International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

Improved Speech Recognition Processes using Hybrid Genetic Vector Quantizer

Sakshi Choudhary†* and Sarabjeet Kaur†

†ECE Department. Doon Valley Institute of Engineering and Technology, Karnal, India

Accepted 15 July 2015, Available online 20 July 2015, **Vol.5, No.4 (Aug 2015)**

Abstract

Speech recognition basically means talking to a computer, having it recognize what Speakers are saying. Speech is common and efficient form of communication method for people to interact with each other. The person would like to interact with computer via speech. It can be accomplished by speech recognition system in which computer identifies the word spoken by a speaker into a microphone. Speech recognition is becoming more complex and a challenging task. The research is focusing on large vocabulary, continuous speech capabilities and speaker independence. This paper reviews methods and technologies available for ASR process

Keywords: *Speech Recognition, ASR, Feature Extraction, Speech Models*

1. Introduction

Speech recognition systems have an expansive scope of requests from isolated-word recognition as in term dialing and voice-control of mechanisms to constant usual speech recognition as in auto-dictation or broadcast-news transcription. The Most useful speech recognition systems encompass of two modules: the front conclude feature module and back conclude association module. Figure 1 displays a general arrangement of a speech recognition system.

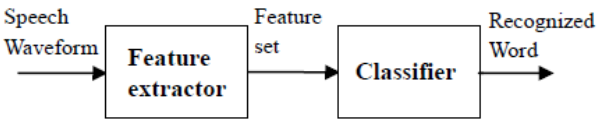


Figure 1 General arrangement of a Speech Recognition System (Frikha, Mondher *et al,* 2012)

The task of ASR is to take an acoustic waveform as an input and to produce a string of words as the output. The following figure 2 shows the general idea of ASR system

In order to understand speech, the arrangement normally consists of two phases. They are shouted pre-processing and post processing. The first phase Pre-processing involves feature extraction and the post-processing period embodies of constructing a speech recognition engine. Speech recognition engine generally consists of vision concerning constructing an aural ideal, lexicon and grammar. After all these features are given accurately, the recognition engine recognizes the best probable match for the given input, and it returns the understood word.

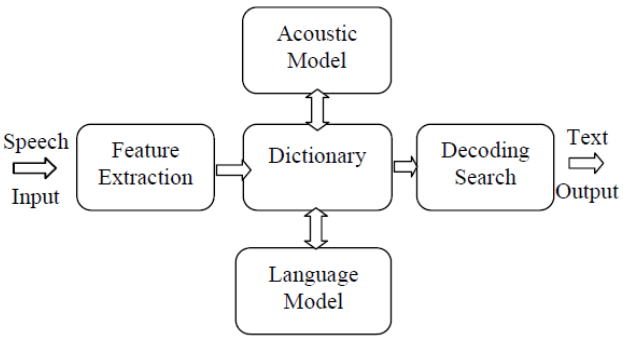


Figure 2 General idea of ASR system

A vital task of growing each ASR arrangement is to select the appropriate feature extraction method and the recognition approach. The appropriate extraction and recognition method can produce good accuracy for the given application. Hence, these two main constituents are studied and contrasted established on its merits and demerits to find out the best method for speech recognition system. The assorted kinds of feature extraction and speech recognition ways are clarified in the pursuing section.

1.1 Feature Extractor

The designs of the front conclude feature extraction module is a relevant aspect for the presentation of the speech recognizer because this module is aimed to

*Corresponding author: **Sakshi Choudhary**

remove the discriminative data utilized by the association module to present recognition. Front conclude design has been an span of alert scrutiny in the last insufficient decades. The two front conclude dominant ways in speech recognition are established on Mel frequency cepstral coefficient (MFCC) (Srinivasan,2012) and perceptual linear forecast (PLP) (Huang, Yi-bo *et al*, 2014). They are the most extensively utilized aural features in present ASR systems. The steps followed in computing those features are methodical in figure 2.

In the case of the speech gesture, the feature extractor will early have to deal alongside the long-term non stationary. For this reason, the speech gesture is normally cut into constructions of concerning 10-30ms and feature extraction is gave on every single piece of the waveform of speech signal. Secondly, the feature extraction algorithm has to cope alongside the short-term redundancy so that decreased and relevant aural data is extracted. For this motive, the representation of the waveform is usually swapped from the temporal area to the frequency area, in that the short-term temporal periodicity is embodied by higher power benefits at the frequency corresponding to the period. Thirdly, the feature extraction ought to flat out probable degradations incurred by the gesture after sent on the contact channel. Finally, feature extraction ought to chart the speech representation into a form that is compatible alongside the association instruments in the remainder of the processing chain. Codebook generation algorithms

1.2 Compression Methods

Subspace allocation clustering and Gaussian tying are two methods utilized to compress speech recognition systems by allocating parameters amid Gaussian distributions. These methods ons*et al*ongside a fully trained HMM set and tie parameters amid the Gaussian constituents of disparate states. Later tying, more training can be gave if necessary. An alternative to clustering afterward training is to early delineate a tiny, fixed set of basis allocations, and next to train a number of interpolation coefficients, but this option is not learned here.

