VECTOR QUANTIZATION TECHNIQUES FOR GMM BASED SPEAKER VERIFICATION

Gurmeet Singh[1], Ashish Panda[2], Saurav Bhattacharyya[2], Thambipillai Srikanthan[2]

[1]Indian Institute of Technology, Kanpur. India.

[2]Centre for High Performance Embedded Systems,
School of Computer Engineering, Nanyang Technological University, Singapore.

[1]gurmeet@iitk.ac.in, { PA8288972,assaurav,astsrikan}@ntu.edu.sg

ABSTRACT

This paper explores the novel application of two Vector Quantization algorithms, namely *Linde, Buzo, Gray* (LBG) and *K*-means algorithm for efficient speaker verification. Automatic Speaker Verification (ASV) is a memory and compute intensive process, giving rise to area and latency concerns in the way of its implementation for real-time efficient embedded systems. The training schemes for computing the speaker models, such as the expectation maximization are highly iterative and contribute significantly to the overall complexity in the implementation of the system. In this paper, we demonstrate the use of the LBG and the *K*-means algorithm to realize compute efficient training method. Models trained with the *LBG* algorithm achieves as much as 99.88% of EM accuracy, whilst *K-means* achieves as much as 99.91% of EM accuracy. Moreover, the EM computational complexity is almost twice that of *LBG* or *K*-means. Thus, using LBG and *K*-means algorithms for training Gaussian mixture speaker models for text-independent speaker verification, we show that, that they deliver comparable performance as the EM algorithm at significantly reduced computational complexity. Thus making them an ideal choice for low-cost applications.

1. INTRODUCTION

ASV is a biometric based identity verification process, where personal identity is verified by the voice of a person. Biometrics based identification and verification processes have received much attention in recent times as such characteristics come natural to each individual and they are not required to be memorized, unlike passwords and personal identification numbers. Further, in text-independent SV, the subject is not constrained by a prompted text; rather, the candidate is free to speak any text.

1.1. Major Steps in ASV

There are three important steps in text-independent SV. Feature vector extraction is the first step, where vectors representing the speaker distinguishing characteristics are isolated. The second step is to find a model of a particular speaker. In stochastic modeling, this translates to finding the distribution of the feature vectors. The third and final step is verification. This is the decision step, which determines whether the test utterance is from the claimed speaker. Verification step emulates statistical hypothesis testing and uses likelihood ratio test for decision-making.

2. Speaker Modeling

This section shall discuss the EM algorithm as one of the popular and iterative algorithm for ASV. Further, we shall introduce the adaptation of two popular VQ algorithms: K-Means and LBG, as speaker modeling algorithms.

2.1 EM Algorithm

Gaussian Mixture Model (GMM) approach to text-independent SV [1] has gained widespread popularity in recent years. This is due to the fact that Gaussian mixture modeling is a powerful tool for representing virtually any distribution. Although, the expression of GMM is simple, the training of a GMM, i.e., finding a model given the feature vectors, is rather complex and time consuming due to its computational complexity and iterative nature. Training of a GMM is generally accomplished by the EM algorithm [2], which guarantees convergence to a local maximum. However, the high computational complexity of the algorithm necessitates high hardware cost as well as large training time. In this paper, we have proposed and experimented speaker modeling with LBG [3] and contrasted it with the popular EM algorithm training process.

Although EM algorithm guarantees convergence to a local maximum, it has the following disadvantages:

(a) The computational complexity of EM algorithm is very high. EM algorithm involves many computationally intensive operations like square root, exponentiation and division. The number of such operations required grows exponentially with the number of training vectors and linearly with the number of iterations. It is clear that the high-speed single-chip implementation of EM algorithm would be expensive.

(b) EM algorithm is iterative. It usually takes 10 iterations for EM algorithm to converge. The iterative nature and the complex operations involved are the cause of the large training time reported in [5].

(c) EM algorithm requires an initial model. In order to estimate the initial model a separate algorithm is required. Implementation of the initial model estimation algorithm adds to the cost of the training module.

