Speech Recognition using Different Models and their Comparitive Review and Biased Testing Classification

Vinamra Benara(201331007)
Deep Jahan(201364124)
Dhrumil Patel(201364028)
IIIT-Hyderabad

Email: @research.iiit.ac.in

***Abstract*—Isolated Speech Recognition has caught attention of the researchers since 1950s. Many methods/models have been devised since then like Vector Quantization, Dynamic Time Wrapping, etc. Then in 1970s HMM was actively used for the purpose. The shift from Vector Quantization and DTW to HMM was largely due to the fact that former models were not capable of supporting a large vocabulary and were highly erroneous in certain cases. Hidden Markov Model solved this problem by inherently forming acoustic and language models during its training. But HMM could not account for the common features of the human languages. The HMMs dominated the market till 2000s when the methods like Gaussian Mixture Model and Deep Learning, Neural Networks came into picture.In this paper, we aim to implement GMM, HMM, RNN and Vector Quantization and their comparitive performance. Based on this study, we devised some improvisations on the GMM model, to increase the accuracy by at least 4%.**

I. Introduction

The aim of the project is to compare different models for speech recognition. The following have been implemented and compared in this project:

1) Vector Quantization
2) Gaussian Mixture Model
3) Hidden Markov Model
4) Recurrent Neural Network

The training data set includes different types of words including some rhyming words. Then, further we compare the above models based on some biased for some efficient speech recognition.

II. Breif Theory on Different Models

A. Gaussian Mixture Model

GMM is a probabilistic clustering method which does probabilistic soft clustering rather than conventional hard clustering. It is represented as a weighted sum of Gaussian component densities. It is parametric problem where the means ,priors and variances are optimised using the EM algorithm.

$$p(x|\lambda) = \sum_{i=1}^{M} w_i g(x|\mu_i, \Sigma_i) \tag{1}$$

where

$$\lambda = \{w_i, \mu_i, \Sigma_i\} \qquad i = 1, 2....., M \tag{2}$$

Gmm is an approximate method wherein the underlying distribution is assumed to be gaussian. Although this does not cover all the intricacies of the data , but it definitely makes the analysis easier and analytically tractable. They provide a smooth distribution of arbitrary densities. Initially the parameters are assumed to be a random value and then iteratively a local optima is achieved by EM , which is a disadvantage, hence the final result depends on initialization. So to get a global optima, various re-initializations may be required. The EM algorithm is given as:

$$w_i = \frac{1}{T} \sum_{t=1}^{T} Pr(i|x_t, \lambda) \tag{3}$$

$$\mu_i = \frac{\sum_{t=1}^{T} Pr(i|x_t, \lambda) x_t}{\sum_{t=1}^{T} Pr(i|x_t, \lambda)} \tag{4}$$

$$\sigma_i^2 = \frac{\sum_{t=1}^{T} Pr(i|x_t, \lambda) x_t^2}{\sum_{t=1}^{T} Pr(i|x_t, \lambda)} - \mu_i^2 \tag{5}$$

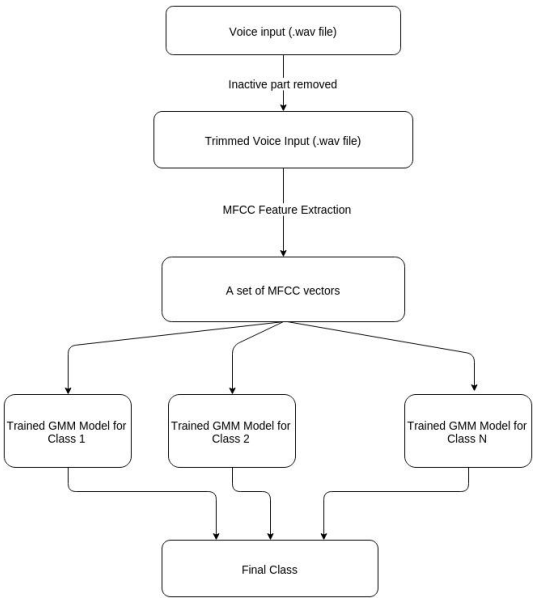
and the posterior is given by

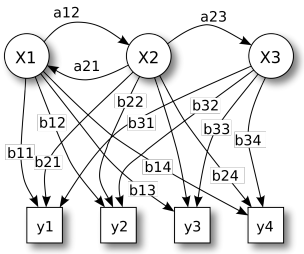
$$Pr(i|x_t, \lambda) = \frac{w_i g(x|\mu_i, \Sigma_i)}{\sum_{k=1}^{M} w_k g(x|\mu_k, \Sigma_k)} \tag{6}$$

This has to be iterated till the parameters does not change much.
The flow chart is shown as:

B. Hidden Markov Model

A hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states.
It models every outcome as a parametric random variable with certain probabilities under different situations. Here:
X = states; (Not visible)
a = transition probabilities;
Y = possible observations; (Visible)
b = output probabilities;





