Simple speech recognition using Hidden Markov Model

Yerin Hong
Dept. of Electronics and Communications Engineering
University of Kwangwoon
Seoul, Korea
yerin0927@kw.ac.kr

***Abstract*—This paper recognizes the human speech using Hidden Markov Model (HMM). The human speeches used for recognition are simple WAV file of number. We used FIR filter for removing the noise, Mel-frequency Cepstrum (MFC) transform for extracting the feature of each speech. The speech signals are clustered by K-means algorithm, trained by Baum-Welch algorithm and test are performed by Viterbi algorithm. We can successfully classify all of the speech signals.**

Keywords—Speech recognition; FIR filter; Mel-frequency cepstrum; EM; HMM;

I. Introduction

There are many researches of input sensor that enable removal of the keyboards or mousses. One of the researches is speech signals for input sensor. In this paper, we study the speech recognition (SR) for using as an input signal. SR is the inter-disciplinary sub-field of computational linguistics which incorporates knowledge and research in the linguistics, computer science, and electrical engineering fields to develop methodologies and robotics.

The speech signals are time varying system that has a dynamic pattern of 2 dimensions. So we decided to use the Hidden Markov Model (HMM) algorithm instead of the static classifier as Linear Regression, the Support Vector Machine (SVM) etc. Since a speech signal can be viewed as a piecewise stationary signal or a short-time stationary signal, HMM algorithm is suitable.

Another reason why HMMs are popular is because it can be trained automatically and are simple and computationally feasible to use. In SR, the HMM would output a sequence of n-dimensional real-valued vector, outputting one of these every 10ms. The vectors would consist of cepstral coefficients, which are obtained by taking a Fourier transform (FT) of a short time window of speech and decorrelating the spectrum using cosine transform, then taking the first coefficients. The HMM will tend to have in each state a statistical distribution that is a mixture of diagonal covariance Gaussians, which will give a likelihood for each observed vector. Each speech words will have a different output distribution and a HMM for a sequence of speech words is made by concatenating the individual trained HMMs for the separate words.

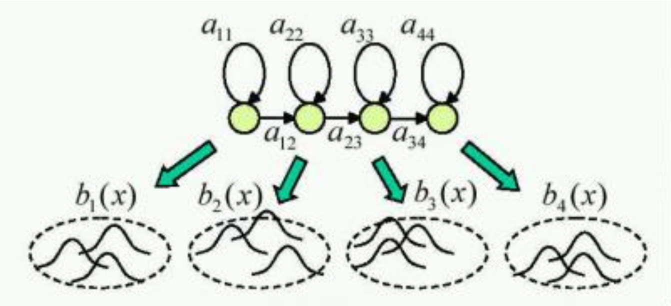


Figure 1 Simple HMM algorithm

II. Speech Recognition using HMM

In this paper, SR procedure is consist of 4-step. Two front steps are pre-processing step, filtering and feature extracting. Next, we perform training on the speech signals. The speech signals are simple number speeches. Finally we test the speech recognition using Viterbi algorithm.

A. Filtering

We use the FIR filter for removing the noise on each of training or test speeches. The FIR filter which coefficients are ‘1’ and ‘-0.9375’ is simple and doesn’t have feedback. Also it is possible to keep the phase between the input signal and the output signal.

B. Features extracting of speech signals

Mel-frequency Cepstrum (MFC) transform is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a non-linear mel scale of frequency. The value of MFC transform is contained by vectors of 12 dimensions. These vectors are the feature of the input speech.

The MFC transform’s procedure is very complex, so we use the function ‘mlcepst’ in VOICEBOX that is MATLAB tool box easily can be implemented. Mel-frequency cepstrum coefficients (MFCCs) that the results of this function contain the features of each speech signals.