Hence applicability of alternative training algorithms is worth investigating. The *K*-means algorithm has been applied for finding a robust model approximation to the GMM in [6]. We investigate the applicability of two popular vector quantization algorithms namely *K*-means and LBG algorithm for training speaker models. Subsequently, we compare the performance as well as the complexity of these two algorithms with the EM algorithm.

2.2. K-Means Algorithm

K-means algorithm [4] was originally designed for vector quantization codebook generation. It is an unsupervised clustering algorithm, which represents each cluster by the mean of the cluster. Assuming a set of vectors $X = \{\vec{x}_1, \vec{x}_2, ..., \vec{x}_T\}$ is to be divided into M clusters represented by their mean vectors $\{\vec{\mu}_1, \vec{\mu}_2, ..., \vec{\mu}_M\}$, the objective of the K-means algorithm is to minimize the total distortion given by

$$total\ distortion = \sum_{i=1}^{M}\sum_{t=1}^{T}\left\|\vec{x}_t - \vec{\mu}_i\right\| \qquad (1)$$

K-means follows an iterative approach to meet the objective. In each successive iteration, it redistributes the vectors in order to minimize the distortion. Although originally meant for codebook generation, it can be adapted to train GMM. The procedure is outlined below:

(a) To initialize, M random vector from the training set are selected as the means of M clusters.

(b) Each vector $\vec{x}_t$, $1 \le t \le T$, is assigned to cluster j, iff, $\left\|\vec{x}_t - \vec{\mu}_j\right\| < \left\|\vec{x}_t - \vec{\mu}_k\right\|$, $\forall k \ne j$, $1 \le j,k \le M$.

(c) The new mean of a cluster is obtained by calculating the mean of all the vectors assigned to that particular cluster.

(d) The weights are determined by calculating the proportion of the vectors assigned to the cluster and the covariance matrix is the covariance matrix of the assigned vectors.

Steps (b) and (c) are repeated till the clusters are stable, i.e., the distortion is minimized. When the distortion is minimized, redistribution does not result in any movement of vectors among the clusters. This could be used as an indicator to terminate the algorithm. The total distortion can also be used as an indicator of convergence of the algorithm. Upon convergence, the total distortion does not change as a result of redistribution. When the clusters are stable, the weight and covariance matrix can be found out as described in step IV. It is to be noted that in each iteration, *K*-means estimates the means of all the M clusters.

2.3. LBG Algorithm

LBG algorithm shares many of the characteristics of K-means algorithm. Like K-means, it too was developed originally for vector quantization purpose. The objective of LBG algorithm, like K-means, is to minimize the total distortion as given in Equation (1). However, unlike K-Means, LBG does not estimate the means of all M clusters in each iteration. Rather, it starts with a single cluster and arrives at M clusters by splitting it. With each successive iteration, the number of clusters is doubled. Thus the final number of clusters M could only be a power of 2. The steps in training a GMM with the LBG algorithm, using the same notations as used for K-means, could be described as below:

(a) The vectors of the training set $\{\vec{x}_1, \vec{x}_2, ..., \vec{x}_T\}$ are considered as belonging to a single cluster. The number of present clusters n is set to 1. The mean of the cluster is calculated as $\vec{\mu}_1^1 = \frac{1}{T}\sum_{t=1}^{T}\vec{x}_T$.

(b) Let $\vec{\varepsilon}$ be a vector with small magnitude. The number of clusters is then doubled by splitting the mean $\vec{\mu}_1^1$ into two: $\vec{\mu}_1^2 = \vec{\mu}_1^1 + \vec{\varepsilon}$ and $\vec{\mu}_2^2 = \vec{\mu}_1^1 - \vec{\varepsilon}$. The vectors are redistributed between these two clusters, represented by their means, and the two means $\vec{\mu}_1^2$ and $\vec{\mu}_2^2$ are re-estimated by calculating

the means of the vectors assigned to the respective clusters.