Similar parameter tying methods are frequently utilized across training in order to vanquish the data sparsity problem. For example, decision tree tying of HMM (Suni, Antti Santeri *et al*, 2013*)* states or Gaussian constituents pools the obtainable data and permits a larger parameter estimate. The compression methods discussed below can be utilized in conjunction alongside decision tree tying but, unlike decision tree tying, do not use each specialist phonetic knowledge.

Subspace compression and Gaussian tying on s*et al*ongside a fully trained HMM s*et al*ongside N physical states, every single alongside M Gaussian constituents encompassing of a mean and a variance of dimension D, the dimension of the input feature vector. There are a finished of NM Gaussian components. For an uncompressed ideal alongside diagonal covariance matrices, flouting constituent priors and transition matrices, the finished number of Gaussian parameters is

Number of Parameters = 2NMD

Many of these Gaussian constituents could be comparable to every single supplementary, and the compression methods seize supremacy of this by tying parameters that are close in aural space. Subspace clustering early undertakings every single Gaussian constituent onto K tinier subspaces. In finish, the dimensions of every single of the K tinier subspaces have to add up to D. After K = 1, the method is equivalent to Gaussian tying. After K = D, the method is equivalent to feature-parameter-tying HMMs

For each subspace, all of the subspace-Gaussians from all states of the HMMs are pooled, and clustered to L prototypes, where L << NM. Each original subspace-Gaussian is replaced by its nearest prototype. There is no limitation on the algorithm used for clustering Gaussians, which can be top-down or bottom-up, using any appropriate distance metric. The likelihood of observation ot at time t for nth state sn becomes.

$$P(\mathbf{o}_t|s_n) = \sum_{m=1}^{M} c_{nm} \left(\prod_{k=1}^{K} \mathcal{N}(\mathbf{o}_t|\mu_{nmk}, \Sigma_{nmk}) \right)$$

Subspace compression tends to give better results when there are more subspaces, it allows for clustering to fewer Gaussians with less distortion between the dimensions. Compression can be used together with other approaches, such as Gaussian selection (Rabiner *et al,* 2011*)* for improved efficiency.

There are two portions to the models afterward subspace compression. The early is the pool of prototype Gaussians, and the succeeding is the index from the HMM states to the correct subspace prototypes. The Gaussian pool is plainly the set of subspace Gaussians, as the index corresponds to the links amid the HMM states and the subspace Gaussians. The size of the Gaussian pool depends on the feature vector dimension, D, and the number of prototypes say, L. The size of the index depends on the early number of Gaussian constituents in the arrangement and the number of subspaces, K. The finished number of parameters in the compressed ideal is

#Parameters = 2LD +NMK

In exercise, the size of the index inclines to be colossal contrasted to the size of the Gaussian pool, as L can be moderately tiny lacking discerning degradation in performance.

1.3 Vector quantization

Vector quantization (VQ) is a classical quantization method from signal handling which allows the modeling of probability density functions by the distribution of model vectors. Vector quantization was

originally employed for data compression. It functions by dividing a large ready of factors (vectors) into groups having approximately equivalent number of points nearest to them. Each group is symbolized by its centroid point, as in k-means and several other clustering algorithms.

Vector quantization is dependent on the aggressive learning paradigm, so that it is closely pertaining to the self-organizing map model and to sparse programming models utilized in deep studying formulas such as Autoencoder. Vector quantization is actually used for lossy data compression, lossy data modification, design recognition, occurrence estimation and clustering.

For occurrence evaluation, the area/volume that is closer to a certain centroid than to any additional is actually inversely proportional for the density. The density coordinating property of vector quantization is powerful, specially for identifying the existence of large and high-dimensioned data. Since data factors are represented by the index of the nearest centroid, generally occurring data have low error, and unusual data have high error. This really is why VQ would work for lossy data compression. It may also be used for lossy information correction and density estimate.

Lossy data correction, or forecast, is used to recover data lacking from some dimensions. It is actually completed by locating the nearest group with the data dimensions available, then anticipating the result considering the prices for the missing dimensions, making the hypothesis that they have the same value as the group's centroid.

2. Proposed Work

Speech recognition systems typically contain many distributions patterns. A large number of compression and other parameters are associated with speech data. This makes them slow to decode speech, and large to store. Methods have been proposed to decrease the number of parameters and hence increase compression of digital media.