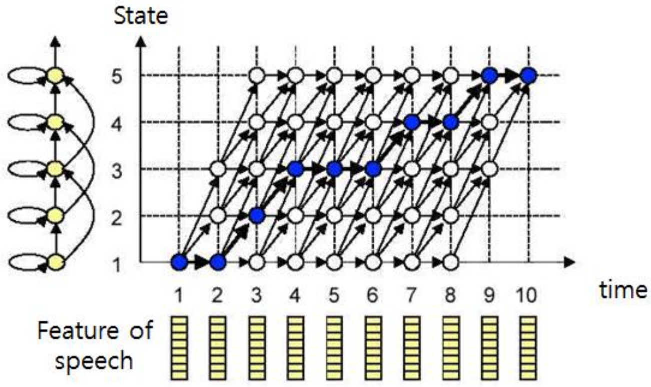


Figure 2 Simple Viterbi decoding example

C. Training

To train the speech signals, we use the Baum-Welch algorithm that is used to find the unknown parameters of a HMM. The Baum-Welch algorithm is based on Expectation Maximization (EM) algorithm to find the maximum likelihood estimate of the parameters of HMM given a set of observed feature vectors. It makes use of the forward-backward algorithm.

Decoding of the speech would probably use the Viterbi algorithm to find the best path, and here there is a choice between dynamically creating a combination HMM, which includes both the acoustic and language model information, and combining it statically beforehand.

A possible improvement to decoding is to keep a set of good candidates instead of just keeping the best candidate, and to use a better scoring function to rate these good candidates so that we may pick the best one according to this refined score. The set of candidates can be kept either as a list or as a subset of the models. Re-scoring is usually done by trying to minimize the Bayes risk.

D. Test

To recognize the speech signals, we use the Viterbi algorithm between training data and test data with transition matrix. A generalization of the Viterbi algorithm, termed the max-sum algorithm can be used to find the most likely assignment of all or some subset of latent variables in a large number of graphical models as Bayesian networks, Markov random fields and conditional random fields.

III. Result

We perform the implement using simple speech signals, one to ten. Figure 3 which is the result of FIR filtering using number 1 WAV file shows that the FIR filter effectively removes the noise on speech signals. And extracting features of speech signals is performed with melcepst function VOICEBOX on MATLAB. Figure 4 which is the result of melcepst function using number 1 WAV file shows the vectors of 12 dimensions which are represented the difference between each speech.

