International Conference on
Information, Communications and Signal Processing
ICICS '97
Singapore, 9-12 September 1997

Vector Quantization using Genetic K-Means Algorithm for Image Compression

K Krishna K R Ramakrishnan M A L Thathachar
Department of Electrical Engineering
Indian Institute of Science
Bangalore 560012 INDIA
Email: [kkrishna,krr,malt]@ee.iisc.ernet.in

1 Introduction and Motivation

In Vector Quantization (VQ), minimization of Mean Square Error (MSE) between code book vectors and training vectors is a non-linear problem. Traditional LBG type of algorithms converge to a local minimum, which depends on the initial code book. While most of the efforts in VQ have been directed towards designing efficient search algorithms for code book, little has been done in evolving a procedure to obtain an optimum code book. This paper addresses the problem of designing a globally optimum code book using Genetic Algorithms (GAs). GAs have been applied to many function optimization problems and have been shown to be good in finding optimal and near optimal solutions.

GAs work on a coding of the parameter set over which the search has to be performed, rather than the parameters themselves. These encoded parameters are called *solutions* or *chromosomes* and the objective function value at a solution is the objective function value at the corresponding parameters. GAs solve optimization problems using a population of a fixed number solutions. A solution is a string of symbols. GAs evolve over *generations*. During each generation, they produce a new population from the current population by applying genetic operators *viz., natural selection, crossover, and mutation.* A solution consists of a string of symbols, typically binary symbols. Each solution in the population is associated with a figure of merit (fitness value) depending on the value of the function to be optimized. The selection operator selects a solution from the current population for the next population with probability proportional to its fitness value. Crossover operates on two solution strings and results in another two stings. Typical crossover operator exchange the segments of selected stings across a crossover point with a probability. The mutation operator toggles each position in a string with a probability, called *Mutation probability*. For a detail study on GA, readers are referred to [2].

There have been some attempts to use Genetic Algorithms (GAs) in finding the optimal code book vectors that minimizes MSE. These algorithms face the following problems in terms of computational efforts. In the algorithms where the representation of chromosome is such that it favors easy crossover, the fitness evaluation is very expensive. In the algorithms where the fitness evaluation is simple, either the crossover operation is complicated or it needs to be repeatedly applied on chromosomes to obtain legal strings. As observed by Davis[1], GAs are almost never the best algorithms to use for any problem and he suggests hybridization of GAs with the existing algorithms for the problem under consideration. Krishna *et. al.,* [3] have proposed a hybrid GA called Genetic K-Means Algorithm (GKA) that combines the advantages of gradient descent algorithms and GAs. In this algorithm, Gradient descent part helps to speed up the algorithm whereas GA features help to obtain a global optimum. Standard K-Means algorithm is used instead of crossover and a distance based mutation is defined specifically for this problem to effectively pur-

0-7803-3676-3/97/$10.00 © 1997 IEEE

turb the solutions. GKA has been proved to converge to a global optimum with probability one. In this paper, we have applied GKA to VQ for image compression. Since most of the code book design algorithms are based on LBG[4] we have compared the performance of GKA with that of LBG algorithm.

2 Genetic K-Means Algorithm (GKA)

Finding a set of optimal code vectors is equivalent to finding an optimal partition of the training set. In this case, the code book vectors are the centroids of different cells in the partition. In this algorithm we find a partition that minimizes the MSE with respective these centroids.

Let $\{\boldsymbol{x}_i, i = 1, 2, \ldots, n\}$ be the set of n training vectors of d dimension each. Let x_{ij} denote jth feature of $\boldsymbol{x}_i$. Define for $i = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, K$,

$$w_{ik} = \begin{cases} 1 & \text{if } i\text{th vector belongs to } k\text{th cluster,} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Let the centroid of the kth cluster be $c_k = (c_{k1}, c_{k2}, \ldots, c_{kd})$, then

$$c_{kj} = \frac{\sum_{i=1}^{n} w_{ik} x_{ij}}{\sum_{i=1}^{n} w_{ik}}. \tag{2}$$

Then MSE associated with $\boldsymbol{W} = [w_{ij}]$ is given by

$$\mathcal{E}(\boldsymbol{W}) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i=1}^{n} w_{ik} \sum_{j=1}^{d} (x_{ij} - c_{kj})^2.$$

The objective is to find a $\boldsymbol{W}^*$ that minimizes $\mathcal{E}(\boldsymbol{W})$.

