A Tabu Search Based Fuzzy-means Algorithm for VQ Codebook Design

Zhe-Ming Lu[1], Jeng-Shyang Pan[2] and Sheng-He Sun[1]

1.Department of Automatic Test and Control, Harbin Institute of Technology, Harbin, China; 2.Department of Electronic Engineering, Kaohsiung Institute of Technology, Taiwan China

P.O.Box 339, Harbin Institute of Technology, Email:zhemingl@yahoo.com

Abstract: The fuzzy c-means algorithm (FCM) has been presented for vector quantization (VQ) codebook design. In this paper, we present a tabu search (TS) based fuzzy c-means algorithm for VQ codebook design. Two key problems of the tabu search algorithm are how to define a solution and how to generate neighboring solutions from the current solution. A codebook-based method is presented to describe a solution for tabu search approach, and the corresponding codebook design algorithm is called CB-TSFCM. Test results show that the CB-TSFCM generates better codebooks than the traditional FCM algorithm and the c-means algorithm.

Keywords: Tabu search, Codebook design, c-means, Fuzzy c-means, Vector quantization

1. Introduction

Vector quantization (VQ) has been intensively investigated and successfully used in speech coding and image compression [1][2]. Codebook design is the key problem of VQ, which is usually based on the minimization of an average distortion measure between the training vectors and the codewords. The minimization of the distortion measure is widely performed by a gradient descent based iterative algorithm that is known as c-means algorithm or the Generalized Lloyd algorithm [2]. Although the c-means algorithm is simple and intuitively appealing, it strongly depends on the selection of the initial codebook, and it can easily be trapped into the local minimum. The search for a globally optimum codebook was attempted in recent years by minimizing the average distortion measure using some global optimum techniques. Zeger et.al. proposed a codebook design technique that combines simulated annealing and the Lloyd iteration[3]. Although simulated annealing generally produces better codebooks than the c-means algorithm, it is a very time consuming process. Genetic algorithm (GA)[4] and stochastic relaxation (SR) approach [5] have also been proposed to codebook design.

The techniques mentioned above are all based on crisp decisions in the sense that each training vector is only assigned to a single cluster according to some criterion. A common ingredient of all these techniques is that they assign each training vector to a single cluster and ignore the possibility that this training vector may also belong to other clusters. Fuzzy clustering algorithms [6] consider each cluster as a fuzzy set, while a membership function measures the possibility that each training vector belongs to a cluster. As a result, each training vector may be assigned to multiple clusters with some degree of certainty measured by the membership function. Thus the partition of the training vector space is based on soft decisions. These algorithms, however, are local search techniques that search for the optimum by using hill-climbing procedures.

The tabu search **(TS)** approach was first proposed by Glover [7]. It is a global optimization technique with short-term memory, it can be used to

0-7803-5747-7/00/$10.00©2000 IEEE.

solve a lot of hard combinatorial optimization problems. In this paper, a new tabu search based fuzzy c-means algorithm is presented for the codebook generation problem.

2. Background

Let us consider training vectors in a k-dimensional Euclidean space and denote the number of vectors in the training set by M. Vector quantization partitions the input space into c non-overlapping regions called *partitions* or *clusters* so that the input space is completely covered. A representative *codeword* is assigned to each partition. The aim of the codebook construction is to minimize the average distance (*distortion*) between the training vectors and their representatives. We denote the training set as $X=\{X_1,X_2,\ldots,X_M\}$ and the codebook as $Y=\{Y_1, Y_2,\ldots,Y_c\}$. The distance between the training vector $X_i=(x_{i1}, x_{i2},\ldots,x_{ik})$ and the codeword $Y_j=(y_{j1}, y_{j2},\ldots,y_{jk})$ is defined by

$$d(X_i,Y_j)=\sum_{l=1}^{k}(x_{il}-y_{jl})^2 \qquad (1)$$

The average distortion of the codebook is calculated by *mean squared error* (*MSE*):

$$MSE=\frac{1}{M}\sum_{i=1}^{M}d(X_i,f(X_i)) \qquad (2)$$

where $f(X_i)$ is a mapping function from training vector X_i to its representative codeword.