Large vocabulary speech recognition is a computationally expensive task with models requiring a large amount of parameters to obtain good error rates. As we have discussed in this report that there are various techniques available for Automatic Speech recognition (ASR) namely: Vector quantization, Neural Networks, Dynamic Time Warping, Hidden Markov Models and Genetic Algorithms and others.

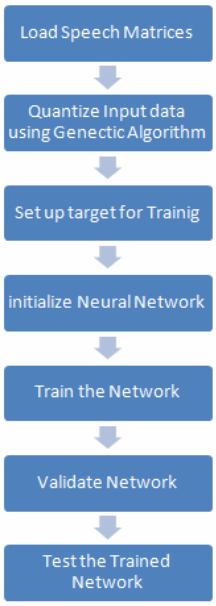
For speech recognition by using neural networks there are few points which are to be considered

a) Grouping of vocabulary
b) Parallel execution of neural networks
c) Enhancement in BPTT algorithm

Enhancement means to enhance performance of speech recognition system using Artificial Neural Network technique by grouping of vocabulary. Speech in neural network requires the huge amount of storage space is not only the consideration but also the data transmission rates for communication of continuous media are also notably large. This type of data transfer rate is not attainable with today's technology, or else in near the future with reasonably priced hardware.

Our objective is to make the voice recognition more efficient by solving memory problem to store voice data. For this purpose we shall use the previous implemented speech algorithms and shall compare the new implemented algorithms for more number of speakers and different mode of languages like aggression, sad, happy and angry.

The main advantage of using Vector Quantization in Pattern Recognition is its low computational burden when compared with other techniques such as Dynamic Time Warping and Hidden Markov Models. The main drawback when compared to Dynamic Time Warping and Hidden Markov Models is that it does not take into account the temporal evolution of the signals because all the vectors are mixed up in the input Signal. The neural network have to face many difficulties during training process due to this large data, so for ease of neural network training we want to reduce the memory space but data should not be lost. We have used **Genetic Algorithm Concept** and **vector quantization method (speech compression technique)** for compression.



Proposed Work flow of GA VQ based Neural network Training

3. Results

Speech recognition systems typically contain many distributions patterns. A large number of compression and other parameters are associated with speech data. This makes them slow to decode speech, and large to store. Various techniques have been proposed to decrease the number of parameters and hence increase compression of digital media.