(c) Each of the clusters, thus obtained, is split as described in step (b) and means are re-estimated. This procedure is repeated till the number of present clusters $n = M$.

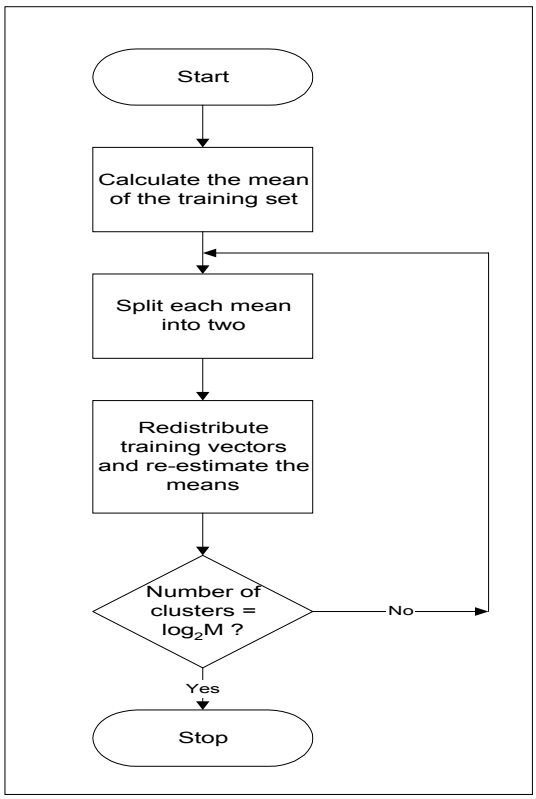

Figure 1: The LBG algorithm

Note that the splitting needs to be done $\log_2 M$ times in order to obtain M clusters. After the clusters are stable, i.e., the number of desired clusters is reached, the weights can be found out by calculating the proportion of the number of vectors assigned to a particular cluster. The covariance matrix can be obtained by computing the covariance matrix of the assigned vectors. The flow diagram of the LBG algorithm is illustrated in Figure 1.

3. Experimental Results and Discussion

Experiments were conducted on KING and TIMIT databases to investigate the suitability of *K*-means and LBG algorithm for training a GMM. In this section the results of the experiments are presented and a cost-performance analysis is given.

Experimental Set-up: Silence frames were removed by setting a adaptive amplitude threshold [7] and 20 mel-cepstrums were calculated from a 20 ms window progressing at 10 ms. Models were trained using *K*-means, LBG and EM algorithms. The training utterance duration was 1 minute for KING database and 24 seconds (approx) for TIMIT database. Training speech was collected from the first three sessions of wideband speech of KING database and the first 8 sessions of test portion of the TIMIT database. The test utterance duration was 10 seconds for KING database and 3 seconds in case of TIMIT database. A Global Background Model (GBM) GBM [8] with 32 components trained on 20 minutes of speech and a GBM with 64 components trained on 8 minutes of speech were used for KING and TIMIT databases respectively.

Experimental Results: The EER obtained from *K*-means and LBG training schemes are compared against the EER from EM algorithm training scheme in Tables 1 and 2. The complexity analysis for one iteration of the three schemes is shown in Table 3. Note that EM algorithm will first be required to be initialized by the result of approximately 10 iterations of *K*-means algorithm.

From the Tables 1 and 2, it can be readily observed that *LBG* achieves 99.39% and 99.88% of EM accuracy for KING and TIMIT databases. Meanwhile, *K*-means achieves 99.39% and 99.91% of EM accuracy for KING and TIMIT databases. NOTE : accuracy = 100% - EER%.

Training Algorithm	Components		Utterance Duration		EER
	Claimant	Background	Training (min)	Test (sec)	%
EM	32	32	1	10	1.4
K-Means	32	32	1	10	2.0
LBG	32	32	1	10	2.0

Table 1: Performance comparison of training schemes on KING database

Training Algorithm	Components		Utterance Duration		EER
	Claimant	Background Model	Training (sec)	Test (sec)	%
EM	32	64	24	3	0.39
K-Means	32	64	24	3	0.47
LBG	32	64	24	3	0.50

Table 2: Performance comparison of training schemes on TIMIT database.