2.1 c-means algorithm

The c-means algorithm assigns each training vector to a certain cluster on the basis of the nearest neighbor condition. According to this strategy, the training vector X_i is assigned to the jth cluster if $d(X_i,Y_j)=d_{\min}(X_i)=\min_{Y_j\in Y}(X_i,Y_j)$. The nearest neighbor condition can be conveniently described by a membership function, which is defined as

$$u_{ji}=\begin{cases}1 & if\ d(X_i,Y_j)=d_{\min}(X_i)\\ 0 & otherwise\end{cases} \qquad (3)$$

The codebook vectors are evaluated by minimizing a distortion measure defined as

$$J_1=\sum_{j=1}^{c}\sum_{i=1}^{M}u_{ji}d(X_i,Y_j) \qquad (4)$$

For a given set of membership functions, the minimization of $J_1=J_1(Y_j, j=1,2,\ldots,c)$ with respect to Y_j result in

$$Y_j=\frac{\sum_{i=1}^{M}u_{ji}X_i}{\sum_{i=1}^{M}u_{ji}} \qquad \forall j=1,2,\ldots,c \qquad (5)$$

The codebook vector Y_j defined in (5) is the Euclidean center of gravity or centroid of all the training vectors assigned to the jth cluster.

2.2 Fuzzy c-means algorithm

The c-means algorithm assigns each training vector to a single cluster on the basis of the nearest neighbor condition. The training vector assignment is based in this case on hard or crisp decisions. Fuzzy c-means algorithms assign to each element of the training set a membership value between zero and one that indicates to what extent the particular vector is to be regarded as belonging to a certain cluster. In this way, a fuzzy partition of the training vectors specifies the degree of membership of each vector in each of the c clusters. A fuzzy membership matrix can be used to represent the fuzzy partition. The set of all possible $c\times M$ fuzzy partition matrices is denoted by M_{fcM} and is defined as

$$M_{fcM}=\{U\in R^{c\times M}\,|\sum_{j=1}^{c}u_{ji}=1,\ 0<\sum_{i=1}^{M}u_{ji}<M \text{ and } u_{ii}$$

$$\in[0,1];\ 1\ \ j\ \ c, 1\ \ i\ \ M\} \qquad (6)$$

Given a method for creating fuzzy clusters, the problem is to find the corresponding best fuzzy partition matrix in M_{fcM}.

The derivation of the fuzzy c-means algorithms was based on the constrained minimization of the objective function

$$J_m(U,Y)=\sum_{j=1}^{c}\sum_{i=1}^{M}u_{ji}^{m}d(X_i,Y_j) \tag{7}$$

where $U\in R^{c\times M}$ is a fuzzy membership matrix to describe the membership functions u_{ij}, i.e. $U=(u_{ij})_{c\times M}$, $m>1$. For a given codebook Y, the minimization of $J_m=J_m(U,Y)$ results in the new fuzzy partition matrix U whose element u_{ij} has the form

$$u_{ji}=\frac{1}{\sum_{l=1}^{c}\left[\frac{d(X_i,Y_j)}{d(X_i,Y_l)}\right]^{\frac{1}{m-1}}} \tag{8}$$

For a given fuzzy partition matrix U, the new codebook Y whose codeword Y_j (j=1, 2, ..,c) can be evaluated by minimizing $J_m=J_m(U,Y)$ as follows

$$Y_i=\frac{\sum_{i=1}^{M}u_{ji}^{m}X_i}{\sum_{i=1}^{M}u_{ji}^{m}} \tag{9}$$

The "fuzziness" of the fuzzy c-means algorithm is controlled by the parameter m, which is greater than unity. As the parameter m approaches unity, the partition of the training vector space is a nearly crisp decision-making process. In addition, U and Y depend on each other in an interesting way: if one of them is given, the optimal solution to the other one can be uniquely constructed.

