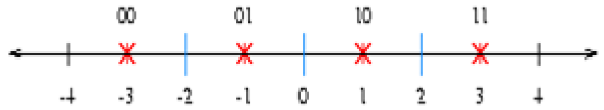
problem due to the need for multi-dimensional integration. In 1980, Linde, Buzo, and Gray (LBG) proposed a VQ design algorithm based on a training sequence. The use of a training sequence bypasses the need for multi-dimensional integration. A VQ that is designed using this algorithm are referred to in the literature as an LBG-VQ.

A VQ is nothing more than an approximator. The idea is similar to that of ``rounding-off" (say to the nearest integer). An example of a 1-dimensional VQ is shown below:



Here, every number less than -2 is approximated by -3. Every number between -2 and 0 are approximated by -1. Every number between 0 and 2 are approximated by +1. Every number greater than 2 is approximated by +3. Note that the approximate values are uniquely represented by 2 bits. This is a 1-dimensional, 2-bit VQ. It has a rate of 2 bits/dimension.

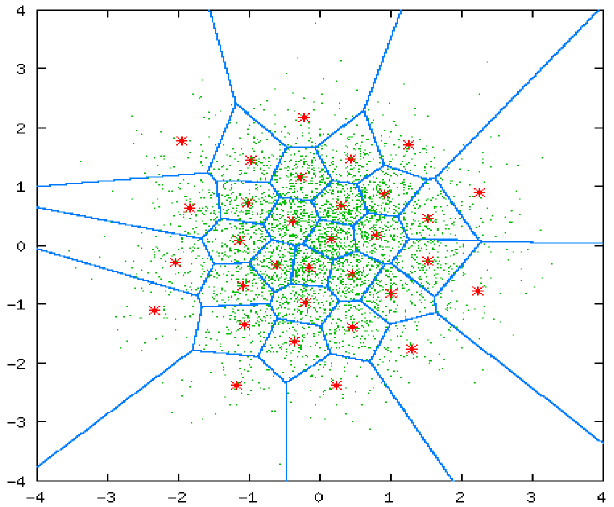


Figure 1: 2-dimensional VQ

In the above two examples, the red stars are called code vectors and the regions defined by the blue borders are called encoding regions. The set of all code vectors is called the codebook and the set of all encoding regions is called the partition of the space.

By using these training data features are clustered to form a codebook for each speaker. In the recognition stage, the data from the tested speaker is compared to the codebook of each speaker and measure the difference. These differences are then use to make the recognition decision.