GKA uses a string of n symbols where, each symbol takes values from $\{1, 2, \ldots, K\}$. This coding uniquely represents $\boldsymbol{W}$ defined in (1). Because the problem under consideration is minimization and the fitness value of a string should be more if MSE of the associated partition is less. σ-truncation mechanism is used to do this [2]. To mutate symbols in a string, Distance Based Mutation (DBM) is used. Mutating a symbol amounts to changing the partition number of the associated pattern. Let $d_j = d(x_i, c_j)$ be the Euclidean distance between $\boldsymbol{x}_i$ and $\boldsymbol{c}_j$. Then probability of the ith symbol taking a value j is given by

$$\frac{c_m \, d_{max} - d_j}{\sum_{i=1}^{K} (c_m \, d_{max} - d_i)}$$

where c_m is a constant usually ≥ 1 and $d_{max} = \max_j \{d_j\}$. Instead of crossover operator, K-Means operator is used. K-Means operator is a one-step K-Means algorithm. Each solution in the initial population is initialized by assigning p, the greatest integer less than n/K, randomly chosen symbols to each partition and the rest of the symbols to randomly chosen partitions. This is done to avoid the illegal strings, the strings representing less than K partitions. After initialization the subsequent populations are obtained by the application of selection, mutation and K-means operators over the previous populations. The algorithm is terminated when the limit on the number of generations is exceeded. The output of the algorithm is the best solution encountered during the evolution of the algorithm.

3 Results

Initial experiments were conducted on two standard gray level images. These images are of size 128×128. We consider 4×4 blocks for VQ so, for each image we have 1024 training vectors. All the MSE values reported here are computed over the training set.

Since GKA is a stochastic algorithm, the code book vectors it generates in different trials may be different. So, we give the best and average MSE values of the output strings of 5 different runs of GKA. We consider population of size 20 and mutation probability of 0.01 in the results reported here. We compared these results with those obtained using the standard LBG algorithm. Since LBG algorithm is dependent on the initial code book, we ran LBG with 20 initial conditions and reported the best and average MSE values of these runs. The results are tabulated in Table 1.

In the simulations, LBG converged to the reported values within 70 iterations in all the cases. The difference between the best and the average values obtained by LBG is non-zero. This implies that LBG is not assured to reach the best MSE quoted in the table always. Whereas, in case of GKA the averages are converging to the best values always. This implies that GKA converges to the optimum in almost all the runs, which is in concurrence with the convergence result presented in [3]. The results reported

Code book size	MSE Best (Average)					
	Image 1			Image 2		
	LBG	GKA	% Improvement	LBG	GKA	% Improvement
32	172.24 (181.00)	164.07 (165.45)	5 (9)	102.28 (106.36)	96.83 (97.13)	5 (9)
64	139.58 (145.16)	118.08 (118.59)	15 (18)	70.96 (75.31)	62.59 (62.97)	12 (17)
128	105.53 (111.22)	76.36 (76.89)	28 (31)	50.26 (51.89)	37.36 (37.48)	26 (28)
256	58.53 (64.20)	39.74 (39.99)	32 (37)	26.48 (30.14)	16.71 (16.81)	37 (44)

Table 1: Best and average MSE values obtained by applying LBG algorithm and GKA over two images for different sizes of code book.

under GKA are the values obtained after 300 iterations.

As the code book size increases the search space increases exponentially in VQ. Therefore for larger code book sizes, the locally convergent algorithms could find very bad solutions. This is observed in our simulation studies too. As can be seen from the above table for the code book size of 256, the percentage improvement of GKA over LBG is as high as 37 for the best MSE and 44 for the average MSE.

It was observed in the simulations that at every iteration the average and the best values obtained by GKA are less than those obtained by LBG in all the cases presented above. This implies that for the same amount of computational effort GKA gives a better solution than LBG. All these results are encouraging us to try apply GKA in speech coding too.

References

[1] L. Davis, editor. *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York, 1991.

[2] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, Reading, MA, 1989.

[3] K. Krishna and M. N. Murty. Genetic K-means Algorithms. Communicated to IEEE Transactions on System, Man and Cybernetics.

[4] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, COM-28:84–95, January 1980.