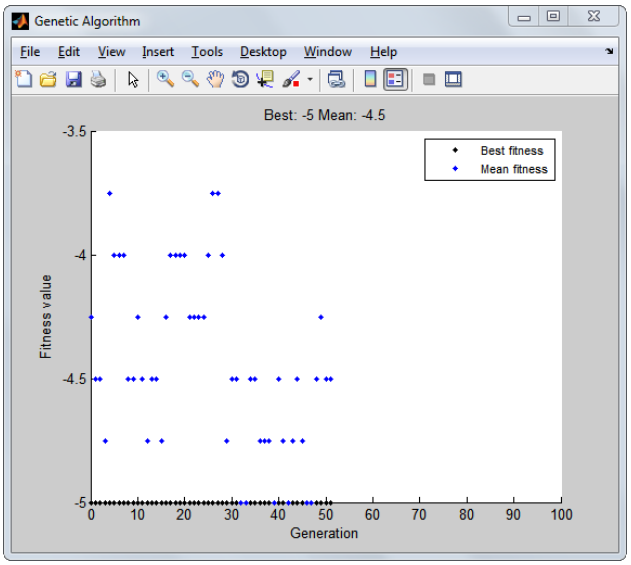


Figure 3 Best Candidate Selection Using Genetic Algorithm

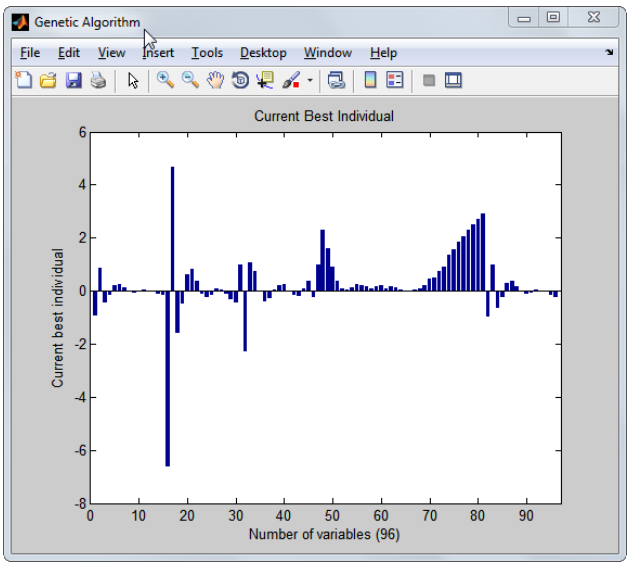


Figure 4 Current Best individual with Genetic Algorithm

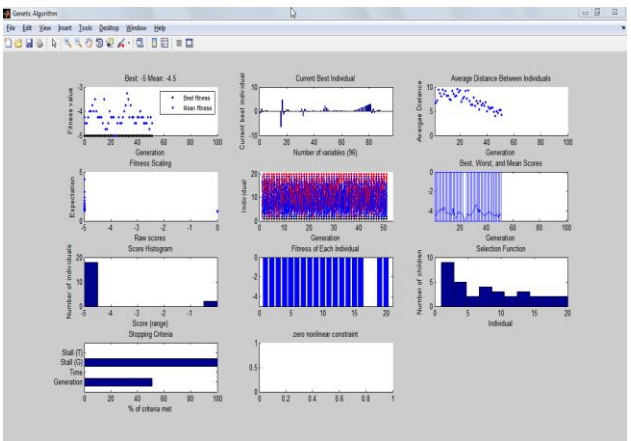


Figure 5 Genetic Algorithm output for Speech recognition

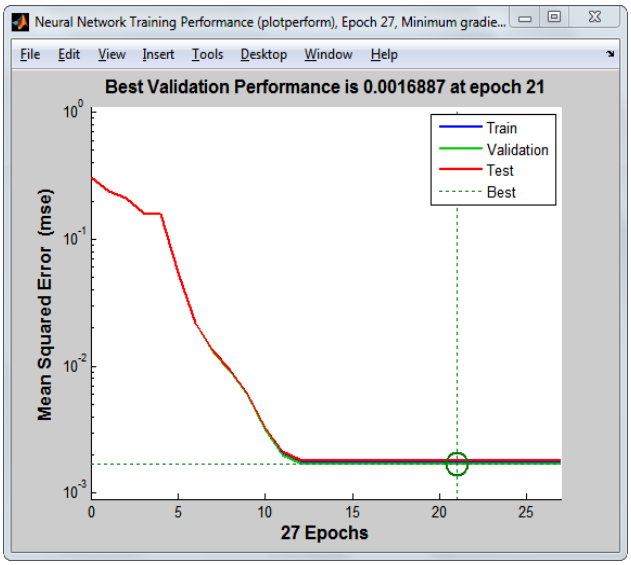


Figure 6 Best Validation performance Neural Network

Large vocabulary speech recognition is a computationally expensive task with models requiring a large amount of parameters to obtain good error rates. As we have discussed in this report that there are various techniques available for Automatic Speech recognition (ASR).

The main advantage of using Vector Quantization in Pattern Recognition is its low computational burden when compared with other techniques such as Dynamic Time Warping and Hidden Markov Models. The main drawback when compared to Dynamic Time Warping and Hidden Markov Models is that it does not take into account the temporal evolution of the signals, because all the vectors are mixed up in the input Signal.

4. Conclusion

In this work we have discussed mainly about Speech recognition and using Genetic algorithms for the same. We have successfully demonstrated that genetic algorithms can be used for the Automatic speech recognition in with more than 79.9% success rate. As we concluded in our results that the system we devised using Genetic algorithms and neural networks produced less error rates in speaker recognition as oppose to using only one method at a time. Feature extraction is the most important part of speech recognition system. Every speech has different individual features. These features can be extracted by using feature extraction techniques proposed and successfully utilized for speech recognition task. The extracted feature should meet some criteria while dealing with the speech signal, Previous methods have focused on ASR using LVQ, MFCC, HMM, and ANN based approaches. In our future works we would like to improve ASR by utilizing Hybrid HMM -VQ based feature selection and classification approach.

References

Frikha, Mondher, and Ahmed Ben Hamida. (2012): A Comparitive Survey of ANN and Hybrid HMM/ANN Architectures for Robust Speech Recognition. *American Journal of Intelligent Systems* 2, no. 1 1-8.

Srinivasan, A. (2012) Speaker identification and Verification using Vector quantization and Mel frequency Cepstral Coefficients. *Engineering and Technology* 4, no. 1: 33-40.

Huang, Yi-bo, Qiu-yu Zhang, and Zhan-ting Yuan. (2014) Perceptual Speech Hashing Authentication Algorithm Based on Linear Prediction Analysis. TELKOMNIKA Indonesian Journal of Electrical Engineering 12, no. 4: 3214-3223.

Suni, Antti Santeri, Daniel Aalto, Tuomo Raitio, Paavo Alku, and Martti Vainio.(2013).Wavelets for intonation modeling in HMM speech synthesis. In 8th ISCA Workshop on Speech Synthesis, Proceedings, Barcelona.

Rabiner, L. R., and R. W. Schafer. (2011) Digital Speech Processing. The Froehlich/Kent Encyclopedia of Telecommunications 6: 237-258.

Haizhou Li Bin Ma, and Kong Aik Lee. (2013) Spoken language recognition: from fundamentals to practice. Proceedings of the IEEE 101, no. 5: 1136-1159.