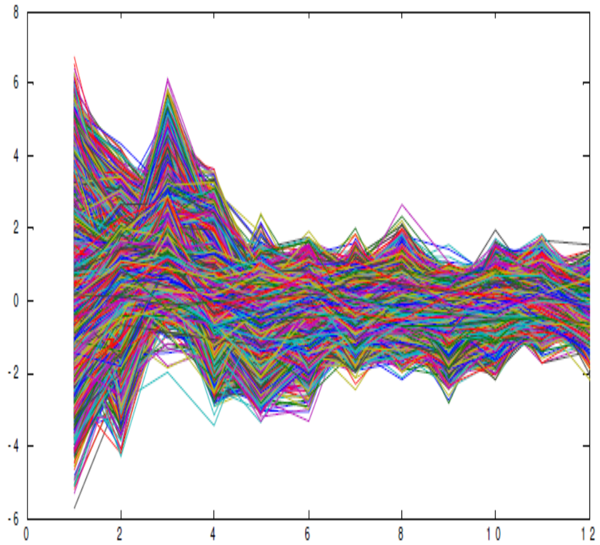


Figure 2: The vectors generated from training speech file before VQ

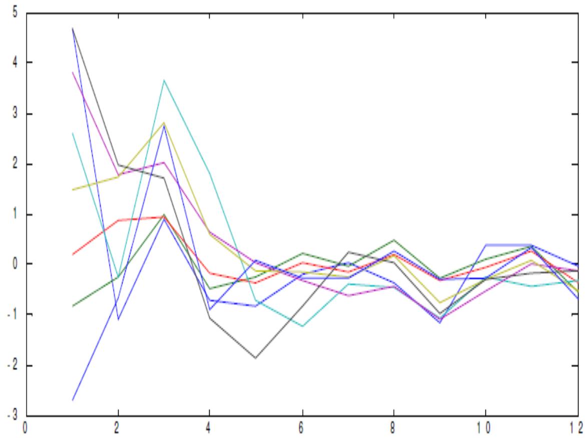


Figure 3: The representative feature vectors from speech file resulted after VQ

3.3 Algorithmic module:

Step1: Read input train file.
Step2: Calculate MFCC for train file.
Step3: Calculate code book for the train file using Vector Quantization with code book size 16.
Step4: Repeat Step 1, 2, 3 for all train files and calculate the code book representing each train file.
Step5: Read input test file.
Step6: Calculate MFCC for test file.
Step7: Calculate code book for the test file using Vector Quantization with code book size 16.
Step8: Calculate the distortion (Euclidean Distance) between the training vector codebook and testing vector.
Step9: Check if the distortion is minimum. If yes go to Step 10 else go to Step 12.
Step10: Print that the test file belong to train file and Update the minimum distortion.
Step 11: Take up the next test file go to Step 5.
Step12: If the distortion is not minimum then take up the next train file code book and repeat Step 8.

3.4 Symbolic Representation based Speaker Identification

Let [D1, D2, D3……Dn] be a set of 'n' training speech files of a class Cj:j=1,2,3….p (p denotes the number of categories) and let Xm={xm,1,xm,2,………xm,k} be k-dimensional code vectors (vector quantized) characterizing the speech file Dn of the class Cj. We have computed the mean and standard deviation of the code vectors in each category. Then we add mean and standard deviation to obtain the maximum interval and we subtracted the mean and standard deviation to obtain the minimum interval. The obtained intervals of all speech files with respect to each category are combined to form a feature vector of length k. This process is repeated for all the speech files present in the class Cj and also for all other trained speech files of all other classes. These minimum and maximum class intervals of each class i.e., interval valued type of class Cj are represented as Cj = { Cj+ , Cj-}.The Cj = { Cj+ , Cj-} represents the upper and lower limits of feature value of a class in the knowledge base.

Now the representative vector for the class Cj, is formed by representing each 'm' feature in the form of an interval and is given by,

Sj={ [fj1- , fj1+],[fj2- , fj2+],
[fj3- , fj3+],………..,[fjm- , fjm+]}

This is a vector of interval-valued features and this symbolic feature vector is stored in the knowledge base as a representative of the jth class. Similarly we compute symbolic feature vectors for all the classes (j = 1, 2, 3 ,…, p) and store them in the knowledge base for classification. Thus, the knowledge base has 'p' number of symbolic vectors each corresponding to a class instead of p × n vectors in case of conventional representation.

Given a test speech, which is described by a set of 'm' feature values that is code vectors derived from the vector quantization compare it with the corresponding interval type feature values of the respective class that is stored in the knowledge base. Let Ft = [ft1 , ft2 , ft3……… ftm] be a 'm' dimensional feature vector describing a test document. Let Sj be the interval valued symbolic feature vector of jth class. Now, each mth feature value of the test speech is compared with the corresponding interval in Sj to examine whether the feature value of the test speech lies within the corresponding interval. The number of features of a test speech, which fall inside the corresponding interval, is defined to be the degree of belongingness. We make use of Belongingness Count Bc as a measure of degree of belongingness for the test speech to decide whether it belongs to the correct class or not

$$B_c = \sum_{m=1}^{k} C\left(f_{tm},\left[f_{jm}^{-}, f_{jm}^{+}\right]\right), where$$

$$C\left(f_{tm},\left[f_{jm}^{-}, f_{jm}^{+}\right]\right) = \begin{cases} 1 & if \left(f_{tm} \geq f_{jm}^{-}, f_{tm} \leq f_{jm}^{-}\right) \\ 0 & otherwise \end{cases}$$

The value of a test speech that falls into its respective feature interval of the reference class contributes a value '1' towards belongingness count and there will be no contribution from other features which fall outside the interval. The time required to classify each test speech is less as we consider interval features for all the train files belonging to each class.