Each word v modeled as distinct HMM Hv Training set of k occurrences per word Each of which is an observation sequence we need to perform the following steps: Estimate parameters for each Hv that maximize $P(O1,,Ok|Hv)$ (i.e. Baum-Welch Algorithm) Extract features $O = (O1,,OT)$ from unknown word Calculate $P(O|Hv)$ for all v (Forward Algorithm), find v which maximizes

C. Vector Quantization

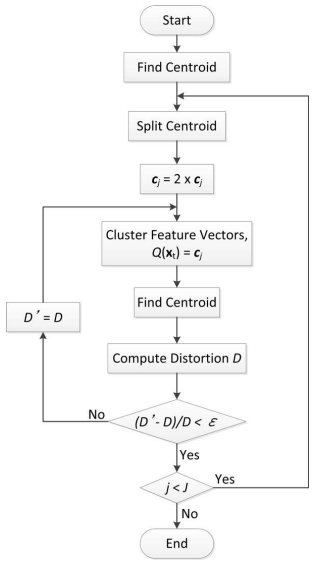
In this method , the data is compressed via lossy technique. A codebook is formed for each data which is a compressed version and is representative of that data. It is based on the principle of block coding. A code-book is a reduced dataset version which can be achieved by many methods such as incremental K-Means Clustering, spectral clustering etc. 1 codebook is generated for each training dataset.

$$X_m - - > C_n \tag{7}$$

where $X_m = \{x_1, x_2....x_m\}$ *and* $C_n = \{c_1, c_2.....c_n\}$ C are the centroids.

Then the optimization criterion is maximised based on a Dmin function where the each of the codebooks is compared with test data and a intermatrix distance is computed.

The iterations are done till The distotion is less than some threshold vlaue.



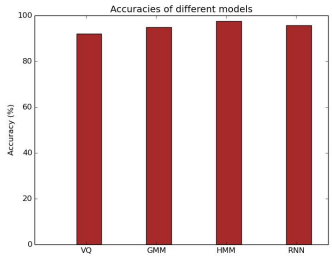
D. Hidden Markov Model

III. Results

We made the data set on our own with approximately 500 data set. Here are the graphs for accuracy of different models v/s different kinds of words.

A. Accuracy for different words

The graph below shows the cumulative accuracy of different models.

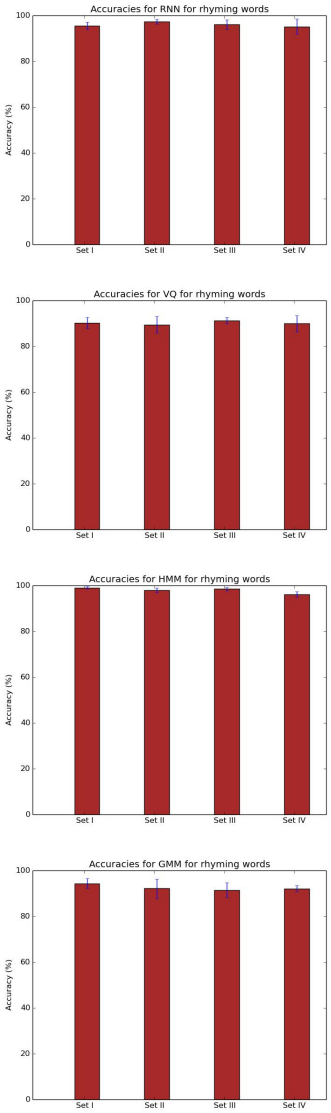


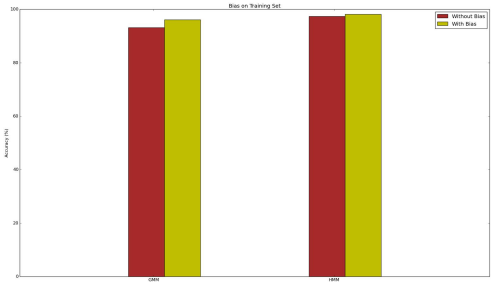
B. Accuracy for Rhyming words

To measure the peculiarities of the classifers we took a set of similar sounding words and measured the accuracy on the classifiers. The word catagories are:
a.)Riddle and similar
b.)Share and similar
c.)God and similar
d.)Around and similar

IV. Novelity

The idea is that given training data from different speakers, how can we improve speech recognition. We know that if we know the speaker, then speech recognition should be better if the training data is from that speaker. So, we recognize the speaker first and then use our models for speech recognition





using biased training data. An illustration for this is: Suppose there are 4 speakers A, B, C, D. And we train the dataset using the model for eg: GMM. But here we take the learning process corresponding to every speaker differently as well as together. So for every word there are five GM models possible corresponding to every speaker. So when we already know that B speaker is going to say the successive words then we can bias the posterior probability of a test sound towards that speaker by some large amount and giving some weights to the remaining speaker posterior probability. And the final posterior probability can be viewed as the linear combination of the different posterior probabilities from GM model from different speaker for a particular word.

REFERENCES

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.