A. FIR filter

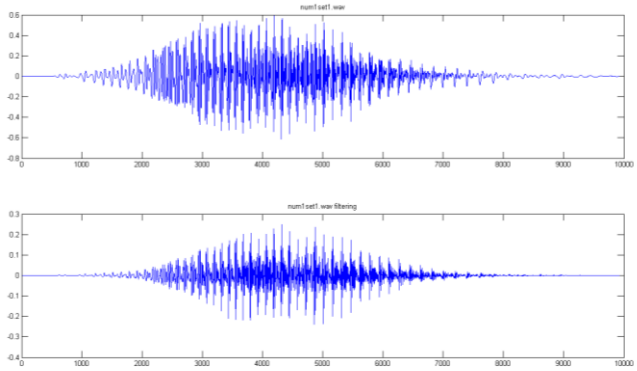


Figure 3 FIR filtering of number 1 WAV file

B. Extracting features of speech signals

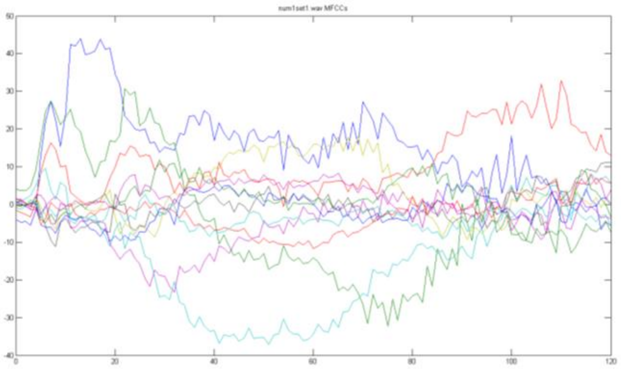


Figure 4 MFCCs of number 1 WAV file

C. Training

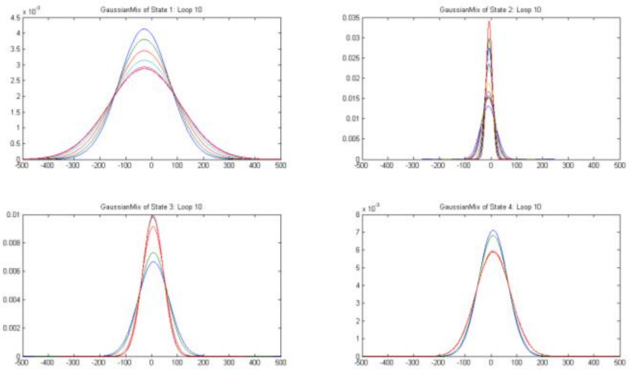


Figure 5 Converging the Gaussian distribution of each state at Number 10: loop 10

The speech signals have 4 states which initial values are 3. The number of training loop is 10, 20 and 30. Figure 5 represent converging the Gaussian distribution of each state at Number 10 in 10 loops. Figure 6 represent converging the Gaussian distribution of each state at Number 10 in 30 loops. We can show that a HMM of number 10 is converged at loop 18. In Figure 7, Figure 8, and Figure 9, we can show that Gaussian mixture in 10 loops is difference between 20 and 30 loops.

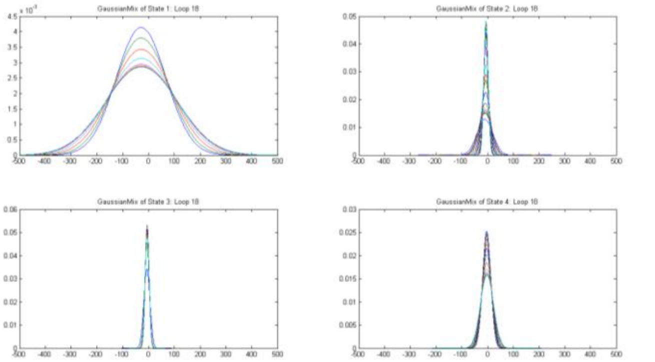


Figure 6 Converging the Gaussian distribution of each state at Number 10: loop 30