3.5 Algorithmic Module

Step1: Input train file belonging to class Cj: j=1,2,3,…p.
Step 2: Find MFCC feature for each train file in Cj.
Step 3: Calculate code vectors for each train file using Vector Quantization in Cj.
Step 4: Calculate the Mean of code vectors representing each training file of class Cj.
Step 5: Calculate the Standard Deviation of code vectors representing each training file of class Cj.
Step 6: Calculate minimum interval for class Cj by Subtracting standard deviation from mean.
Step 7: Calculate Maximum interval for class Cj by Adding standard deviation to mean.
Step 8: Repeat the Steps 1,2,3,4,5,6,7 for all j=1,2,3,4,…p classes and find out the interval features representing each class Cj.
Step 9: Input test file.
Step 10: Find MFCC feature for test file.
Step 11: Calculate code vectors for test file using Vector Quantization.
Step 12: Calculate the degree of belongingness count Bc for the test code vectors in class Cj for all j=1,2,3,….p.
Step 13: Identify the class Cj with j for which highest belongingness count for the test file is recorded.
Step 14: Output test file belongs to the class Cj with j for which Bc is maximum.

4. Experimental Settings

During experimentation, we conducted three sets of experiments; where each set contain three different trails. In the first set of experiment we used 40% for training and remaining 60% for testing purpose. For the second set of experimentation we used 60% for training and remaining 40% for testing. For third set we used 50% for training and remaining 50% for testing. Each set of experiments contain three different trials. In each trails documents are shuffled between training and testing set.

We have used a database of 29 speakers taken from TIMIT database with 9 samples for each speaker. All the 9 samples are different utterances with different sentences for each speaker. We considered our own dataset of ten speakers also. To evaluate any system we use Precision, Recall and F-Measure as metrics to find the efficiency and robustness of the adopted method.

In order to check the robustness and to study the behavior of the LBG-vector quantization method on different speakers, we have conducted experiments on datasets viz., 29 class dataset, 10 class dataset. We analyzed the results obtained from different datasets. The maximum F-measure values are stated in the table 4.

5. Conclusion

The method works well on dataset 1. In 40% training and 60% testing trail we got highest F-measure for 1st dataset. It shows that the method works well on standard dataset. The two sets of experiments show the efficiency of the methods.

Table 1: Maximum F – Measure table obtained from the method

Datasets	Max F-Measure		
	Training / Testing Ratio		
	40 : 60	60 : 40	50:50
Dataset 1	85.93	85.87	84.87
Dataset 2	60.92	67.07	66.09

To study the behaviour of the speaker identification method using symbolic interval valued representation for speech, extensive experiments were carried out on the dataset 1. The maximum F-measure values obtained for the proposed method are stated in table 5.3. The method works well on the dataset 1. In 60% training and 40% testing trail we got highest F-measure for dataset 1.

A brief introduction to various feature extraction techniques, a study of speaker recognition techniques are addressed in this paper. In addition to this, considering distance as a proximity measure a MFCC and LBG-Vector Quantization method is adopted to classify the speakers. A novel symbolic representation for speech is presented. A technique to use symbolic speech data for speaker recognition is also explored. To check the efficiency and robustness of the proposed models, an extensive experiment is carried out on speech datasets, the details of the results are presented in respective chapter. The result evaluations of all the experiments are carried out by considering precision, recall and F-measure as metrics.