3 TabuSearchBasedFCM Algorithms

The basic idea of the tabu search is to explore the search space of all feasible solutions by a sequence of moves and to forbid some search directions at a present iteration in order to avoid cycling and jump off local optima. The elements of move from the current solution to its selected neighbor are partially or completely recorded in the tabu list for the purpose of forbidding the reversal of the replacement in a number of future iterations. A basic tabu search algorithm is formulated as follows.

Step 0: Generate an initial solution S_{INIT} using any existing algorithm.

Step 1: Set the current solution $S_{CURR}=S_{INIT}$ and the best solution $S_{BEST}=S_{INIT}$

Step 2: Iterate the following steps for I_m times

Step 2.1 Generate N_S test solutions S_i(i=1,2.., N_S) by making small modifications to the current solution S_{CURR}

Step 2.2 Calculate the objective values of all test solutions S_i(i=1,2 .., N_S).

Step 2.3 Sort the objective values of all test solutions in increasing order. From the best test solution to the worst test solution,if the test solution is a non-tabu solution or it is a tabu solution but its objective value is better than the objective value of the best solution S_{BEST}, then choose this solution as the current solution S_{CURR}, go to step 2.4; otherwise,try next test solution. If all test solutions are tabu solutions, go to step 2.1.

Step 2.4 If the objective value of S_{CURR} is better than that of S_{BEST},Set $S_{BEST}=S_{CURR}$.

Step 2.5 If the tabu list is full, remove the oldest solution in the tabu list.

Step 2.6 Insert the current solution S_{CURR} into the tabu list.

Step 3: record the best solution and terminate the algorithm.

The minimization of $J_m=J_m(U,Y)$ implies two alternative tabu search based fuzzy c-means approaches to codebook generation algorithms: codebook-based (CB), (i.e. Y) and fuzzy partition-based (FPB), (i.e. U). In the codebook-based variant, a solution is described by a codebook. The codewords are represented by a c-length array of k-dimensional vectors. This is a natural way to describe the problem in the context of VQ. After all, the aim of the algorithm is to create a codebook; the fuzzy partitions are of secondary importance. The fuzzy partition matrix, however, is needed when evaluating the objective function of a solution and its elements are calculated using formula (8). In the fuzzy partition-based variant, a solution is described by a fuzzy partition matrix. Its corresponding codebook can be obtained by calculating formula (9), then its objective function can be calculated. For tabu search, this method needs more memory than the codebook-based variant and has poorer codebook performance than the codebook-based variant, so we adopt the codebook-based variant in tabu search-based fuzzy c-means algorithm.

The number of iterations (I_m) and the number of test solutions (N_S) together induce the total number of trail solutions ($I_m; N_S$). In general, the quality of the final solution is relatively independent of I_m and N_S as long as $I_m; N_S$ remains constant. Here, we set the values I_m=100 and N_S=20. In general, a small tabu list size (T_S) is usually sufficient. After all, the size of tabu list is limited by the number of test solutions N_S because only one solution per iteration is included in tabu list. Here we use T_S=20

The key question in the tabu search is the definition of neighborhood of a solution, i.e. the way new test solutions are generated. In the CB-variant, the randomizing can be implemented by the following methods: adding noise to codewords and swapping the codeword. The first method which is adopted from simulated annealing has the problem that the changes are local. In the swapping method, a codeword is replaced (with probability P) by a randomly chosen training vector. Thus more radical and non-local changes can be made in the codebook.

With the above materials in hand, we now illustrate our CB-TSFCM algorithm as follows:

Step 0: Generate an initial codebook C_{INIT} by randomly selecting codewords in the training set.

Step 1: Set the current codebook C_{CURR}=C_{INIT} and the best codebook C_{BEST}=C_{INIT}

Step 2: Iterate the following steps for I_m times

Step 2.1 Generate N_S neighboring codebooks C_i (i=1,2 ..., N_S) of the current solution C_{CURR} using the swapping method mentioned above.