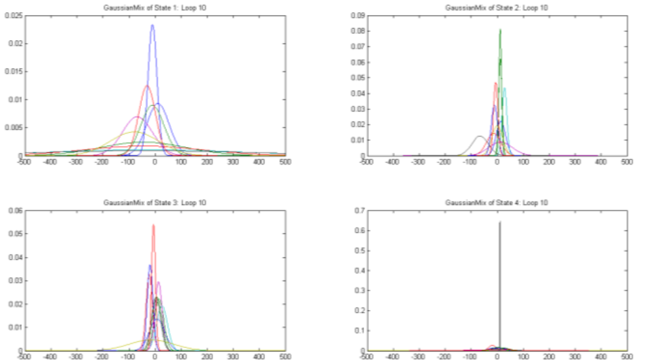


Figure 7 Gaussian mixture of each state: the number of training loop is 10

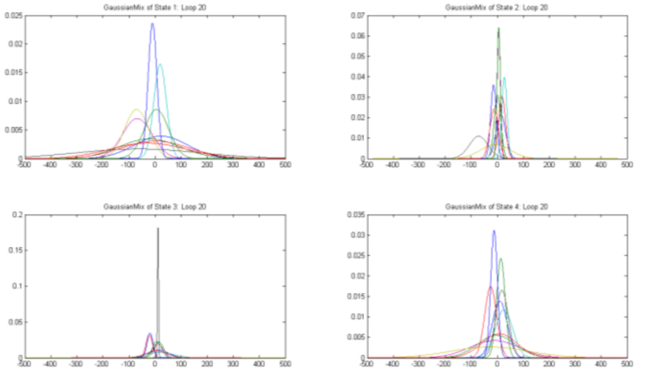


Figure 8 Gaussian mixture of each state: the number of training loop is 20

IV. Conclusion

In this paper, we study the SR system used for input signal of computer. This paper describes the HMM algorithm and performs experiments using HMM to recognize speech signals. if the number of feature vectors is increased, the total probability is infinite number or Not-a-Number (NaN) value. But In general result of mel-frequency transform, short-term words as human speech of number, these experiments are successfully. So we can expect the input system using the speech signals.

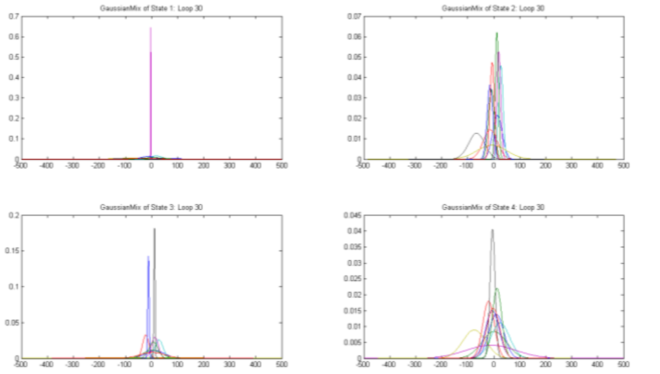
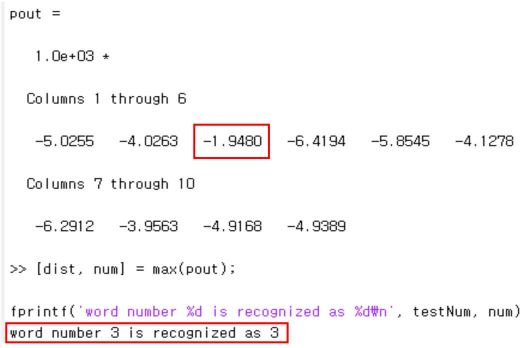


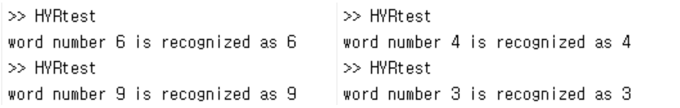
Figure 9 Gaussian mixture of each states: the number of training loop is 30

D. Test



```
pout =

   1.0e+03 *

  Columns 1 through 6

   -5.0255   -4.0263   -1.9480   -6.4194   -5.8545   -4.1278

  Columns 7 through 10

   -6.2912   -3.9563   -4.9168   -4.9389

>> [dist, num] = max(pout);

fprintf('word number %d is recognized as %d₩n', testNum, num)
word number 3 is recognized as 3
```

Figure 10 Distances of a test speech signal and recognition result



```
>> HYRtest
word number 6 is recognized as 6
>> HYRtest
word number 9 is recognized as 9
```

```
>> HYRtest
word number 4 is recognized as 4
>> HYRtest
word number 3 is recognized as 3
```

Figure 11 Recognition of random WAV file

References

[1] Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." *Proceedings of the IEEE* 77.2 (1989): 257-286.

[2] Bilmes, Jeff A. "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models." *International Computer Science Institute* 4.510 (1998): 126.

[3] Bishop, Christopher M. "Pattern Recognition and Machine Learning (Information Science and Statistics) Springer-Verlag New York." *Inc. Secaucus, NJ, USA*(2006).

[4] Tu, Stephen. "Derivation of baum-welch algorithm for hidden markov models." (2015).

[5] Gales, Mark, and Steve Young. "The application of hidden Markov models in speech recognition." *Foundations and trends in signal processing* 1.3 (2008): 195-304.

[6] Nilsson, Mikael, and Marcus Ejnarsson. "Speech recognition using hidden markov model." (2002).