The proposed method is efficient on bench mark dataset and there by indicates that the proposed speaker recognition method is an effective tool for authentication that can be adopted in near future.

6. Future Work

The vector quantized data can be represented in a better way as a tree to make the matching faster. A similar kind of an attempt can also be made on interval valued data representation. Indexing and hashing can be used to improve the results. These methods may be implemented for speech related research.

References

[1] Atal B.S., 1974 “Effective of linear prediction characteristics of speech wave for automatic speaker identification and verification”, J. Acoust. Soc. Am. 55 1304-1312.

[2] Basiri, M. E., Ghasem-Aghaee, N., & Aghdam, M. H.. 2008 “Using ant colony optimization-based selected features for predicting post-synaptic activity in proteins”. EvoLNCS. LNCS (vol. 4973). Heidelberg: Springer-Verlag. pp. 12–23.

[3] Campbell J. P. 1997 “Speaker recognition: A tutorial. Proceedings of IEEE”, 85(9), 1437–1462.

[4] Campbell W, Campbell J, Reynolds D, Singer E, Torres-Carrasquillo, P., 2006. “Support vector machines for speaker and language recognition”. Comput. Speech Lang. 20 (2–3), 210– 29.

[5] Campbell W.M., Sturim D.E., Reynolds D.A, Solomonoff A. 2006 “SVM based speaker verification using a GMM supervector kernel and nap variability compensation”. In: IEEE ICASSP-2006, Toulouse, France, pp. 97–100.

[6] Cheung-chi L., 2004 “GMM-based speaker recognition for mobile embedded systems”. Ph.D. thesis, University of Hong Kong.

[7] Furui.S, 1994 “An overview of speaker recognition technology”, Proceedings of Workshop on Automatic Speaker Recognition, Identification and Verification, Martigny, Switzerland, pp.1-9.

[8] Guru, D.S., Harish, B.S. and Manjunath .S.”Symbolic representation of text documents.” In proceedings of third annual ACM Bangalore conference. 2010.

[9] Kenny P, Boulianne G., Dumouchel P. 2005 “Eigenvoice modeling with sparse training data”. IEEE Trans. Speech Audio Process. 13, 345–354.

[10] Kenny P. Oueleet P. Dehak N. Gupta V. Dumouchel P. 2008 “A study of inter-speaker variability in speaker verification”. IEEE Trans. Audio Speech Lang. Process. 16, 980–988.

[11] Linde Y, Buzo A, & Gray R. M. 1980 “An algorithm for vector quantizer design. IEEE Transactions on Communications”, 28, 84–95.

[12] Reynolds, D. A., & Rose, R. C. 1995. “Robust text-independent speaker identification using Gaussian mixture speaker models”. IEEE Transactions on Speech and Audio Processing, 3(1), 2–83.

[13] Reynolds. D, Quatieri T., Dunn R., 2000 “Speaker verification using adapted gaussian mixture models”. Digital Signal Process. 10 (1), 19–41.

[14] Solomonoff, A., Campbell, W.M., Boardman, I., 2005. Advances in channel compensation for SVM speaker recognition. In: IEEE ICASSP-2005, Philadelphia, PA, pp. 629–632.

[15] Voiers .W D, "Perceptual Bases of Speaker Identity," J Acoust. Soc. Am. 36, 1065.

Author Profile

Akshay S received M.S. degree in Computer Science from University of Mysore in 2012. From 2012 he is working as a lecturer in Department of Computer Science, Amrita Vishwa Vidyapeetham, Mysore Campus. His areas of interests are Pattern Recognition, Digital Signal Processing, Image Processing, Algorithms and Data structures.

Apoorva P received M.Sc. degree in Computer Science from University of Mysore in 2011. From 2011 she is working as a lecturer in Department of Computer Science, Amrita Vishwa Vidyapeetham, Mysore Campus. Her areas of interests are Pattern Recognition, Digital Signal Processing, Computer networks and network security.