coefficients are calculated. After cepstral calculation, we apply cepstral mean normalization and the same set of experiments has been repeated with noisy version of the speech signal. The results of the experiments are given in table-4.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Table.4: Recognition of Assamese spoken digits using noisy version of the speech signal and using Cepstral Mean Normalization method

Digit	Accuracy (in %)			
	Same Speaker/ Same Sensor	Same Speaker/ Different Sensor	Different Speaker/ Same Sensor	Different Speaker/ Different Sensor
Zero	74.4	77.3	69.7	76.0
One	75.6	78.7	71.7	75.5
Two	74.9	76.9	70.9	76.0
Three	77.6	76.4	72.0	75.5
Four	77.2	78.2	71.7	74.5
Five	76.4	78.7	70.9	73.7
Six	77.2	79.2	72.9	75.0
Seven	76.8	77.8	70.5	74.5
Eight	76.4	76.9	70.5	75.5
Nine	74.9	77.8	70.1	75.5
Average Accuracy	**76.1**	**77.8**	**71.1**	**75.2**

Finally, we combine both the spectral subtraction and Cepstral Mean Normalization method. Spectral subtraction is applied before the framing of the speech signal whereas Cepstral mean Normalization is applied after calculating the cepstral coefficients. The results of the experiments are reported in table-5.

Table-5: Recognition of Assamese spoken digits using noisy version of the speech signal and using combination of Spectral subtraction and Cepstral Mean Normalization method

Digit	Accuracy (in %)			
	Same Speaker/ Same Sensor	Same Speaker/ Different Sensor	Different Speaker/ Same Sensor	Different Speaker/ Different Sensor
Zero	95.7	95.1	93.0	93.5
One	97.2	96.8	95.6	92.9
Two	96.3	94.5	94.5	93.5
Three	99.8	93.9	96.0	92.9
Four	99.2	96.2	95.6	91.7
Five	98.3	96.8	94.5	90.6
Six	99.2	97.4	97.2	92.3
Seven	98.7	95.7	93.9	91.7
Eight	98.3	94.5	93.9	92.9
Nine	96.3	95.7	93.5	92.9
Average Accuracy	97.9	95.6	94.8	92.4

VI. CONCLUSION

From the above experiments, it has been observed that the performance of a speech recognizer degrades considerably with environmental condition. It the present work, effort has been made to evaluate the performance of the speech recognizer in two types of noisy conditions – moderate level environmental noise, which we consider to be the most typical working environment for a speech recognizer and microphone (sensor) mismatch condition. From table-1 and table-2 it has been observed that sensor variability has more prominent negative influence on system performance than typical environmental noise. Another important observation made during the present study is that for MFCC features and HMM based recognizer, speaker variability has very less influence on the system performance. Two most commonly applied noise elimination method for speech recognition has been applied separately and also both of them has been combined in the present study. It has been observed from the experimental result that spectral subtraction method is efficient is eliminating environmental noise. Nearly 15% improvement in system performance has been achieved when only environmental noise factor is considered. However, only 5% improvement has been achieved for sensor mismatch problem. When only Cepstral Mean Normalization has been applied, a sharp improvement of 14% system performance has been achieved for channel mismatch condition and nearly 4% improvement has been achieved for environmental noise. To explore the best of both the methods, both the methods are combined and it has been observed that under moderately noisy environment and channel mismatched condition, the system given a consistent performance.

REFERENCES

1. Z. Junhui, X. Xiang and K. Jingming, Noise Suppression Based on Auditory-Like Filters for Robust Speech Recognition, *Proc. ICSP*'02, 560-563, 2000.
2. Steven F. Boll, Suppression of Acoustic Noise in Speech using Spectral Subtraction, *IEEE Transaction on ASSP*, 27(2), 113-120, 1979.
3. Hossan, M.A.; Memon, S.; Gregory, M.A.; , "A novel approach for MFCC feature extraction," *Signal Processing and Communication Systems (ICSPCS)*, 2010 4th International Conference on , vol., no., pp.1-5, 13-15 Dec. 2010
4. Patel, I.; Rao, Y.S.; , "Speech Recognition Using Hidden Markov Model with MFCC-Subband Technique," Recent Trends in Information, Telecommunication and Computing (ITC), 2010 International Conference on , vol., no., pp.168-172, 12-13 March 2010.
5. L.R. Rabiner, A Tutorial on Hidden Markov Model and Selected Application in Speech Recgnition, Proc. of IEEE, Vol. 77, No. 2, PP. 257-285, 1989.
6. Ashraf, J.; Iqbal, N.; Khattak, N.S.; Zaidi, A.M.; , "Speaker Independent Urdu speech recognition using HMM," *Informatics and Systems (INFOS)*, 2010 The 7th International Conference on , vol., no., pp.1-5, 28-30 March 2010
7. D. Van Compernolle, Noise Adaptation in a Hidden Markov Model Speech Recognition System, *Computer Speech and Language*, 152-167, (1989).
8. Nehe, N.S.; Holambe, R.S.; , "Isolated Word Recognition Using Normalized Teager Energy Cepstral Features," *Advances in Computing, Control, & Telecommunication Technologies, 2009*. ACT '09. International Conference on , vol., no., pp.106-110, 28-29 Dec. 2009.
9. Longbiao Wang; Kitaoka, N.; Nakagawa, S.; , "Robust Distant Speech Recognition by Combining Position-Dependent CMN with Conventional CMN," *Acoustics, Speech and Signal Processing, 2007*. ICASSP 2007. IEEE International Conference on , vol.4, no., pp.IV-817-IV-820, 15-20 April 2007
10. Molau, S.; Hilger, F.; Ney, H.; , "Feature space normalization in adverse acoustic conditions," *Acoustics, Speech, and Signal Processing, 2003*. Proceedings. (ICASSP '03). 2003 IEEE International Conference on , vol.1, no., pp. I-656- I-659 vol.1, 6-10 April 2003

AUTHORS PROFILE

Utpal Bhattacharjee, received his Master of Computer Application (MCA) from Dibrugarh University, India and Ph.D. from Gauhati University, India in the year 1999 and 2008 respectively. Currently he is working as an Associate Professor in the department of Computer Science and Engineering of Rajiv Gandhi University, India. His research interest is in the field of Speech Processing and Robust Speech/Speaker Recognition.