It can be observed from Table 1 and 2 that EM algorithm is more accurate than both K-means and LBG algorithms. The complexities of these algorithms are shown in Table

3, where D, M and T denote the number of dimensions of each training vector, number of components and number of training vectors respectively. Note that *n* in the fourth column of Table 3 denotes the number of clusters in a particular step and there will be an overhead of (MD+D-1)T+M additions/subtractions, MDT multiplication and 2M divisions for weight and variance calculation in case of K-means and LBG algorithm. It is obvious from the table that the complexity of EM algorithm much higher than that of both the algorithms. For example, to train a GMM with 32 components from 6000 vectors of 20 dimensions each, the complexity of EM algorithm would be nearly twice the complexity of K-means and LBG algorithms. Hence, for low-cost and low-security applications EM algorithm could be replaced by either *K*-means or LBG algorithm. But for high security applications, EM algorithm would be preferable.

4. Conclusion

In this paper, we applied two popular vector quantization algorithms, *LBG* and *K-means* algorithms for training the Gaussian mixture speaker models. Experimental results revealed *LBG* achieves 99.39% and 99.88% of EM accuracy for KING and TIMIT databases, meanwhile, *K*-means achieves 99.39% and 99.91% of EM accuracy for KING and TIMIT databases. Moreover, the complexity of K-means and LBG algorithm is almost half that of the EM algorithm, thus, justifying their use for training purpose in a low-cost and low-security application.

Operations per Iteration	EM Algorithm	K-Means Algorithm	LBG Algorithm
Addition/ Subtraction	(4MD+M)T	(2MD-M+D+1)T	D[(2n+1)T+n-1]-T(n-1)+1
Multiplication	(4MD+2M)T +3MD+M	MDT	nDT
Division	2MT+MD+M+1	MD	nD
Comparison	-	(M-1)T	(n-1)T
Exponentiation	MT	-	-
Square-root	M	-	-

Table 3: Complexity comparison of training schemes

REFERENCES

[1] D.A. Reynolds and R.C. Rose, "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models", *IEEE Trans. Speech and Audio Processing*, vol. 3, no. 1, 1995, pp. 72-82.

[2] B.S. Everitt and D.J. Hand, *Finite Mixture Distributions*, Chapman and Hall, New York, 1981.

[3] Y. Linde, A. Buzo and R.M. Gray, "An Algorithm for Vector Quantizer Design", *IEEE Trans. Comm.*, vol. COM 28, Jan. 1980, pp. 84 – 95.

[4] S. Furui, *Digital Speech Processing, Synthesis and Recognition*, Marcel Dekker Inc., New York, 1989.

[5] D.A. Reynolds, R.C. Rose and J.T. Smith, "PC Based TMS320C30 Implementation of Gaussian Mixture Model Text-Independent Speaker Recognition System", *In Proceedings of Int. Conf. On Signal Processing Applications and Technology*, Nov. 1992, pp. 967-973.

[6] J. Pelecanos, S. Myers, S. Sridharan and V. Chandran, "Vector Quantization based Gaussian Modelling for Speaker Verification", *In Proceedings of 15th. Int. Conf. On Pattern Recognition 2000, Vol. 3, pp. 294 – 297.*

[7] C.K. Gan and R.W. Donaldson, "Adaptive Silence Deletion for Speech Storage and Voice Mail Applications", *IEEE Trans .Acoustics, Speech and Signal Processing*, vol. 36, no. 6, 1988, pp. 924-927.

[8] A. Panda, S. Bhattacharyya and T. Srikanthan, "Global Background Model Approach for Embedded Speaker Verification Systems", *In Proceedings of Int. Symposium on Communication Systems, Networks and Digital Signal Proessing,* 2002, pp. 383-386.