Step 2.2 Calculate the fuzzy partition matrixes U_i of all test codebooks C_i (i=1,2.., N_S), then update the test codebooks C_i using formula (9) and calculate the objective function J_m=$J_m(U_i, C_i)$ for each C_i.

Step 2.3 Sort the objective function values of all test codebooks in increasing order. From the best test codebook to the worst test codebook,if the test codebook is a non-tabu codebook or it is a tabu codebook but its objective value is smaller than that of the best codebook C_{BEST}, choose this codebook as the current codebook C_{CURR}, go to step 2.4; otherwise,try next test codebook. If all test solutions are tabu codebooks, go to step 2.1.

Step 2.4 If the objective value of C_{CURR} is smaller than that of C_{BEST},set C_{BEST}=C_{CURR}.

Step 2.5 If the tabu list is full, remove the oldest codebook in the tabu list.

Step 2.6 Insert the current codebook C_{CURR} into the tabu list.

Step 3: record the best codebook and terminate the algorithm.

4 ExperimentaResults

This section presents an experimental evaluation of the involved algorithms based on several experiments using the 256×256 8bits/pixel Lena image. The experiments investigated the effect of the codebook size on the performance of the algorithms tested in this paper. The training vectors were image blocks of size 4×4=16. The c-means algorithm, the fuzzy c-means algorithm and the CB-TSFCM algorithm were used to design codebooks of size 8, 16, 32, 64, 128 and 256. Table 1 shows the PSNR of the images reconstructed from the various codebooks designed in this experiment. According to Table 1, with only few exceptions, the CB-TSFCM algorithm results in better codebooks than those designed by the c-means algorithm and the fuzzy c-means algorithm. Fig. 1 shows the original Lena image of size 256× 256. Figs. 2-4 show the images reconstructed from the codebook designed by various algorithms. The PSNR achieved by the c-means algorithm, the fuzzy c-means algorithm and the proposed algorithm are 27.69, 29.43, 30.13dB respectively.

5. Conclusions

This paper presents a tabu search based fuzzy *c*-means algorithm that is used to design codebooks. In this paper, a solution is described by a codebook for tabu search. Experimental results show that the CB-TSFCM algorithm can obtain better codebooks than the conventional fuzzy *c*-means algorithm and the *c*-means algorithm, although the proposed algorithm is rather time-consuming.

References

1. A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* Boston, MA: Kluwer, 1992.
2. Y. Linde, A. Buzo and R .M. Gray. An Algorithm for Vector Quantizer Design. *IEEE Trans. Commun.* 1980, COM-28:84-95
3. J. Vaisey and A. Gersho. Simulated Annealing and Codebook Design. *Proceedings ICASSP'88*,Apr. 1988:1176-1179
4. J. S. Pan, F. R. McInnes and M. A. Jack, VQ Codebook Design Using Genetic Algorithms. *Electronics Letters*, August, 1995, 31(17):1418-1419
5. K.A.Zeger, and A.Gersho. Stochastic Relaxation Algorithm for Improved Vector Quantizer Design. *Electronic Letters*, 1989, 25(14):896-898
6. V. Delport and D. Liesch, Fuzzy-c-mean algorithm for codebook design in vector quantization, Electronics Letters, 1994, 30(13): 1025-1026
7. F. Glover, M. Laguna, Tabu search, Kluwer Academic Publishers, 1997.

Table 1 PSNR(dB) of the images reconstructed from codebooks of various sizes designed by different algorithms (I_m=100, T_s=N_s=20, m=1.2)

Method	Codebook Size					
	8	16	32	64	128	256
CM	20.78	23.56	25.16	25.99	26.52	27.69
FCM	22.91	24.30	25.56	26.69	27.89	29.43
TSFCM	22.90	24.32	25.86	27.03	28.11	30.13

Fig. 1 Lena image

Fig. 2 Lena image reconstructed by CM method

(c) Fig. 3 Lena image reconstructed by FCM mthod

Fig. 4 Lena image reconstructed by TSFCM method